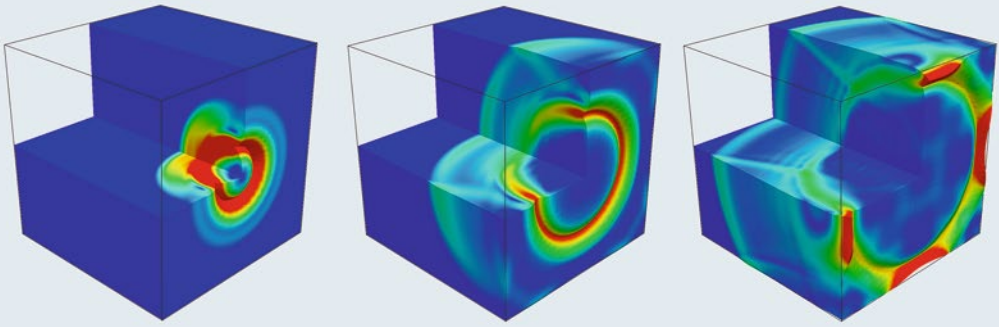


LECTURE NOTES IN COMPUTATIONAL SCIENCE AND ENGINEERING 100

Michael Griebel
Marc Alexander Schweitzer *Editors*

Meshfree Methods for Partial Differential Equations VII

Springer

Lecture Notes in Computational Science and Engineering

I OO

Editors:

More information about this series at
http://www.springer.com/series/3527

Michael Griebel • Marc Alexander Schweitzer
Editors

Meshfree Methods for Partial Differential Equations VII

Springer

Editors
Michael Griebel
Marc Alexander Schweitzer
Institut für Numerische Simulation
Universität Bonn
Bonn
Germany

ISSN 1439-7358 ISSN 2197-7100 (electronic)
ISBN 978-3-319-06897-8 ISBN 978-3-319-06898-5 (eBook)
DOI 10.1007/978-3-319-06898-5
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014956376

Mathematics Subject Classification (2010): 65N30, 65N75, 65M60, 65M75, 65Y99

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

The Seventh International Workshop on *Meshfree Methods for Partial Differential Equations* was held from September 9 to September 11, 2013 in Bonn, Germany. This workshop series was installed in 2001 to bring together European, American and Asian researchers working in this exciting field of interdisciplinary research on a regular basis.

To this end Ivo Babuška, Ted Belytschko, Jiun-Shyan Chen, Michael Griebel, Wing Kam Liu, Marc Alexander Schweitzer and Harry Yserentant invited scientists from all over the world to Bonn to strengthen the mathematical understanding and analysis of meshfree discretizations but also to promote the exchange of ideas on their implementation and application.

The workshop was again hosted by the Institut für Numerische Simulation at the Rheinische Friedrich-Wilhelms-Universität Bonn with the financial support of the Sonderforschungsbereich 1060 *The Mathematics of Emergent Effects* and the Hausdorff Center for Mathematics.

This volume of LNCSE now comprises selected contributions of attendees of the workshop. Their content ranges from applied mathematics to physics and engineering and even industrial applications, which clearly indicates the maturity meshfree methods have reached in recent years. They are becoming more and more mainstream in many areas of applications due to their flexibility and wide applicability.

Bonn, Germany
June 2014

Michael Griebel
Marc Alexander Schweitzer

Contents

A Galerkin Radial Basis Function Method for Nonlocal Diffusion

Stephen D. Bond, Richard B. Lehoucq, and Stephen T. Rowe

Abstract We introduce a meshfree Galerkin method for solving nonlocal diffusion problems. Radial basis functions are used to construct an approximation scheme that requires only scattered nodes with no triangulation. A quadrature scheme specific to radial basis functions is implemented to produce a Galerkin radial basis function method that yields fast assembly of a sparse stiffness matrix. We provide numerical evidence for convergence rates using one and two dimensional nonlocal problems.

Keywords Radial basis functions • Meshless method • Galerkin method • Nonlocal diffusion • Integral operator

1 Introduction

Classical diffusion models are formulated as partial differential equations that rely on Fick's first law. However, it has been observed in cases such as diffusion through heterogeneous material that the classical model is not an adequate description of diffusion [15]. Various models have been proposed for these cases of anomalous diffusion, which include models based on integral operators and fractional derivatives. The nonlocal operator has applications in a variety of fields besides anomalous diffusion. Examples include image analyses, nonlocal heat conduction, machine learning, and peridynamic mechanics. The nonlocal model allows for discontinuous solutions and also includes the fractional Laplacian as a special case; see [1, 5] for further discussion. In this paper, we introduce a numerical method for solving a model for nonlocal diffusion.

S.D. Bond • R.B. Lehoucq
Sandia National Laboratories, Computational Mathematics, P.O. Box 5800, MS 1320, Albuquerque, NM 87185-1320, USA
e-mail: sdbond@sandia.gov; rblehou@sandia.gov

S.T. Rowe (✉)
Department of Mathematics, Texas A&M University, College Station, TX 77843, USA
e-mail: srowe@math.tamu.edu

M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations VII*, Lecture Notes in Computational Science and Engineering 100,
DOI 10.1007/978-3-319-06898-5_1

Let $\Omega \subset \mathbb{R}^n$ denote a bounded, open domain. For $u : \Omega \cup \Omega_{\mathcal{I}} \to \mathbb{R}$, we define

$$\mathcal{L}u(x) := 2 \int_{\Omega \cup \Omega_{\mathcal{I}}} \big(u(y) - u(x)\big)\gamma(x, y)\, \mathrm{d}y \qquad \forall x \in \Omega \subseteq \mathbb{R}^n \,, \tag{1}$$

where $\gamma : \Omega \cup \Omega_{\mathcal{I}} \times \Omega \cup \Omega_{\mathcal{I}} \to \mathbb{R}$ is a nonnegative symmetric map, i.e., $\gamma(x, y) = \gamma(y, x) \geq 0$. The operator $\mathcal{L}$ is *nonlocal* because the pointwise value of $\mathcal{L}u(x)$ depends on points $y \neq x$. In contrast, $\Delta u(x)$, requires only information about u at x. We refer to $\Omega_{\mathcal{I}}$ as the interaction domain and γ as the kernel. As discussed below, the interaction domain is the nonlocal analogue of the boundary and is necessary in order to impose the nonlocal analogue of boundary conditions, what we refer to as volume constraints.

Our goal is to solve the steady-state nonlocal diffusion equation. Given a source function $f : \Omega \to \mathbb{R}$ and Dirichlet constraint function $g : \Omega_{\mathcal{I}} \to \mathbb{R}$, the problem is to find $u : \Omega \cup \Omega_{\mathcal{I}} \to \mathbb{R}$ such that

$$\begin{cases} \mathcal{L}u(x) = f(x) & x \in \Omega \\ u(x) = g(x) & x \in \Omega_{\mathcal{I}} \,. \end{cases} \tag{2}$$

For comparison, the classical steady-state form of the diffusion problem is

$$\begin{cases} \nabla \cdot (\mathbf{C}\nabla u)(x) = f(x) & x \in \Omega \\ u(x) = g(x) & x \in \partial\Omega \,, \end{cases} \tag{3}$$

where $\mathbf{C}$ is the diffusion tensor. Comparing (2) and (3), we see at least two differences. The first difference is that the nonlocal problem replaces a differential operator with an integral operator. The second difference is that the boundary condition on $\partial\Omega$ is replaced by a *volume* constraint on $\Omega_{\mathcal{I}}$. Imposing the volume constraint guarantees that the nonlocal problem is well-posed [5]. When $g \equiv 0$, then the solution u (2) is the probability density for the exit-time problem; see [2,4,6] for further details and information.

In this note, we propose a Galerkin radial basis function (RBF) method to numerically solve (2). We also exploit a recently developed Lagrange function quadrature scheme [7] for the necessary integrations. Consequently, our proposed method requires only information at the radial basis function nodes and also yields a straight forward assembly of a sparse stiffness matrix. A conforming discontinuous Galerkin method for a nonlocal diffusion problem was introduced in [5] where the basis functions are given by discontinuous piecewise polynomials. Assembly of this stiffness matrix results in a challenging problem in quadrature for two reasons. The first is that there are $2n$ iterated integrals, and the second is that the regions of integration involve partial element volumes. In contrast, the primary advantage of the Galerkin RBF method is that entries in the stiffness matrix only require

a pointwise evaluation of the kernel and multiplication by quadrature weights—complications arising from overlapping partial element volumes are irrelevant. A disadvantage of the use of radial basis functions relative to discontinuous piecewise polynomials includes a "Gibbs phenomenon" at any discontinuities of the solution u for (2).

For a fixed $x \in \Omega$, the horizon of γ at x is the radius of support of $\gamma(x, y)$. It is possible for γ to have multiple horizons that depend on x. The integrability of the kernel determines the smoothing action of the inversion of $\mathcal{L}$. We consider only integrable γ with no singularities, which implies that the inversion process does not smooth the data. In particular, discontinuous solutions are to be expected for discontinuous source functions. This should be contrasted with the case of a second order elliptic differential operator, where the solution is two orders smoother than the data.

The meshfree Galerkin method we introduce uses radial basis functions (RBFs). Radial basis functions have been extensively studied for meshfree interpolation and approximation. Radial basis functions have also been applied in a variety of areas besides interpolation of scattered data on subsets of $\mathbb{R}^n$. They have seen notable success in collocation methods for elliptic, parabolic, and hyperbolic partial differential equations and a variety of other PDEs, including stochastic differential equations. In particular, RBF collocation methods are well suited for high dimensional, high smoothness PDEs. RBF Galerkin methods have been investigated, but previously had difficulty with quadrature. In the setting of certain Riemannian manifolds such as the n-sphere $\mathbb{S}^n$, RBF methods can be extended and yield interpolation methods as well as collocation and Galerkin methods for solving partial differential equations on spheres [3, 8, 9, 12, 13, 16].

In Sect. 2, we discuss radial basis functions and interpolation. We motivate their use by considering the question of scattered data interpolation in arbitrary dimensions, followed by a discussion of positive definite and conditionally positive definite functions. We discuss a specific choice of basis, the Lagrange basis, along with a quadrature method that is used in the Galerkin radial basis function method we introduce. In Sect. 3, we consider a variational formulation of (2). Combining the variational form with radial basis functions and the quadrature technique described in Sect. 2, we propose a meshfree Galerkin RBF method. We then present numerical experiments in both one and two dimensions and discuss the results from the experiments in Sect. 4.

2 Radial Basis Functions

We now provide a brief review of radial basis function methods needed to discretize the nonlocal diffusion problem; the reader is referred to [17] for a comprehensive introduction and further information.

Let $\Omega \subset \mathbb{R}^n$. We say that $\Phi : \Omega \to \mathbb{R}$ is *radial* if there exists $\varphi : \mathbb{R}^+ \to \mathbb{R}$ such that $\Phi(x) = \varphi(\|x\|)$. Given a set of scattered centers $\{x_1, \ldots, x_N\} = X$ and a radial function Φ, we construct a collection of *radial basis functions* ϕ_j by defining $\phi_j(x) = \Phi(x - x_j) = \varphi(\|x - x_j\|)$. To interpolate a function $f : \Omega \to \mathbb{R}$ on the centers X, we construct an interpolant by enforcing

$$f(x_j) = \sum_{k=1}^{N} c_k \varphi(\|x_k - x_j\|) \qquad j = 1, \ldots, N \ ,$$

which yields a linear system $Tc = f$ of N equations in N unknowns, with $T_{jk} = \varphi(\|x_k - x_j\|)$ and $f_j = f(x_j)$. Note that we assume there no repeated points in the collection of centers. For the interpolation problem to have a unique solution, the interpolation matrix T must be invertible. The set of functions that generate an invertible matrix for any collection of centers is unknown, but the restriction to the set of functions which generate positive definite matrices for arbitrary sets of centers has been studied. These functions are positive definite functions and they uniquely solve the interpolation problem for any set of centers.

A popular choice of RBFs on $\mathbb{R}^n$ are the thin plate splines

$$\varphi(r) = \begin{cases} r^{2\ell - n} & n \text{ is even} \\ r^{2\ell - n} \log r & n \text{ is odd,} \end{cases} \tag{4}$$

for $\ell > \frac{n}{2}$. The approximation and interpolation properties of the thin plate splines have been studied extensively, but the thin plate splines are only conditionally positive definite functions. For example, let $2\ell - n = 2$ and given centers X, define $P|_X = \text{span}\{\rho_1, \ldots, \rho_{n+1}\}$, where $(\rho_k)_j = p_k(x_j)$, and $\{p_1, \ldots, p_{n+1}\}$ is a basis for the space of degree 1 polynomials. We say that the thin plate spline is conditionally positive definite if and only if for any collection of scattered sites $\{x_1, \ldots, x_N\}$, the quadratic form $\xi^T T \xi$ is positive for all nonzero ξ orthogonal to $P|_X$, with $T_{j,k} = \varphi(\|x_j - x_k\|)$. To ensure that interpolation with the conditionally positive definite thin plate spline is well defined, a linear polynomial must be included in the interpolant. This has the advantage of interpolating scattered data and reproducing constants and linear polynomials. The interpolant is of the form

$$s(x) = \sum_{j=1}^{N} \xi_j \varphi(\|x - x_j\|) + \sum_{k=1}^{n+1} \zeta_k p_k(x) \ ,$$

where n is the space dimension. The interpolation linear system is

$$\begin{pmatrix} T & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} \xi \\ \zeta \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix} \tag{5}$$

where $T_{j,k} = \varphi(\|x_j - x_k\|)$ is symmetric, $P_{k,l} = p_l(x_k)$, and $f_j = f(x_j)$.

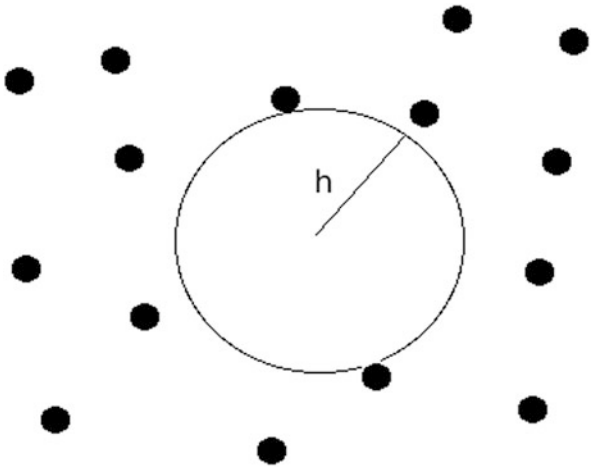


Fig. 1 The geometric interpretation of the mesh norm is the largest ball in Ω that does not contain any centers

The error between a function and its RBF interpolant can be shown to decrease as the density of the centers X, we define the mesh norm (or fill distance)

$$h = \sup_{x \in \Omega} \min_{x_j \in X} \|x - x_j\| ; \tag{6}$$

see Fig. 1. The convergence rate of the interpolant to the function depends on the smoothness of both the function and the RBF, with smoother functions being approximated at faster rates. For an example of an error estimate for a thin plate spline RBF in $W_2^m(\Omega)$, the Sobolev space of order m, we have

Theorem 1. *Given certain restrictions on* Ω *and for the thin plate spline in* $W_2^m(\Omega)$ *and function* $f \in W_2^\beta(\Omega)$ *for* $\frac{n}{2} < \beta \leq m$,

$$\|f - I_X f\|_{W_2^\mu(\Omega)} \leq C h^{\beta-\mu} \|f\|_{W_2^\beta(\Omega)} \qquad 0 \leq \mu < \beta \ , \tag{7}$$

where $I_X f$ *denotes the RBF interpolant of* f *on centers in* X *[14].*

The restriction $2\beta > n$ guarantees that f is a continuous function, by the Sobolev embedding theorem. Faster convergence rates are known when the function f is smoother than the RBF.

The thin plate spline was chosen over other RBFs due to its approximation properties and because the corresponding Lagrange basis functions enjoy an exponential decay. Recent results have noted the Lagrange function of the thin plate spline decays exponentially away from its center, which is not known for other RBFs [10, 11]. There is evidence that the Lagrange functions for the thin plate splines can be replaced with *local* Lagrange functions, which are cheaper to construct than the Lagrange functions. The local property has been recently proven for spheres,

and current work investigates these methods for domains in $\mathbb{R}^n$ [7, 8, 10]. The local property is due to results on both the decay of the Lagrange functions away from their centers and the decay of the coefficients of the Lagrange functions. Numerical evidence suggests these results can be extended to domains in $\mathbb{R}^n$ for Lagrange functions away from the boundary.

2.1 Lagrange Basis Functions and Quadrature

In this section, we discuss an alternate basis for RBF interpolation and a quadrature method unique to the basis. Given a thin plate spline and a set of centers $\{x_j\}_{j=1}^N$, we can construct a *Lagrange* basis of RBFs, $\{\chi_j\}_{j=1}^N$ that satisfy $\chi_j(x_k) = \delta_{j,k}$ and are given by

$$\chi_j(x) = \sum_{k=1}^{N} \xi_k \varphi(\|x - x_k\|) + \sum_{l=1}^{n+1} \zeta_l p_l(x) \, .$$

Figures 2–5 display a thin plate spline and Lagrange basis function. Figure 6 demonstrates that the Lagrange basis function is essentially of compact support due to the associated exponential decay. The Lagrange function χ_j is constructed by solving the linear system (5) with $f_k = \delta_{j,k}$. Therefore, the interpolant $I_X f$ of a function f is given by

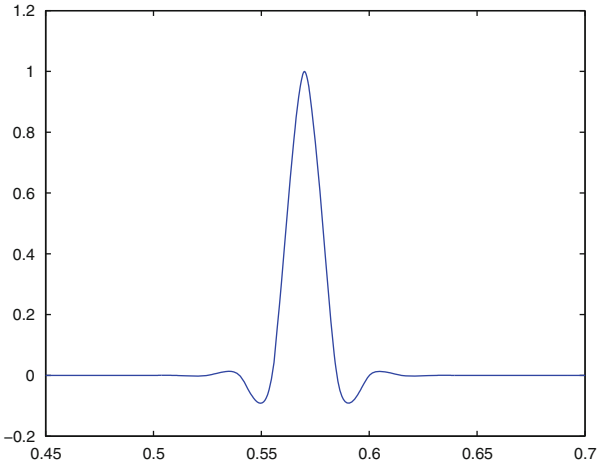


Fig. 2 A 1D Lagrange function constructed from 67 uniformly spaced points is displayed

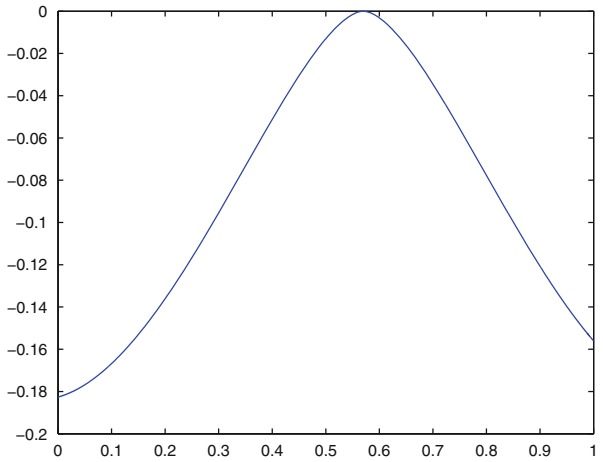


Fig. 3 A 1D thin plate spline centered at 0.57 is displayed

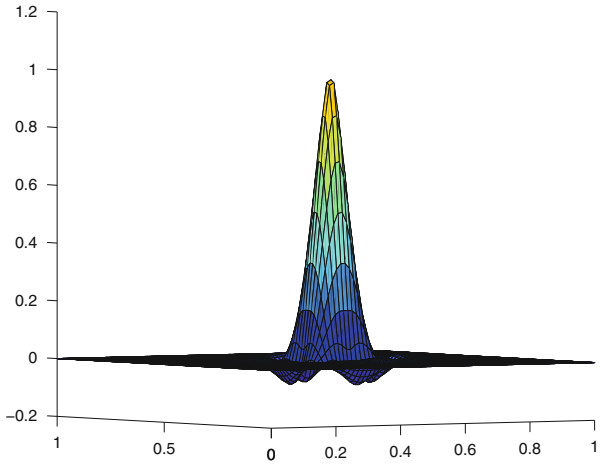


Fig. 4 A 2D Lagrange function constructed from 169 uniformly spaced points is displayed

$$I_X f(x) = \sum_{j=1}^{N} f(x_j)\chi_j(x).$$

A quadrature scheme can be constructed by using Lagrange functions. Given $F : \Omega \to \mathbb{R}$ and centers $\{x_j\}_{j=1}^N$, we approximate the integral of F over $S \subset \Omega$ by

$$\int_S F(x)\,\mathrm{d}x \approx \sum_{x_j \in S} F(x_j)w_j, \qquad w_j = \int_\Omega \chi_j(x)\,\mathrm{d}x\ . \tag{8}$$

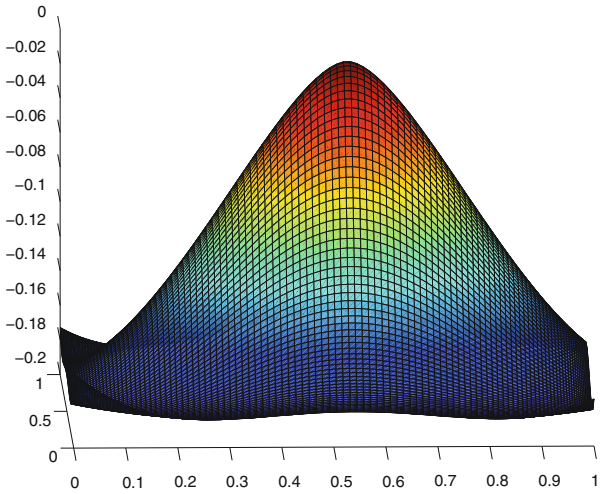


Fig. 5 A thin plate spline basis function centered at $(0.54, 0.54)$ is displayed

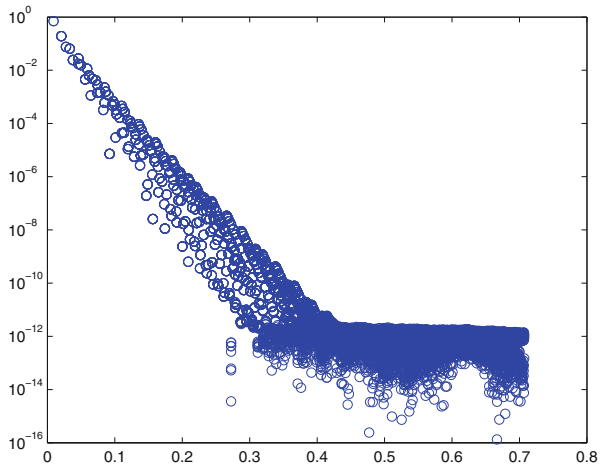


Fig. 6 The exponential decay of a Lagrange function away from its center is displayed. The Lagrange function was constructed using 1,681 centers and evaluated on 14,400 points

When $S = \Omega$, we can derive the following quadrature error estimate:

Lemma 1. *Let* $\Omega \subset \mathbb{R}^n$ *and* $X = \{x_j\}_{j=1}^N$, $\{w_j\}_{j=1}^N$ *be the quadrature weights in* (8)*, and* $F(x) \in W_2^\beta(\Omega)$ *for* $\frac{n}{2} < \beta \leq 2$*. Then, the quadrature error is bounded by*

$$\left| \int_\Omega F(x)\,\mathrm{d}x - \sum_{j=1}^{N} F(x_j) w_j \right| \leq C h^\beta \|F\|_{W_2^\beta(\Omega)} .$$

Proof. This follows by an application of Theorem 1, (8), and the Cauchy-Schwarz inequality.

$$\begin{aligned} \left| \int_\Omega F(x)\,\mathrm{d}x - \sum_{j=1}^{N} F(x_j) w_j \right| &= \left| \int_\Omega F(x) - \sum_{j=1}^{N} F(x_j)\chi_j(x)\,\mathrm{d}x \right| \\ &\leq \sqrt{\mu(\Omega)} ||F - I_X F||_{L^2(\Omega)} \\ &\leq C\sqrt{\mu(\Omega)} h^\beta ||F||_{W_2^\beta(\Omega)} \end{aligned}$$

where $\mu(\Omega)$ is the measure of Ω, which we absorb into the constant C to achieve the result. □

The restriction on β ensures that F is continuous and we assume necessary geometric regularity on the domain Ω. Our use of the above Lemma will be consistent with these assumptions. A scheme for directly computing the quadrature weights without computing the χ_j is described in Sect. 3.1.

3 Variational Formulation

In this section, we derive a variational formulation of (2), which then can be solved using a Galerkin method. We define the bilinear form $a(u, v)$ by

$$a(u,v) = \int_{\Omega\cup\Omega_\mathcal{I}} \int_{\Omega\cup\Omega_\mathcal{I}} \big(u(x) - u(y)\big)\big(v(x) - v(y)\big)\gamma(x,y)\,\mathrm{d}y\,\mathrm{d}x.$$

We define the energy functional E by

$$\begin{aligned} E(u) &= \frac{1}{2}a(u,u) - \int_\Omega u f\,\mathrm{d}x \\ &\text{subject to } u = g \text{ over } \Omega_\mathcal{I}. \end{aligned} \tag{9}$$

In [5], the problem of finding the minimum of the energy functional was shown to be well-posed for u in an energy constrained space $L_c^2(\Omega \cup \Omega_\mathcal{I})$. In contrast, we minimize the functional by the method of Lagrange multipliers because the RBF basis is not contained in the energy constrained space. The Lagrangian is defined as

$$L(u,\lambda) = E(u) + \int_{\Omega_{\mathcal{I}}} \lambda(y)\big(u(y) - g(y)\big)\,\mathrm{d}y,$$

where $\lambda \in L^2(\Omega_{\mathcal{I}})$ is the Lagrange multiplier. Then, for every $v \in L^2(\Omega \cup \Omega_{\mathcal{I}})$ and $w \in L^2(\Omega_{\mathcal{I}})$, we minimize the Lagrangian by

$$\frac{\mathrm{d}}{\mathrm{d}t} L(u + tv, \lambda)\big|_{t=0} = \frac{\mathrm{d}}{\mathrm{d}t} E(u + tv)\big|_{t=0} + \int_{\Omega_{\mathcal{I}}} \lambda(y) v(y)\,\mathrm{d}y = 0$$

$$\frac{\mathrm{d}}{\mathrm{d}t} L(u, \lambda + tw)\big|_{t=0} = \int_{\Omega_{\mathcal{I}}} w(y)\big(u(y) - g(y)\big)\,\mathrm{d}y.$$

Computing these derivatives, the variational form of the problem is to find $u \in L^2(\Omega \cup \Omega_{\mathcal{I}})$ such that for each $v \in L^2(\Omega \cup \Omega_{\mathcal{I}})$ and each $w \in L^2(\Omega_{\mathcal{I}})$,

$$\begin{cases} a(u,v) + \displaystyle\int_{\Omega_{\mathcal{I}}} \lambda(y) v(y)\,\mathrm{d}y = \int_{\Omega} v(x) f(x)\,\mathrm{d}x \\ \displaystyle\int_{\Omega_{\mathcal{I}}} w(y) u(y)\,\mathrm{d}y = \int_{\Omega_{\mathcal{I}}} w(y) g(y)\,\mathrm{d}y. \end{cases} \tag{10}$$

We discretize this system by choosing finite dimensional subspaces $U_h \subset L^2(\Omega \cup \Omega_{\mathcal{I}})$ and $\Lambda_h \subset L^2(\Omega_{\mathcal{I}})$. Let $U_h = \operatorname{span}\{\phi_i\}_{i=1}^{N}$ and $\Lambda_h = \operatorname{span}\{\psi_k\}_{k=1}^{N_I}$. Then, we express the discrete solution and Lagrange multipliers in these bases as

$$u_h = \sum_{i=1}^{N} \alpha_i \phi_i \qquad \lambda_h = \sum_{k=1}^{N_I} \beta_k \psi_k.$$

To construct the linear system to solve for the coefficients of u_h and λ_h, we insert u_h and λ_h into (10), and we choose $v \in U_h$ and $w \in \Lambda_h$. The resulting linear system is

$$\begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} b \\ c \end{pmatrix}, \tag{11}$$

where the entries in the matrices are given by

$$A_{i,j} = \int_{\Omega\cup\Omega_{\mathcal{I}}} \int_{\Omega\cup\Omega_{\mathcal{I}}} \big(\phi_j(x) - \phi_j(y)\big)\big(\phi_i(x) - \phi_i(y)\big)\,\gamma(x,y)\,\mathrm{d}y\,\mathrm{d}x \tag{12a}$$

$$b_i = \int_{\Omega} \phi_i\, f\,\mathrm{d}y, \qquad B_{i,k} = \int_{\Omega_{\mathcal{I}}} \psi_k\, \phi_i\,\mathrm{d}y, \qquad c_k = \int_{\Omega_{\mathcal{I}}} \psi_k\, g\,\mathrm{d}y. \tag{12b}$$

3.1 Galerkin Radial Basis Function Method

We propose a Galerkin RBF method for two reasons. First, the Galerkin RBF method leads to a sparse stiffness matrix by exploiting a Lagrange function quadrature rule. Otherwise, since the basis functions are globally supported, the stiffness matrix is dense. The second reason to consider this method is the ease of assembly of the stiffness matrix. In contrast with a finite element method using discontinuous piecewise polynomials, assembly of the Galerkin RBF stiffness matrix only requires pointwise evaluations of the kernel and multiplication by quadrature weights. However, we must overcome the difficulty of computing the quadrature weights, which we discuss Sect. 4.3 and in the remainder of this section.

We construct a Galerkin RBF method by using Lagrange function quadrature (8) on each of the entries in the matrices and vectors in (12a) and (12b). Choose the approximation space $U_h = \text{span}\{\chi_i\}_{i=1}^{N}$, where χ_i are the Lagrange functions of the RBF $\varphi(r) = r^2 \log r$ over some collection of centers $X \subset \Omega \cup \Omega_{\mathcal{I}}$. Let $\Lambda_h = \text{span}\{\chi_{j(k)}\}_{k=1}^{N_I}$, where the function $j : \{1, \ldots, N_I\} \to \{1, \ldots, N\}$ selects each index in $\{1, \ldots, N\}$ such that $\chi_{j(k)}$ is centered at a point in $\Omega_{\mathcal{I}}$.

A practical quadrature method is realized by a slight modification of a recent scheme proposed by Fuselier, Hangelbroek, Narcowich, Ward, and Wright [7]. The necessary quadrature weights w_j, see (8), can be constructed by solving the linear system

$$\begin{pmatrix} T & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} w \\ d \end{pmatrix} = \begin{pmatrix} v \\ \eta \end{pmatrix} \tag{13}$$

where $v_k = \int_\Omega \varphi(\|x - x_k\|)\mathrm{d}x$ and $\eta_l = \int_\Omega p_l(x)\mathrm{d}x$; see the paper [7] for further details. This choice of quadrature scheme has important practical considerations; see Sect. 4.3.

In (12), we substitute $\phi_i(x) = \chi_i(x)$ and $\psi_k(x) = \chi_{j(k)}(x)$. We approximate the integrals by using the Lagrange function quadrature. Using this quadrature rule, the entries in $A_{i,j}$ are approximated by

$$A_{i,j} \approx 2\delta_{i,j} w_i \int_{\Omega \cup \Omega_{\mathcal{I}}} \gamma(x, x_i)\, \mathrm{d}x - 2 w_i w_j \gamma(x_i, x_j). \tag{14}$$

We apply the quadrature scheme to the other values in (12) and compute

$$b_i \approx f(x_i) w_i 1_\Omega(x_i), \qquad B_{i,k} \approx \delta_{i,j(k)} w_i 1_{\Omega_{\mathcal{I}}}(x_i), \qquad c_k \approx g(x_k) w_k 1_{\Omega_{\mathcal{I}}}(x_k),$$

where 1_S is the indicator function for a set S defined by

$$1_S(x) = \begin{cases} 1 & x \in S \\ 0 & x \notin S. \end{cases}$$

The sparsity of the Galerkin RBF matrix (14) depends on the mesh norm h and the horizon ϵ of the kernel, for centers such that $\|x_i - x_j\| \geq \epsilon$, $A_{ij} = 0$. If the number of centers is increased, then the number of centers such that $\|x_i - x_j\| < \epsilon$ can increase, which increases the number of nonzero entries per row. If the Lagrange function quadrature is not used, then a dense stiffness matrix is generated due to the global support of the Lagrange functions.

4 Numerical Results

We present results from numerical experiments in both one and two dimensions. We present numerical convergence rates in terms of the mesh norm h given by (6). The source functions are generated by applying the integral operator to solutions u with prescribed behavior. These choices allow us to access the effect of quadrature error upon kernel choice, and convergence rate when the solution u contains a jump discontinuity. Since there is currently no approximation theory for the Galerkin RBF method, we also present interpolation error rates on uniformly spaced centers as a comparison, and compare with the theoretical estimates established in [5] for a finite element discretization of the nonlocal diffusion equation using a piecewise discontinuous polynomial basis.

For all tests, we assume zero Dirichlet volume constraints (i.e., $g(x) = 0$ in (2)). In both one and two dimensions, we consider an infinitely smooth kernel and a continuous, but not everywhere differentiable, triangular kernel

$$\gamma(x, y) = \begin{cases} \exp(-\frac{1}{1-\epsilon^{-2}\|x-y\|^2}) 1_{\{\|x-y\|<\epsilon\}}(x, y) & \text{Smooth kernel} \\ (1 - \frac{1}{\epsilon}\|x - y\|) 1_{\{\|x-y\|<\epsilon\}}(x, y) & \text{Triangular kernel} \end{cases} \tag{15}$$

where $1_{\{\|x-y\|<\epsilon\}}$ is the indicator function for the set $\{(x, y) : \|x - y\| < \epsilon\}$; see Figs. 7 and 8. We consider kernel functions of varying smoothness to observe

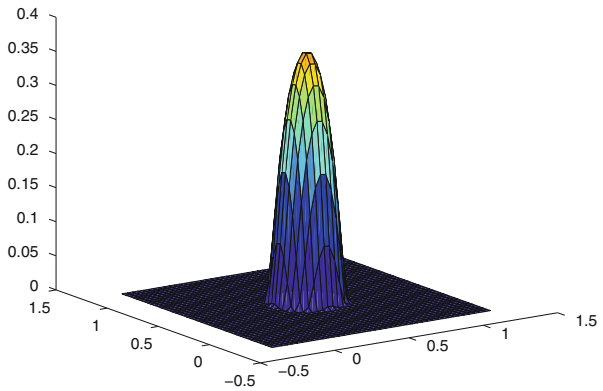


Fig. 7 The smooth kernel $\gamma(x, y)$ with $y = (0.5, 0.5)$ and $\epsilon = \frac{1}{4}$

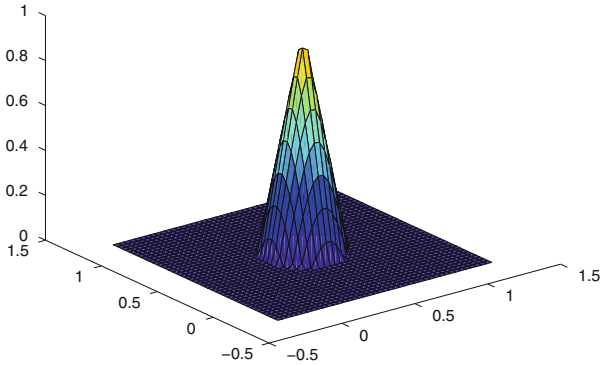


Fig. 8 A 1D thin plate spline centered at 0.57 is displayed

how the smoothness of the kernel affects the convergence rate of the RBF solution. The smoothness of the kernel affects the quadrature error induced by the Lagrange function quadrature method when assembling the stiffness matrix, so we expect lower convergence rates for lower smoothness kernel functions.

4.1 1D Experiments

Let $\Omega = (0, 1)$ and $\Omega_{\mathcal{I}} = [-\frac{1}{4}, 0] \cup [1, \frac{5}{4}]$, and we set $\epsilon = \frac{1}{4}$. We choose the thin plate spline $\varphi(r) = r^2$, which corresponds to the case $\ell = 1$ and $d = 1$ in (4). We choose two solutions and we manufacture the corresponding source function for each. We consider continuously differentiable and discontinuous solutions u_1 and u_2, respectively

$$u_1(x) = \big(1 - \cos(2\pi x)\big)1_\Omega(x)\,, \text{ and } u_2(x) = 1_{\{0 \le x \le \frac{1}{2}\}}(x).$$

For both u_1 and u_2, we solve the Galerkin RBF problem for both the smooth and triangular kernels, respectively. We tested using uniformly spaced centers with mesh norm $h = 0.005, 0.0025, 0.00125$, and 0.0005. For a second set of tests, we used the same sets of uniformly spaced centers, but the interior points were perturbed by a random number bounded in magnitude by $\frac{h}{5}$. Figures 9–12 display convergence rate plots for the non-uniformly spaced centers experiment. Table 1 displays convergence rates for the uniformly spaced experiments and the scattered experiments. For comparison, we display the convergence rate for interpolation using the uniformly spaced centers. As can be seen in Table 2, the condition number of the stiffness matrix is not increasing as the mesh norm decreases. In comparison, the paper [5] establishes that the condition number of the stiffness matrix does not increase as the mesh size decreases for a finite element discretization using a piecewise discontinuous polynomial basis.

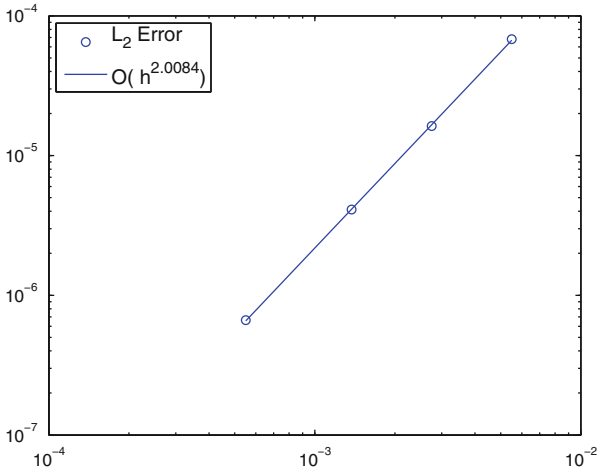


Fig. 9 The log of h versus the log of the L^2 error for the continuous solution u_1 with the smooth kernel functions using non-uniformly spaced centers is displayed

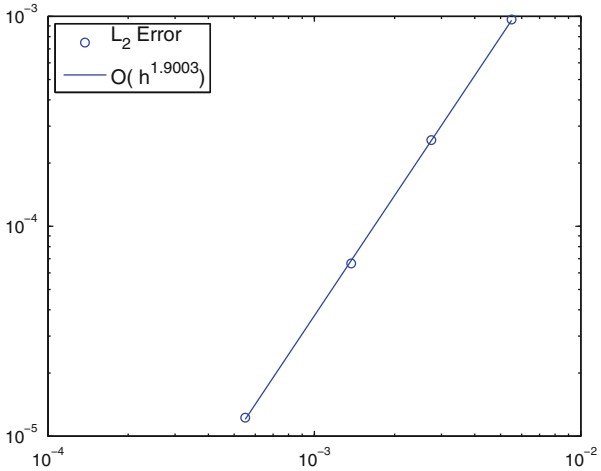


Fig. 10 The log of h versus the log of the L^2 error for the continuous solution u_1 with the triangular kernel functions using non-uniformly spaced centers is displayed

4.2 2D Experiments

Let $\Omega = (0,1) \times (0,1)$ and $\Omega_{\mathcal{I}} = ([-\frac{1}{4}, \frac{5}{4}] \times [-\frac{1}{4}, \frac{5}{4}]) \backslash \Omega$. We choose the thin plate spline $\varphi(r) = r^2 \log r$, which corresponds to the case $\ell = 2$ and $d = 2$ in (4). We consider two solutions and we manufacture the corresponding source function for each. We consider the continuous function w_1 and the discontinuous function w_2

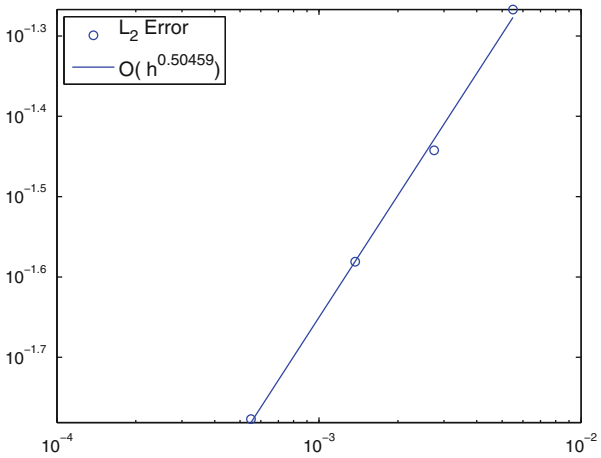


Fig. 11 The log of h versus the log of the L^2 error for the discontinuous solution u_2 with the smooth kernel functions using non-uniformly spaced centers is displayed

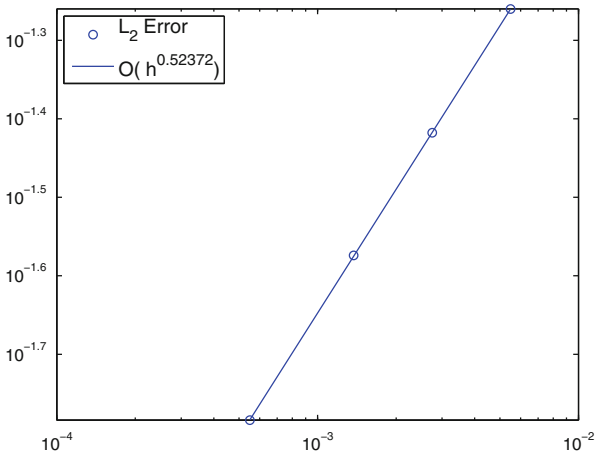


Fig. 12 The log of h versus the log of the L^2 error for the discontinuous solution u_2 with the triangular kernel functions using non-uniformly spaced centers is displayed

Table 1 Convergence rates for 1D experiments are displayed. The interpolation rates use the uniformly spaced centers

	Uniformly spaced		Non-uniformly spaced		
	Smooth	Triangular	Smooth	Triangular	Interpolation
u_1	2.0	1.8	2.0	1.9	2.0
u_2	0.52	0.52	0.50	0.52	0.51

Table 2 For 1D experiments, the mesh norm h, number of rows n of the stiffness matrix (11), and the reciprocal condition number for (11) for both the smooth and triangular kernel are displayed

		Reciprocal condition number	
h	n	Smooth	Triangular
5.00e−3	303	4.97e−3	8.36e−3
2.50e−3	603	4.93e−3	8.28e−3
1.25e−3	1,203	4.92e−3	8.24e−3
5.00e−4	3,003	4.91e−3	8.22e−3

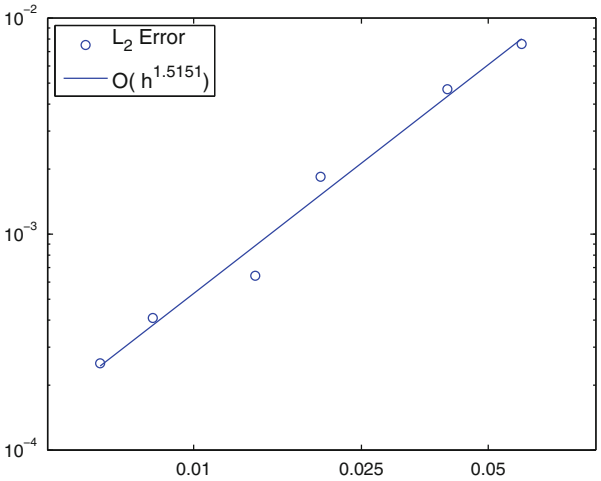


Fig. 13 The log of h versus the log of the L^2 error for the 2D continuous solution w_1 with the smooth kernel functions using non-uniformly spaced centers is displayed

$$w_1(x, y) = \sin(2\pi x)\sin(2\pi y)1_\Omega(x, y),$$
$$w_2(x, y) = (1 - \cos(2\pi x))(1 - \cos(2\pi y))1_{\Omega_1}(x, y)$$

with $\Omega_1 = [0, \frac{1}{2}] \times [0, 1]$.

For both w_1 and w_2, we solve the Galerkin RBF problem for both the smooth kernel and the triangular kernel. For w_1, we tested with six sets of uniformly spaced centers with mesh norms given in Table 4. For a second set of tests, we used the same sets of uniformly spaced centers, but the interior points were perturbed by a random vector bounded in magnitude by $\frac{2}{15}h$. For w_2, the tests were computed with the first four sets of centers. Figures 13–16 displays convergence rate plots for the non-uniformly spaced centers experiment. Table 3 displays convergence rates for the uniformly spaced experiments and the scattered experiments. For comparison, we display the convergence rate for interpolation using the uniformly spaced centers. The L^2 error is computed by evaluating the solution on a set of

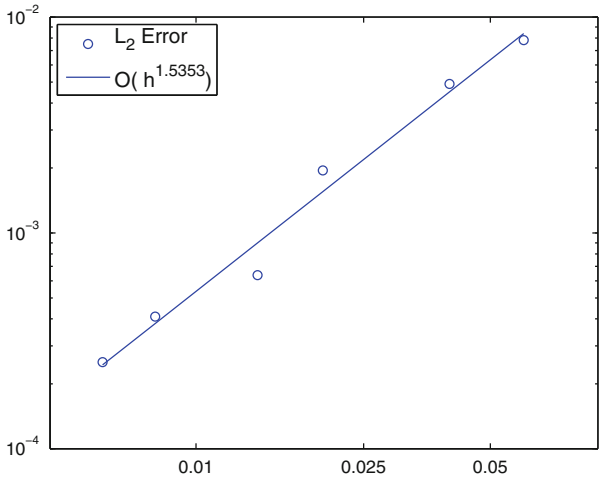


Fig. 14 The log of h versus the log of the L^2 error for the 2D continuous solution w_1 with the triangular kernel functions using non-uniformly spaced centers is displayed

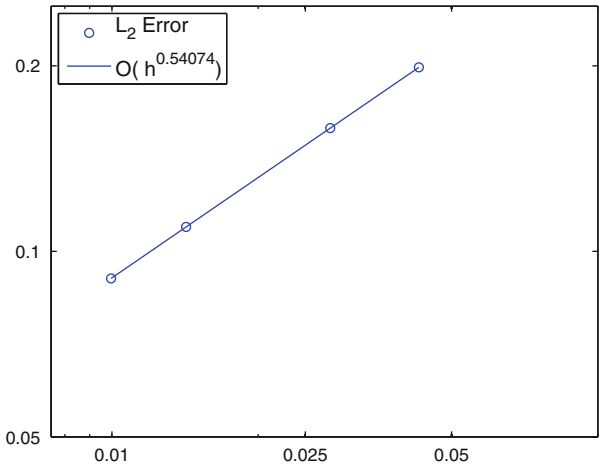


Fig. 15 The log of h versus the log of the L^2 error for the 2D discontinuous solution w_2 with the smooth kernel functions using non-uniformly spaced centers is displayed

22,500 nodes for the experiments with w_1 and 16,000 nodes for the tests with w_2; see Sect. 4.3 for details on the discontinuous solution w_2. The evaluation point set X_e is constructed by taking tensor products of Gauss-Legendre nodes on $[-\frac{1}{4}, \frac{5}{4}]$. In the case of non-uniformly spaced centers, the mesh norm h is approximated by $\max_{x\in X_e} \min_{x\in X} \|x - x_e\|$. As can be seen in Table 4, the condition number of the stiffness matrix is not increasing as the mesh norm decreases.

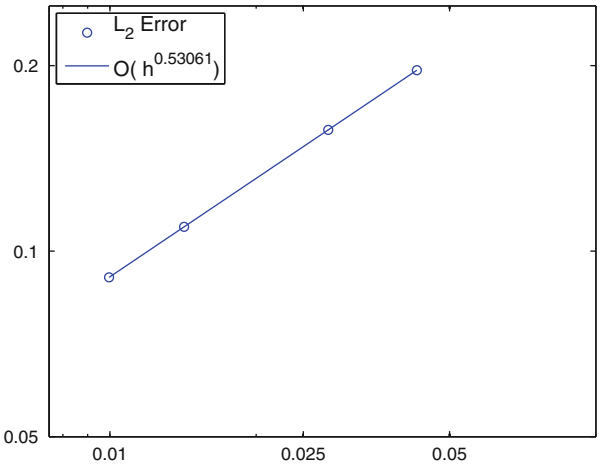


Fig. 16 The log of h versus the log of the L^2 error for the 2D discontinuous solution w_2 with the triangular kernel functions using non-uniformly spaced centers is displayed

Table 3 Convergence rates for 2D experiments are displayed. The interpolation rates use the uniformly spaced centers

	Uniformly spaced		Non-uniformly spaced		
	Smooth	Triangular	Smooth	Triangular	Interpolation
w_1	1.49	1.49	1.5	1.5	1.5
w_2	0.56	0.55	0.54	0.53	0.56

Table 4 For 2D experiments, the mesh norm h, number of rows n of the stiffness matrix (11), and the reciprocal condition number for (11) for both the smooth and triangular kernel are displayed

		Reciprocal condition number	
h	n	Smooth	Triangular
4.24e−2	1,096	2.10e−3	5.42e−3
2.82e−2	2,263	2.05e−3	4.93e−3
1.41e−2	9,052	2.10e−3	5.03e−3
9.90e−3	18,144	2.07e−3	5.01e−3
5.66e−3	55,063	2.31e−3	5.16e−3
4.24e−3	98,113	2.31e−3	5.17e−3

4.3 Computational Issues

In this section, we discuss some computational issues and aspects of the Galerkin RBF method. We discuss the costs of assembly, computation of L^2 errors, and construction of the source function for the experiments in Sect. 4.

Assembly of the stiffness matrix requires the construction of the Lagrange function quadrature weights. The quadrature weights are constructed by solving a

dense, symmetric $(N+3)\times(N+3)$ linear system, where N is the number of centers. The quadrature weights for experiments with $N \leq 12{,}000$ are computed directly by inverting the interpolation matrix with MATLAB's backslash operator. For the 2D experiments with $N = 35{,}344$ and $N = 63{,}001$, the quadrature weights are approximated by the generalized minimum residual method (GMRES). The matrix-vector product can be computed using the formulas for entries in the interpolation matrix (5), which avoids storing the dense matrix in memory.

Future work will investigate ways of computing the weights faster. Assembly of the stiffness matrix with the quadrature weights follows by the formula (14), which requires pointwise evaluations of the kernel and multiplication by the computed quadrature weight.

The L^2 error between the Galerkin solution u_h and the solution u is computed on the set $\Omega \cup \Omega_{\mathcal{I}}$ using Gauss-Legendre quadrature for 1D problems and tensor products of Gauss-Legendre nodes in 2D. For 1D experiments, 10,000 Gauss-Legendre nodes are used to compute the error. In 2D, 14,400 nodes in total are used, which are formed from products of 120 Gauss-Legendre nodes over $[-\frac{1}{4}, \frac{5}{4}]$. For a 2D discontinuous solution, the domain is split into two sets, Ω_1 and Ω_2, on which the solution is continuous. On each set, 8,000 quadrature nodes are formed from products of 1D Gauss-Legendre quadrature weights, and the L^2 error is computed by

$$\|u - u_h\|^2_{L^2(\Omega\cup\Omega_{\mathcal{I}})} = \|u - u_h\|^2_{L^2(\Omega_1)} + \|u - u_h\|^2_{L^2(\Omega_2)}.$$

For the experiments in Sect. 4, we require pointwise evaluations of the source function f. We manufacture $f(x_i)$ by computing $\mathcal{L}u(x_i)$ by (2). For one dimensional experiments, this is computed by using MATLAB's integral function. The integrand is written as an anonymous function, which is then integrated from $\max(-\epsilon, x_i - \epsilon)$ to $\min(x_i + \epsilon, 1 + \epsilon)$. For two dimensional experiments, the kernel is supported on a ball of radius ϵ. Converting the integral to polar coordinates, we use tensor product Gauss-Legendre quadrature weights over the rectangular region $[0, \epsilon] \times [0, 2\pi]$. The right hand side vector is assembled by $b_i = f(x_i)w_i$, where w_i is the Lagrange function quadrature weight. It has been observed that if f is discontinuous and x_i is placed on or near the discontinuity, the approximation $b_i = f(x_i)w_i$ did not perform well in experiments. For these nodes, we instead apply Gauss-Legendre quadrature to compute $b_i = \sum f(y_l)\chi_i(y_l)q_l$, where y_l are quadrature nodes and q_l are quadrature weights.

The RBF Galerkin solution u_h suffers from a "Gibbs phenomenon" at a point of discontinuity of u. In a neighborhood of the discontinuity, u_h overshoots the values of u; see Fig. 17. As the mesh norm decreases, the width of the overshoot decreases, although the height does not.

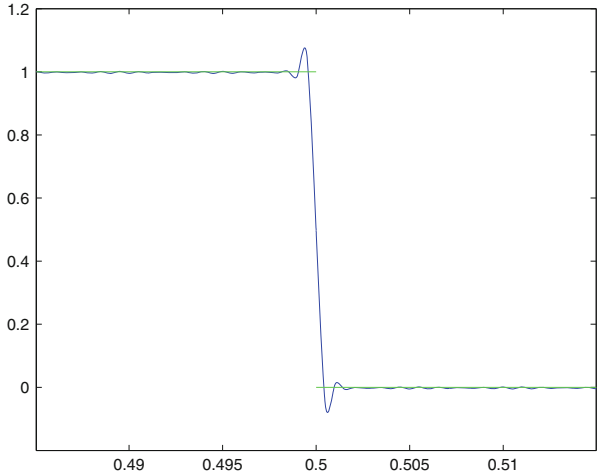


Fig. 17 A step function u (*green*) and the Galerkin RBF approximation u_h (*blue*) are displayed. The overshoot and undershoot of u_h occurs to the left and right of the discontinuity at 0.5

5 Conclusions and Future Work

We developed a Galerkin method for nonlocal diffusion by using Lagrange functions of radial basis functions and a recently developed quadrature method. We observed numerical evidence of L^2 convergence for both continuous and discontinuous solutions using kernel functions of varying smoothness in one and two dimensions. Further work is required to speed up the computation of the quadrature weights, establish error estimates, consider kernel functions with multiple horizons, and develop experiments in three dimensions.

Acknowledgements The authors would like to thank Francis Narcowich and Joseph Ward of Texas A&M University for advice and assistance on the project. The work of the second and third authors was supported by the Laboratory Directed Research and Development (LDRD) program at Sandia National Laboratories. Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94AL85000.

References

1. N. Burch, R.B. Lehoucq, Classical, nonlocal, and fractional diffusion equations on bounded domains. Int. J. Multiscale Comput. Eng. **9**, 661–674 (2011)
2. N. Burch, R.B. Lehoucq, Computing the exit-time for a symmetric finite-range jump process. Technical report SAND 2013-2354J, Sandia National Laboratories (2013). Available at http://www.sandia.gov/~rblehou/snl-sand2013-2354J.pdf

3. I. Cialenco, G.E. Fasshauer, Q. Ye, Approximation of stochastic partial differential equations by a kernel-based collocation method. Int. J. Comput. Math. **89**(18), 2543–2561 (2012)
4. M. D'Elia, Q. Du, M. Gunzburger, R.B. Lehoucq, Finite range jump processes and volume-constrained diffusion problems. Technical report SAND 2014-2584J, Sandia National Laboratories (2014). Available at http://www.sandia.gov/~rblehou/snl-sand2014-2584J.pdf
5. Q. Du, M. Gunzburger, R.B. Lehoucq, K. Zhou, Analysis and approximation of nonlocal diffusion problems with volume constraints. SIAM Rev. **54**(4), 667–696 (2012)
6. Q. Du, Z. Huang, R.B. Lehoucq, Nonlocal convection-diffusion volume-constrained problems and jump processes. Discret. Contin. Dyn. Syst. Ser. B (DCDS-B) **19**(2), 373–389 (2014). doi:10.3934/dcdsb.2014.19.373
7. E.J. Fuselier, T. Hangelbroek, F.J. Narcowich, J.D. Ward, G.B. Wright, Kernel based quadrature on spheres and other homogeneous spaces. Numer. Math. (2013). Appearing electronically, available at http://dx.doi.org/10.1007/s00211-013-0581-1
8. E.J. Fuselier, T. Hangelbroek, F.J. Narcowich, J.D. Ward, G.B. Wright, Localized bases for kernel spaces on the unit sphere. SIAM J. Numer. Anal. **51**(5), 2538–2562 (2013)
9. Q.T.L. Gia, I.H. Sloan, H. Wendland, Multiscale RBF collocation for solving PDEs on spheres. Int. J. Comput. Math. **121**(1), 99–125 (2012)
10. T. Hangelbroek, The penalized Lebesgue constant for surface spline interpolation. Proc. Am. Math. Soc. **140**, 173–187 (2012)
11. T. Hangelbroek, F.J. Narowich, J.D. Ward, Kernel approximation on manifolds I: bounding the Lebesgue constant. SIAM J. Math. Anal. **42**(4), 1732–1760 (2010)
12. E.J. Kansa, Multiquadrics – a scattered data approximation scheme with applications to computational fluid dynamics – II solutions to parabolic, hyperbolic, and elliptic partial differential equations. Comput. Math. Appl. **19**(8–9), 147–161 (1990)
13. F.J. Narcowich, S.T. Rowe, J.D. Ward, A novel Galerkin method for solving PDEs on the sphere using highly localized Kernel bases. ArXiv e-prints (2014)
14. F.J. Narcowich, J.D. Ward, H. Wendland, Sobolev error estimates and a Bernstein inequality for scattered data interpolation via radial basis functions. Constr. Approx. **24**(2), 175–186 (2006)
15. S.P. Neuman, D.M. Tartakovsky, Perspective on theories of non-Fickian transport in heterogeneous media. Adv. Water Resour. **32**(5), 670–680 (2009)
16. H. Wendland, Meshless Galerkin methods using radial basis functions. Math. Comput. **68**(228), 1521–1531 (1999)
17. H. Wendland, *Scattered Data Approximation* (Cambridge University Press, Cambridge/New York, 2005)

A Partition of Unity Method for the Obstacle Problem of Simply Supported Kirchhoff Plates

Susanne C. Brenner, Christopher B. Davis, and Li-yeng Sung

Abstract We consider a partition of unity method (PUM) for the displacement obstacle problem of simply supported Kirchhoff plates. We show that this method converges optimally in the energy norm on general polygonal domains provided that appropriate singular enrichment functions are included in the approximation space. The performance of the method is illustrated by numerical examples.

Keywords Partition of unity method • Fourth order variational inequality

1 Introduction

Let Ω be a bounded polygonal domain in $\mathbb{R}^2$, $f \in L_2(\Omega)$, and $\psi_1, \psi_2 \in C^2(\Omega) \cap C(\bar{\Omega})$ be two obstacle functions such that

$$\psi_1 < \psi_2 \text{ in } \Omega \quad \text{and} \quad \psi_1 < 0 < \psi_2 \text{ on } \partial\Omega. \tag{1}$$

Consider the following variational inequality: Find $u \in K$ such that

$$a(u, v-u) \geq (f, v-u) \quad \forall v \in K \tag{2}$$

where $(\cdot,\cdot)$ is the $L_2(\Omega)$ inner product, $a(\cdot,\cdot)$ is the bilinear form defined by

$$a(w, v) = \int_\Omega \Delta u \Delta v \mathrm{d}x,$$

and

$$K = \{v \in H^2(\Omega) \cap H_0^1(\Omega) : \psi_1 \leq v \leq \psi_2 \text{ in } \Omega\}.$$

S.C. Brenner (✉) • C.B. Davis • L.-y. Sung
Department of Mathematics, Center for Computation & Technology, Louisiana State University, Baton Rouge, LA 70803, USA
e-mail: brenner@math.lsu.edu; cdav135@lsu.edu; sung@math.lsu.edu

M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations VII*, Lecture Notes in Computational Science and Engineering 100,
DOI 10.1007/978-3-319-06898-5_2

By the standard theory [21], it is easy to see that (2) characterizes the unique solution of the following obstacle problem for simply supported Kirchhoff plates: Find $u \in K$ such that

$$u = \operatorname*{argmin}_{v \in K} \left[\frac{1}{2} a(v, v) - (f, v) \right].$$

By the assumptions on f, ψ_1 and ψ_2 and according to the regularity results in [12, 15, 16] for fourth order obstacle problems, the solution u of (2) belongs to $H^3_{loc}(\Omega) \cap C^2(\Omega)$. The assumption (1) implies that the constraints are inactive near $\partial\Omega$ and hence (2) reduces to the biharmonic equation near $\partial\Omega$. It then follows from elliptic regularity theory for the biharmonic equation [3] that there exists $\alpha \in (0, 1]$ (determined by the interior angles of Ω) such that $u \in H^{2+\alpha}(\mathcal{N})$ in a neighborhood $\mathcal{N}$ of $\partial\Omega$ disjoint from the active set. Thus globally we have

$$u \in H^{2+\alpha}(\Omega). \tag{3}$$

We shall refer to α as the index of elliptic regularity for the obstacle problem.

The main difficulty in the numerical analysis of the obstacle problem (2) is due to the fact that the solution u does not have full elliptic regularity, i.e. $u \notin H^4_{loc}(\Omega)$ even for smooth data [12,15,16]. In contrast, for second order problems the solutions have H^2 regularity under appropriate assumptions [9, 20], which played a key role in the convergence analysis of the numerical methods [10, 11, 14]. This difficulty was circumvented in [8] through an auxiliary obstacle problem that connects the continuous and discrete problems. This approach does not require the full elliptic regularity of the solution and thus makes the analysis tractable.

Following the approach of [8], a partition of unity method (PUM) was presented in [5] for a class of fourth order elliptic variational inequalities posed on convex domains that include the displacement obstacle problem of simply supported Kirchhoff plates and certain distributed optimal control problems with pointwise state constraints as special cases. It was shown in [5] that the optimal $\mathcal{O}(h)$ convergence can be attained with the proper choice of enrichment functions. The goal of this paper is to extend the results for the obstacle problem for simply supported Kirchhoff plates to general polygonal domains. (The optimal control problem, which involves a different function space when the polygonal domain is nonconvex, will not be considered here.)

The choice of the enrichment functions for a general polygonal domain is dictated by the representation of the solution u as the sum of a regular part and a singular part. Let p_ℓ, $1 \le \ell \le M$, be the corners of Ω where the interior angles ω_ℓ satisfy $\pi/2 < \omega_\ell < \pi$, and p_ℓ, $M+1 \le \ell \le N$, be the corners of Ω whose interior angles satisfy $\pi < \omega_\ell < 2\pi$. Near the corners of Ω whose interior angles satisfy $0 < \omega_\ell \le \pi/2$, u is sufficiently smooth and enrichment by singular functions is not needed. Let (r_ℓ, θ_ℓ) be the polar coordinates associated with the corner p_ℓ such that the two edges of Ω emanating from p_ℓ are given by $\theta_\ell = 0$ and $\theta_\ell = \omega_\ell$. Let $\chi(r)$ be any smooth cut-off function that equals 1 near $r = 0$ and vanishes for

$r \geq \epsilon$ $(0 < \epsilon \ll 1)$. We have a singular function representation [3] for the solution u of (2):

$$u = u_R + \sum_{\ell=1}^{M} \kappa_l \xi_\ell \chi(r_\ell) + \sum_{\ell=M+1}^{N} (\lambda_\ell \rho_\ell + \mu_\ell \zeta_\ell)\chi(r_\ell), \tag{4}$$

where the regular part u_R belongs to $H^3(\Omega) \cap H_0^1(\Omega)$,

$$\xi_\ell(r_\ell, \theta_\ell) = r_\ell^{(\pi/\omega_\ell)} \sin((\pi/\omega_\ell)\theta_\ell) \tag{5}$$

is the singular function associated with the corner p_ℓ such that $\pi/2 < \omega_\ell < \pi$, and

$$\rho_\ell(r_\ell, \theta_\ell) = r_\ell^{(2\pi/\omega_\ell)} \sin((2\pi/\omega_\ell)\theta_\ell) \tag{6}$$

$$\zeta_\ell(r_\ell, \theta_\ell) = r_\ell^{2-(\pi/\omega_\ell)} \sin((\pi/\omega_\ell)\theta_\ell) \tag{7}$$

are the two singular functions associated with the corner p_ℓ such that $\pi < \omega_\ell < 2\pi$. The enrichment functions in our PUM will involve the singular functions in (5)–(7).

The rest of the paper is organized as follows. The partition of unity method is introduced in Sect. 2, and the convergence analysis is carried out in Sect. 3. Numerical examples that confirm the theoretical results and illustrate the performance of the method are presented in Sect. 4. Conclusions and future work are discussed in Sect. 5. From this point on, unless otherwise stated, C with or without subscripts represents a generic constant independent of any mesh parameters.

2 A Partition of Unity Method

We begin with the construction of the partition of unity, followed by the construction of the approximation space and the definition of the discrete problem.

2.1 Partition of Unity

We will use the partition of unity in [5,6,13,24], whose construction is recalled here for the convenience of the readers.

Let ϕ be the C^1 piecewise polynomial function defined by

$$\phi(x) = \begin{cases} \phi^L(x) = (1+x)^2(1-2x) & \text{if } x \in [-1, 0], \\ \phi^R(x) = (1-x)^2(1+2x) & \text{if } x \in [0, 1], \\ 0 & \text{if } |x| \geq 1, \end{cases}$$

where

$$\phi^L(x-1)+\phi^R(x)=1 \qquad \text{for} \quad 0\le x\le 1.$$

We define a flat-top function ψ_δ by

$$\psi_\delta(x)=\begin{cases}\phi^L\left(\frac{x-(-1+\delta)}{2\delta}\right) & \text{if } x\in[-1-\delta,-1+\delta],\\ 1 & \text{if } x\in[-1+\delta,1-\delta],\\ \phi^R\left(\frac{x-(1-\delta)}{2\delta}\right) & \text{if } x\in[1-\delta,1+\delta],\\ 0 & \text{if } x\notin[-1-\delta,1+\delta].\end{cases}$$

Here the parameter $\delta\in(0,1)$ is a small number that controls the width of the flat-top part of this function where $\psi_\delta=1$. This function is a special case of the C^r piecewise polynomial flat-top functions in [25].

For simplicity and concreteness we will only present the construction of a flat-top partition of unity for an L-shaped domain based on the flat-top function ψ_δ. A similar construction can be carried out for general polygons as in [13].

Consider an L-shaped domain Ω that is the union of two rectangles $(-L,L)\times(0,L)$ and $(-L,0)\times(-L,0]$, where L is a positive number. For some small positive numbers $\gamma_1,\gamma_2\ll L$, let Ω_γ be the union of $(-L-\gamma_1,L+\gamma_1)\times(-\gamma_2,L+\gamma_2)$ and $(-L-\gamma_1,\gamma_1)\times(-L-\gamma_2,-\gamma_2]$. It is the larger L-shaped domain formed by extending Ω by an amount of γ_i in the x_i direction (cf. Fig. 1).

We subdivide Ω_γ into three rectangles where the re-entrant corner of Ω_γ is a common vertex. This corresponds to the 0-th level partition Λ_0. The k-th level partition Λ_k is then obtained by taking each rectangle in Λ_{k-1} and subdividing it into 4^k congruent rectangles so that $|\Lambda_k|=3(4^k)$. This procedure results in a quasi-uniform background mesh (cf. Fig. 1). The mesh size h_k is defined to be the largest of the sides of the rectangles in Λ_k.

We assume that the numbers

$$\delta_\ell=\gamma_\ell/(h/2) \quad (\ell=1,2)$$

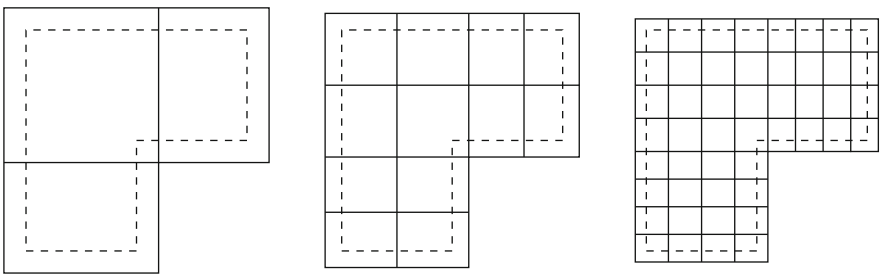


Fig. 1 This figure depicts the first three levels of refinement of the L-shaped domain represented by the *dashed lines*. Λ_0 (resp. Λ_1, and Λ_2) is on the *left* (resp. *middle* and *right*)

belong to the interval $[\beta_1, \beta_2]$, where β_1 and β_2 are constants that satisfy

$$0 < \beta_1 < \beta_2 < 1. \tag{8}$$

Consider the i-th rectangle $R_i = (a, b) \times (c, d)$ in Λ_k. The patch $\Omega_i = (a - \gamma_1, b + \gamma_1) \times (c - \gamma_2, d + \gamma_2)$ is where the i-th partition of unity function

$$\varphi_i(x_1, x_2) = \psi_{\delta_1}\left(\frac{2x_1 - (b + a)}{b - a}\right)\psi_{\delta_2}\left(\frac{2x_2 - (d + c)}{d - c}\right)$$

is defined. It is easy to check that $\{\varphi_i\}_{i=1}^n$ is a partition of unity over Ω, where n is the number of patches in Λ_k.

Note that the flat-top region $\Omega_i^{\text{flat}} = \{x \in \Omega_i : \varphi(x) = 1\}$ associated with φ_i is the rectangle $(a + \gamma_1, b - \gamma_1) \times (c + \gamma_2, d - \gamma_2)$ obtained from Ω_i by contracting an amount of γ_i in the x_i direction.

The idea of flat-top PU was introduced by Griebel and Schweitzer in [17]. One of the key benefits of using the flat-top PU is that a basis for the global approximation space can be defined in terms of the bases of the local approximation spaces. For the obstacle problem of simply supported Kirchhoff plates, it also allows the construction of a global approximation space that satisfies the Kronecker delta property, which simplifies the enforcement of the obstacles in the discrete problem.

2.2 Local Approximation Spaces

The local approximation spaces will be denoted by V_i. It is the sum of a polynomial part V_i^{poly} and a singular part V_i^{sing}.

2.2.1 Polynomial Spaces

For a general polygonal domain, the local polynomial spaces V_i^{poly} is either the quadratic polynomial space $\mathbb{P}_2$ or the biquadratic polynomial space $\mathbb{Q}_2$, and we use Lagrange interpolation polynomials as the basis of V_i^{poly}. The interpolation nodes are carefully placed to ensure one can enforce the homogeneous Dirichlet boundary condition on $\partial\Omega$ exactly. The interpolation nodes are placed by checking the shape of $\Omega_i \cap \Omega$. This can be done manually or by creating a subroutine to catch which edges of $\Omega_i \cap \Omega$ coincide with $\partial\Omega$ and following the strategy presented in [13]. We also place the interpolation nodes inside the flat-top regions so that the global approximation space has the Kronecker delta property, resulting in discrete problems with simple box constraints.

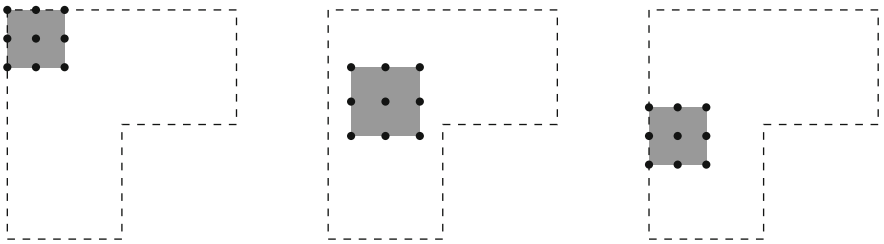


Fig. 2 This figure shows the placement of interpolation nodes that enforce the homogeneous Dirichlet boundary condition exactly. Three representative patches are depicted where the *shaded rectangles* are flat-top regions

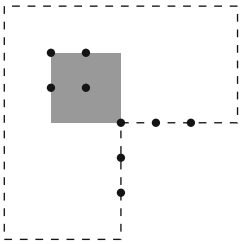


Fig. 3 This figure depicts the patch where the lower-right corner of the flat-top region is at the re-entrant corner. The flat-top region is represented by the *shaded rectangle* and the *dots* represent the modified interpolation nodes that enforce the Dirichlet boundary condition

Here for simplicity we only provide details for an L-shaped domain, where $V_i^{\text{poly}} = \mathbb{Q}_2$ for all i. The placement of the interpolation nodes for most of the patches are illustrated in Fig. 2.

There is one exceptional patch where the lower-right corner of the flat-top region resides at the re-entrant corner (cf. Fig. 3). For this patch, we use a different polynomial basis corresponding to the modified interpolation nodes that enforce the homogeneous Dirichlet boundary condition exactly. Note that it is also possible to use $\mathbb{P}_2$ as the local approximation space for this patch (cf. [13]).

2.2.2 Enrichment Functions

We follow the ideas in [1,19] to construct the enrichment functions from the singular functions in (5)–(7).

For each singular vertex p_ℓ ($1 \le \ell \le N$) of Ω we fix a radius d_ℓ (independent of any mesh parameters) and take $X_\ell = \{x \in \Omega : |x - p_\ell| < d_\ell\}$ to be the enrichment zone for p_ℓ. If $\Omega_i \cap X_\ell \neq \emptyset$, then the patch Ω_i is said to be enriched by p_ℓ.

Remark 1. We assume the radii $d_1, \ldots, d_N$ are small enough so that each patch will intersect at most one enrichment zone when h is small. For the L-shaped domain, there is only one enrichment zone associated with the re-entrant corner.

For each patch Ω_i enriched by a convex singular corner p_ℓ (where $\pi/2 < \omega_\ell < \pi$), we use the enrichment function defined by

$$\xi_{i,\ell} = \pi_i \xi_\ell - \xi_\ell,$$

where π_i is the nodal interpolation operator for the patch Ω_i; and for each patch Ω_i enriched by a nonconvex singular corner p_ℓ (where $\pi < \omega_\ell < 2\pi$), we use the enrichment functions defined by

$$\rho_{i,\ell} = \pi_i \rho_\ell - \rho_\ell \quad \text{and} \quad \zeta_{i,\ell} = \pi_i \zeta_\ell - \zeta_\ell.$$

Remark 2. These enrichment functions are more stable than the singular functions in (5)–(6) in the sense that the condition number of the resulting stiffness matrix is much smaller [1]. Moreover, since these enrichment functions vanish locally at the interpolation nodes, the polynomial part of the global approximation space retains the Kronecker delta property and the discrete problem (cf. (9) below) becomes a quadratic programming problem with box constraints on the polynomial part of the global approximation space.

Let $I_i^c = \{1 \le \ell \le M : \Omega_i \cap X_\ell \ne \emptyset\}$ (resp. $I_i^n = \{M+1 \le \ell \le N : \Omega_i \cap X_\ell \ne \emptyset\}$) be the index set of the singular convex corners (resp. the nonconvex singular corners) that enrich the patch Ω_i. We define the space V_i^{sing} of enrichment functions for the patch Ω_i to be

$$V_i^{\text{sing}} = \sum_{\ell \in I_i^c} \text{span}\{\xi_{i,\ell}\} + \sum_{\ell \in I_i^n} \text{span}\{\rho_{i,\ell}, \zeta_{i,\ell}\}.$$

Note that $V_i^{\text{sing}} = \{0\}$ if the patch Ω_i is not enriched by any singular corners.

For each patch Ω_i the local approximation space is then given by

$$V_i = V_i^{\text{poly}} + V_i^{\text{sing}}.$$

2.3 The Global Approximation Space and the Discrete Problem

The global approximation space is constructed from the local approximation in the standard manner:

$$V_h = \sum_{i=1}^{n} \varphi_i V_i,$$

where n is the total number of patches for the partition of unity.

Let $\mathcal{V}_h$ be the set of all interpolation nodes for V_h inside Ω. We define the discrete convex set K_h by

$$K_h = \{v \in V_h : \psi_1(z) \le v(z) \le \psi_2(z) \quad z \in \mathcal{V}_h\}.$$

The discrete obstacle problem is to find $u_h \in K_h$ such that

$$a(u_h, v - u_h) \ge (f, v - u_h) \quad \forall v \in K_h. \tag{9}$$

2.4 Approximation Properties

On a patch Ω_i outside of the enrichment zones, the solution u of the obstacle problem belongs to $\in H^3(\Omega_i)$ and we can show that there exists a function $u_i^* \in V_i$ such that

$$\sum_{k=0}^{2} h^k |u - u_i^*|_{H^k(\Omega_i)} \le Ch^3 |u|_{H^3(\Omega_i)} \tag{10}$$

using the local nodal interpolation operator [7] and the Bramble-Hilbert lemma [4].

On a patch Ω_i enriched by a nonconvex singular corner p_ℓ (where $\pi < \omega_\ell < 2\pi$), the solution u of the obstacle problem, in view of (4) and Remark 1, can be written as

$$u = u_{R,\ell} + \lambda_\ell \rho_\ell + \mu_\ell \zeta_\ell,$$

where $u_{R,\ell} \in H^3(\Omega)$. Since the restrictions of ρ_ℓ and ζ_ℓ to Ω_i belong to V_i by construction, the approximation of u is reduced to the approximation of $u_{R,\ell}$ and hence there exists, by local interpolation operator and the Bramble-Hilbert lemma, a function $u_i^* \in V_i$ such that

$$\sum_{k=0}^{2} h^k |u - u_i^*|_{H^k(\Omega_i)} \le Ch^3 |u_{R,\ell}|_{H^3(\Omega_i)}. \tag{11}$$

Similarly, on a patch Ω_i enriched by a convex singular corner p_ℓ (where $\pi/2 < \omega_\ell < \pi$), the solution u of the obstacle problem can be written as

$$u = u_{R,\ell} + \kappa_\ell \xi_\ell$$

where $u_{R,\ell} \in H^3(\Omega)$. Again the estimate (11) is valid for a function $v_i \in V_i$.

We can now combine the local estimates (10) and (11) with the estimates for the PU functions (cf. [22, Theorem 2.1] and [25, Theorem 6.1]) to obtain the following estimate for the global approximation error:

$$\sum_{k=0}^{2} h^k |u - u_h^*|_{H^k(\Omega)} \leq Ch^3, \tag{12}$$

where $u_h^* = \sum_{i=1}^{n} \varphi_i u_i^*$ and C depends on the constant β_1 in (8).

Remark 3. The inclusion of the singular functions is responsible for the optimal approximation error estimate (12). If these singular functions were not present, then in general one can only find a $u_h^* \in V_h$ such that

$$\sum_{k=0}^{2} h^k |u - u_h^*|_{H^k(\Omega)} \leq Ch^{2+\alpha}, \tag{13}$$

where α is the index of elliptic regularity in (3).

3 Convergence Analysis

We will use the energy norm $\|v\|_a = \sqrt{a(v,v)}$ in the error analysis. Since the energy norm is equivalent to the $H^2(\Omega)$ norm (by a Poincaré-Friedrichs inequality), it follows immediately from (12) that there exists a function $v_h^* \in V_h$ such that

$$\|u - u_h^*\|_a \leq Ch. \tag{14}$$

3.1 Integration by Parts

We now present an integration by parts formula that will be useful in the convergence analysis.

First we observe that the explicit formulas (5)–(7) imply

$$\Delta\Big(u_R + \sum_{\ell=1}^{M} \kappa_\ell \xi_\ell \chi(r_\ell) + \sum_{l=M+1}^{N} \lambda_\ell \rho_\ell \chi(r_\ell)\Big) \in H_0^1(\Omega)$$

and

$$\Delta^2\Big(\sum_{\ell=M+1}^{N} \mu_\ell \zeta_\ell \chi(r_\ell)\Big) \in L_2(\Omega).$$

Let $v \in H^2(\Omega) \cap H_0^1(\Omega)$. The following integration by parts formula is standard:

$$\int_\Omega \Delta\Big(u_R + \sum_{\ell=1}^{M} \kappa_\ell \xi_\ell \chi(r_\ell) + \sum_{\ell=M+1}^{N} \lambda_\ell \rho_\ell \chi(r_\ell)\Big) \Delta v \mathrm{d}x$$
$$= -\int_\Omega \nabla\Big[\Delta\Big(u_R + \sum_{\ell=1}^{M} \kappa_\ell \xi_\ell \chi(r_\ell) + \sum_{\ell=M+1}^{N} \lambda_\ell \rho_\ell \chi(r_\ell)\Big)\Big] \cdot \nabla v \mathrm{d}x.$$

There is another integration by parts formula

$$\int_\Omega \Delta\Big(\sum_{\ell=M+1}^{N} \mu_\ell \zeta_\ell \chi(r_\ell)\Big) \Delta v \mathrm{d}x = \int_\Omega \Delta^2\Big(\sum_{\ell=M+1}^{N} \mu_\ell \zeta_\ell \chi(r_\ell)\Big) v \mathrm{d}x,$$

that can be derived as follows. First we may assume $v \in C^2(\bar{\Omega}) \cap H_0^1(\Omega)$ since $C^2(\bar{\Omega}) \cap H_0^1(\Omega)$ is a dense subset of $H^2(\Omega) \cap H_0^1(\Omega)$ [18, Theorem 1.6.2]. Let Ω_ϵ be the subset of Ω obtained by removing a disc of radius ϵ from all the vertices p_ℓ of Ω such that $\pi < p_\ell < 2\pi$. Then we have

$$\int_\Omega \Delta\Big(\sum_{\ell=M+1}^{N} \mu_\ell \zeta_\ell \chi(r_\ell)\Big) \Delta v \mathrm{d}x = \lim_{\epsilon \downarrow 0} \int_{\Omega_\epsilon} \Delta\Big(\sum_{\ell=M+1}^{N} \mu_\ell \zeta_\ell \chi(r_\ell)\Big) (\Delta v) \mathrm{d}x$$
$$= \lim_{\epsilon \downarrow 0} \Bigg(\int_{\partial\Omega_\epsilon} \Bigg[\Delta\Big(\sum_{\ell=M+1}^{N} \mu_\ell \zeta_\ell \chi(r_\ell)\Big) \frac{\partial v}{\partial n} - \frac{\partial}{\partial n} \Delta\Big(\sum_{\ell=M+1}^{N} \mu_\ell \zeta_\ell \chi(r_\ell)\Big) v \Bigg] \mathrm{d}s$$
$$+ \int_{\Omega_\epsilon} \Big(\Delta^2 \sum_{\ell=M+1}^{N} \mu_\ell \zeta_\ell \chi(r_\ell)\Big) v \mathrm{d}x \Bigg)$$
$$= \int_\Omega \Delta^2\Big(\sum_{\ell=M+1}^{N} \mu_\ell \zeta_\ell \chi(r_\ell)\Big) v \mathrm{d}x.$$

Here the limit

$$\lim_{\epsilon \downarrow 0} \int_{\partial\Omega_\epsilon} \Bigg[\Delta\Big(\sum_{\ell=M+1}^{N} \mu_\ell \zeta_\ell \chi(r_\ell)\Big) \frac{\partial v}{\partial n} - \frac{\partial}{\partial n} \Delta\Big(\sum_{\ell=M+1}^{N} \mu_\ell \zeta_\ell \chi(r_\ell)\Big) v \Bigg] \mathrm{d}s = 0$$

follows from a direct computation that uses the explicit formula for ζ_ℓ in (7) and the fact that $v \in C^2(\bar{\Omega})$ vanishes on $\partial\Omega$.

Putting (4) and the two integration by parts formulas together, we have

$$\int_\Omega \Delta u \Delta v \mathrm{d}x = -\int_\Omega \nabla\Big[\Delta\Big(u_R + \sum_{\ell=1}^{M} \kappa_\ell \xi_\ell \chi(r_\ell) + \sum_{\ell=M+1}^{N} \lambda_\ell \rho_\ell \chi(r_\ell))\Big)\Big] \cdot \nabla v \mathrm{d}x + \int_\Omega \Delta^2\Big(\sum_{\ell=M+1}^{N} \mu_\ell \zeta_\ell \chi(r_\ell)\Big) v \mathrm{d}x. \tag{15}$$

3.2 A Preliminary Estimate

Let $u_h^* \in V_h$ be the function in (12) and (14). We have, by the discrete variational inequality (9),

$$\begin{aligned} \|u - u_h\|_a^2 &= a(u - u_h, u - u_h^*) + a(u - u_h, u_h^* - u_h) \\ &\le Ch\|u - u_h\|_a + a(u, u_h^* - u_h) - (f, u_h^* - u_h) \\ &\le Ch\|u - u_h\|_a + a(u, u - u_h) - (f, u - u_h) \\ &\quad + a(u, u_h^* - u) - (f, u_h^* - u). \end{aligned} \tag{16}$$

Using (15), (12) and the Cauchy-Schwarz inequality, we find

$$\begin{aligned} |a(u, u_h^* - u)| &= \Big| -\int_\Omega \nabla\Big[\Delta\Big(u_R + \sum_{\ell=1}^{M} \kappa_\ell \xi_\ell \chi(r_\ell) \\ &\qquad + \sum_{\ell=M+1}^{N} \lambda_\ell \rho_\ell \chi(r_\ell)\Big)\Big] \cdot \nabla(u_h^* - u) \mathrm{d}x \\ &\qquad + \int_\Omega \Delta^2\Big(\sum_{\ell=M+1}^{N} \mu_\ell \zeta_\ell \chi(r_\ell)\Big)(u_h^* - u) \mathrm{d}x \Big| \\ &\le C\,|u_h^* - u|_{H^1(\Omega)} + C\,\|u_h^* - u\|_{L_2(\Omega)} \\ &\le Ch^2. \end{aligned} \tag{17}$$

Furthermore it follows from (12) that

$$|(f, u_h^* - u)| \le C\,\|u_h^* - u\|_{L_2(\Omega)} \le Ch^3. \tag{18}$$

Combining (16)–(18) with Young's inequality, we have

$$\|u - u_h\|_a^2 \le Ch^2 + Ch\|u - u_h\|_a + a(u, u - u_h) - (f, u - u_h)$$

$$\leq Ch^2 + \frac{1}{2}\|u - u_h\|_a^2 + a(u, u - u_h) - (f, u - u_h),$$

which implies

$$\|u - u_h\|_a^2 \leq Ch^2 + 2[a(u, u - u_h) - (f, u - u_h)]. \tag{19}$$

3.3 An Auxiliary Problem

In order to obtain a bound for the expression on the right hand side of (19), we need a connection between (2) and (9). We can establish such a connection by considering the following auxiliary variational inequality: Find $\tilde{u}_h \in \tilde{K}_h$ such that

$$a(\tilde{u}_h, v - \tilde{u}_h) \geq (f, v - \tilde{u}_h) \quad \forall v \in \tilde{K}_h, \tag{20}$$

where

$$\tilde{K}_h = \{v \in H^2(\Omega) \cap H_0^1(\Omega) : \psi_1(z) \leq v(z) \leq \psi_2(z) \quad \forall z \in \mathcal{V}_h\}.$$

Since both K and K_h are subsets of $\tilde{K}_h$, the variational inequality (20) provides a link between (2) and (9). By the arguments in [8], there exists a function $\hat{u}_h \in K$ such that

$$\|\tilde{u}_h - \hat{u}_h\|_a \leq Ch^2, \tag{21}$$

which implies

$$\|u - \tilde{u}_h\|_a \leq Ch. \tag{22}$$

We can now obtain a bound for the expression on the right hand side of (19) as follows.

$$\begin{aligned}
a(u, u - u_h) &= a(u, u - \hat{u}_h) + a(u, \hat{u}_h - u_h) \\
&\leq (f, u - \hat{u}_h) + a(u, \hat{u}_h - \tilde{u}_h) + a(u, \tilde{u}_h - u_h) \\
&\leq (f, u - \tilde{u}_h) + (f, \tilde{u}_h - \hat{u}_h) + a(u, \tilde{u}_h - u_h) + Ch^2 \\
&\leq (f, u - \tilde{u}_h) + a(\tilde{u}_h, \tilde{u}_h - u_h) + a(u - \tilde{u}_h, \tilde{u}_h - u_h) + Ch^2 \\
&\leq (f, u - \tilde{u}_h) + (f, \tilde{u}_h - u_h) + a(u - \tilde{u}_h, \tilde{u}_h - u + u - u_h) + Ch^2 \\
&\leq (f, u - u_h) - a(u - \tilde{u}_h, u - \tilde{u}_h) + a(u - \tilde{u}_h, u - u_h) + Ch^2 \\
&\leq (f, u - u_h) + Ch\|u - u_h\|_a + Ch^2,
\end{aligned} \tag{23}$$

where we have used the variational inequality (2) (with $v = \hat{u}_h$), the variational inequality (20) (with $v = u_h$) and the estimates (21)–(22).

3.4 Error Estimates

The following theorem is the main result of the paper.

Theorem 1. *There exists a positive constant C independent of h such that*

$$\|u - u_h\|_a \leq Ch. \tag{24}$$

Proof. Combining (19) and (23) with Young's inequality, we have

$$\|u - u_h\|_a^2 \leq Ch^2 + Ch\|u - u_h\|_a \leq Ch^2 + \frac{1}{2}\|u - u_h\|_a^2. \qquad \square$$

The following corollary is an immediate consequence of Theorem 1 and the Sobolev embedding theorem.

Corollary 1. *There exists a constant C independent of h such that*

$$\|u - u_h\|_\infty \leq Ch.$$

Remark 4. Since the L_∞ norm is weaker than the H^2 norm, the error estimate in Corollary 1 is not sharp. This is observed in the numerical experiments.

Remark 5. The constant C depends on the constant β_1 in (8).

Remark 6. Without the singular enrichment functions the energy norm error and the L_∞ error are bounded by Ch^α, where α is the index of elliptic regularity in (3).

4 Numerical Examples

The computational domain Ω in the following examples is the L-shaped domain formed by the union of the rectangles $(-0.5, 0.5)\times(0, 0.5)$ and $(-0.5, 0)\times(-0.5, 0]$. We solve each problem on seven levels of refinement (cf. Fig. 1). The mesh parameters on the j-th level are given by $h_j = (2^{j+1}-2/3)^{-1}$ and $\gamma_1 = \gamma_2 = h_j/6$. The radius of the enrichment zone around the re-entrant corner is taken to be 0.1 The solution of the j-th level discrete problem is denoted by u_j.

The discrete problems are solved by a primal dual active set strategy [2] and numerical integration is performed by using the generalized Duffy transform [23] with parameter $\beta_D = 3$ and 20 Gaussian quadrature points in the regions containing

the singular enrichment functions. Standard Gauss quadrature is used for the polynomials.

For comparison we also solve the variational inequalities by the PUM method without enrichment. Note that the convergence of the unenriched PUM is determined by the index of elliptic regularity α for the L-shaped domain, which can be any number less than $1/3$.

4.1 A Boundary Value Problem

As a first numerical example we apply our method to a problem with a known solution. We take $\psi_1 = -\infty$ and $\psi_2 = +\infty$ so that the variational inequality (2) reduces to a variational equality (boundary value problem) and the discrete convex set $K_h = V_h$. The discrete problem is to find $u \in V_h$ such that

$$a(u, v) = (f, v) \quad \forall\, v \in V_h.$$

We take

$$u = (r^2 \cos^2\theta - 0.25)^3 (r^2 \sin^2\theta - 0.25)^3 (r^z \sin(z\theta) - r^z \sin((z-2)\theta))$$

to be the exact solution, where $z = 4/3$ and (r, θ) are the polar coordinates centered at the origin.

The error corresponding to the jth level is $e_j = u_j - u$. We estimate the magnitude of e_j in the energy norm and the maximum norm by the following formulas:

$$\|e_j\|_a = \frac{(|u|^2_{H^2(\Omega)} - |u_j|^2_{H^2(\Omega)})^{1/2}}{|u|_{H^2(\Omega)}} \quad \text{and} \quad \|e_j\|_\infty = \frac{\max_{p\in\mathcal{N}_j} |e_j(p)|}{\max_{p\in\mathcal{N}_j} |u(p)|},$$

where $\mathcal{N}_j$ is the set of all the nodes corresponding to the jth level and $|u|_{H^2(\Omega)} \approx 7.423189736575737 \times 10^{-4}$ was computed using Mathematica. The convergence rate β for each norm is computed by

$$\beta = \log(\|e_{j-1}\|/\|e_j\|)/\log(h_{j-1}/h_j).$$

The numerical results in Table 1 demonstrate that the enriched PUM is much more efficient in solving this problem. The magnitude of the L_∞ error is $O(h^2)$, better than the $O(h)$ magnitude stated in Corollary 1.

Table 1 Results of the PUM applied to the boundary value problem. The results from columns 2 to 5 correspond to the PUM without enrichment and the results from columns 6 to 9 correspond to the PUM with enrichment

	Without enrichment				With enrichment			
j	$\|e_j\|_a$	Rate	$\|e_j\|_\infty$	Rate	$\|e_j\|_a/\|u\|_a$	Rate	$\|e_j\|_\infty$	Rate
1	$8.89E-1$	–	$9.35E-1$	–	$8.33E-1$	–	$6.90E-1$	–
2	$7.96E-1$	0.14	$8.90E-1$	0.06	$3.65E-1$	1.05	$9.14E-2$	2.56
3	$6.06E-1$	0.37	$5.18E-1$	0.73	$1.81E-1$	0.95	$1.74E-2$	2.25
4	$4.46E-1$	0.43	$2.90E-1$	0.81	$9.24E-2$	0.94	$4.22E-3$	1.98
5	$3.36E-1$	0.40	$1.68E-1$	0.78	$4.68E-2$	0.97	$1.02E-3$	2.02
6	$2.58E-1$	0.38	$1.00E-1$	0.74	$2.36E-2$	0.98	$2.53E-4$	2.00
7	$2.01E-1$	0.36	$6.11E-2$	0.71	$1.19E-2$	0.99	$6.32E-5$	1.99

4.2 Obstacle Problems

The next two examples are obstacle problems. Since the analytic solutions are not known for these examples, the errors are computed through successive approximations.

Let $\tilde{e}_j$ be defined by

$$\tilde{e}_j = u_j - \Pi_j u_{j-1},$$

where $\Pi_j u_{j-1}$ is the nodal interpolant of u_{j-1} in V_j. The magnitudes of the errors are then estimated by the following formulas:

$$\|\tilde{e}_j\|_a = \frac{|\tilde{e}_j|_{H^2(\Omega)}}{|u_7|_{H^2(\Omega)}} \qquad \text{and} \qquad \|\tilde{e}_j\|_\infty = \max_{p\in\mathcal{N}_j} |\tilde{e}_j(p)|.$$

In each of these examples we perform the same convergence tests as the ones for the boundary value problem. In addition we also plot the discrete coincidence sets $C_{i,7}$ at the finest level for both the enriched and unenriched methods. These are computed by

$$C_{i,j} = \{x \in \mathcal{N}_j : |x - \psi_i| \le \|\tilde{e}_j\|_\infty\}.$$

It is believed that the plots from the enriched PUM are more accurate approximations of the true coincidence sets.

4.2.1 A One-Obstacle Problem

In this example we solve (2) with $f = 0$,

Table 2 Results of the PUM applied to the one-obstacle problem. The results from columns 2 to 5 correspond to the PUM without enrichment and the results from columns 6 to 9 correspond to the enriched PUM with enrichment

	Without enrichment				With enrichment			
j	$\|\tilde{e}_j\|_a$	Rate	$\|\tilde{e}_j\|_\infty$	Rate	$\|\tilde{e}_j\|_a$	Rate	$\|\tilde{e}_j\|_\infty$	Rate
1	$1.77E-0$	–	$1.07E-0$	–	$1.26E-0$	–	$1.06E-0$	–
2	$1.97E-0$	−0.13	$4.86E-1$	1.00	$1.20E-0$	0.06	$1.53E-1$	2.45
3	$7.70E-1$	1.27	$1.91E-1$	1.27	$3.30E-1$	1.75	$1.93E-2$	2.81
4	$3.84E-1$	0.97	$7.20E-2$	1.36	$1.48E-1$	1.12	$3.45E-3$	2.41
5	$2.37E-1$	0.68	$3.48E-2$	1.03	$5.65E-2$	1.37	$6.77E-4$	2.32
6	$1.65E-1$	0.52	$1.89E-2$	0.87	$2.51E-2$	1.16	$1.68E-4$	2.00
7	$1.21E-1$	0.45	$1.07E-2$	0.81	$1.19E-2$	1.07	$4.06E-5$	2.04

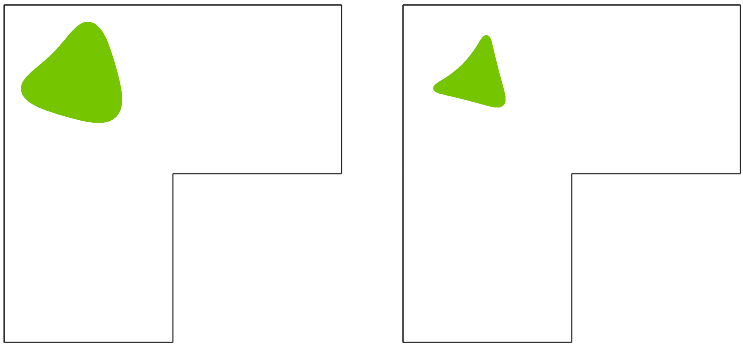


Fig. 4 Plot of the discrete coincidence sets at the finest levels. The plot on the *left* is the result using the unenriched method while the plot on the *right* is the result obtained by the enriched method. The *green dots* represent $C_{2,7}$

$$\psi_1 = -\infty \quad \text{and} \quad \psi_2(x) = \left[\left(\frac{x_1 + 0.25}{0.24}\right)^2 + \left(\frac{x_2 - 0.25}{0.24}\right)^2\right] - 1.$$

The numerical results in Table 2 demonstrate that the enriched method is more accurate than the unenriched version, and the magnitude of the L_∞ error for the enriched PUM is $O(h^2)$. The discrete coincidence sets, which have the correct symmetry, are presented in Fig. 4.

4.2.2 A Two-Obstacle Problem

In this example we take $f = 0$,

$$\psi_1 = 1 - \left[\left(\frac{x_1 + 0.25}{0.2} \right)^2 + \left(\frac{x_2 + 0.25}{0.2} \right)^2 \right]$$

and

$$\psi_2(x) = \left[\left(\frac{x_1 - 0.25}{0.2} \right)^2 + \left(\frac{x_2 - 0.25}{0.2} \right)^2 \right] - 1.$$

The numerical results in Table 3 again shows the improvement due to enrichment. The magnitude of the L_∞ error for the enriched PUM appears to be $O(h)$ but with fluctuations, which is likely due to the fact that the primal dual active set algorithm does not use a stopping criteria based on the point values. The discrete coincidence sets are presented in Fig. 5.

Table 3 Results of the PUM applied to the two-obstacle problem. The results from columns 2 to 5 correspond to the PUM without enrichment and the results from columns 6 to 9 correspond to the PUM with enrichment

	Without enrichment				With enrichment			
j	$\|\tilde{e}_j\|_a$	Rate	$\|\tilde{e}_j\|_\infty$	Rate	$\|\tilde{e}_j\|_a$	Rate	$\|\tilde{e}_j\|_\infty$	Rate
1	$1.48E-0$	–	$9.77E-1$	–	$1.18E-0$	–	$1.08E-0$	–
2	$1.69E-0$	−0.18	$2.11E-1$	2.08	$1.15E-0$	0.04	$1.01E-1$	3.21
3	$6.30E-1$	1.38	$1.15E-1$	0.85	$3.10E-1$	1.84	$3.65E-2$	1.43
4	$3.32E-1$	0.91	$5.41E-2$	1.07	$1.17E-1$	1.38	$3.85E-3$	3.20
5	$2.18E-1$	0.60	$2.58E-2$	1.06	$5.51E-2$	1.08	$9.99E-4$	1.93
6	$1.55E-1$	0.49	$1.40E-2$	0.88	$2.67E-2$	1.04	$1.44E-4$	2.78
7	$1.15E-1$	0.43	$7.95E-3$	0.81	$1.29E-2$	1.04	$5.42E-5$	1.41

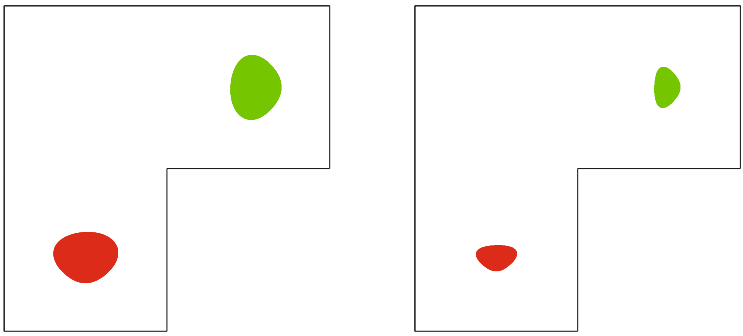


Fig. 5 Plot of the discrete coincidence sets at the finest levels. The plot on the *left* is the result using the unenriched method while the plot on the *right* is the result obtained by the enriched method. The *red dots* represent the nodes in $C_{1,7}$ while the *green dots* represent the nodes in $C_{2,7}$

5 Conclusions

We have presented a partition of unity method for the displacement obstacle problem of simply supported Kirchhoff plates on general polygons. By enriching the approximation space with appropriate singular functions, we are able to obtain optimal convergence in the energy norm. The improvement in the performance of the PUM due to enrichment is demonstrated by numerical examples.

The results in [5, 6] and this paper demonstrate that fourth order variational inequalities can be solved effectively by generalized finite element methods. The extension of the PUM to other fourth order variational inequalities and to variational inequalities of even higher order, where it may be the only feasible method, are topics of ongoing research.

Acknowledgements The work of the first and third authors was supported in part by the National Science Foundation under Grant No. DMS-13-19172. The work of the second author was supported in part by the National Science Foundation through the VIGRE Grant 07-39382.

References

1. I. Babuŝka, U. Banerjee, Stable generalized finite element method (SGFEM). Comput. Methods Appl. Mech. Eng. **201–204**, 91–111 (2012)
2. M. Bergounioux, K. Ito, K. Kunisch, Primal-dual strategy for state-constrained optimal control problems. SIAM J. Control Optim. **37**, 1176–1194 (1999) (electronic)
3. H. Blum, R. Rannacher, On the boundary value problem of the biharmonic operator on domains with angular corners. Math. Methods Appl. Sci. **2**, 556–581 (1980)
4. J.H. Bramble, S.R. Hilbert, Estimation of linear functionals on Sobolev spaces with applications to Fourier transforms and spline interpolation. SIAM J. Numer. Anal. **7**, 113–124 (1970)
5. S.C. Brenner, C.B. Davis, L.-Y. Sung, A partition of unity method for a class of variational inequalities. Comput. Methods Appl. Mech. Eng. **276**, 612–626 (2014)
6. S.C. Brenner, C.B. Davis, L.-Y. Sung, A partition of unity method for the displacement obstacle problem for clamped Kirchhoff plates. J. Comput. Appl. Math. (2013). doi: 10.1016/j.cam.2013.09.033
7. S.C. Brenner, L.R. Scott, *The Mathematical Theory of Finite Element Methods*, 3rd edn. (Springer, New York, 2008)
8. S.C. Brenner, L.-Y. Sung, Y. Zhang, Finite element methods for the displacement obstacle problem of clamped plates. Math. Comput. **81**, 1247–1262 (2012)
9. H.R. Brézis, G. Stampacchia, Sur la régularité de la solution d'inéquations elliptiques. Bull. Soc. Math. Fr. **96**, 153–180 (1968)
10. F. Brezzi, W. Hager, P.-A. Raviart, Error estimates for the finite element solution of variational inequalities. Numer. Math. **28**, 431–443 (1977)
11. F. Brezzi, W. Hager, P.-A. Raviart, Error estimates for the finite element solution of variational inequalities. II. Mixed methods. Numer. Math. **31**, 1–16 (1978/1979)
12. L.A. Caffarelli, A. Friedman, The obstacle problem for the biharmonic operator. Ann. Sc. Norm. Sup. Pisa Cl. Sci. (4) **6**, 151–184 (1979)
13. C.B. Davis, A partition of unity method with penalty for fourth order problems. J. Sci. Comput. (2013). doi: 10.1007/s10915-013-9795-8

14. R.S. Falk, Error estimates for the approximation of a class of variational inequalities. Math. Comput. **28**, 963–971 (1974)
15. J. Frehse, Zum Differenzierbarkeitsproblem bei Variationsungleichungen höherer Ordnung. Abh. Math. Sem. Univ. Hamb. **36**, 140–149 (1971)
16. J. Frehse, On the regularity of the solution of the biharmonic variational inequality. Manuscr. Math. **9**, 91–103 (1973)
17. M. Griebel, M.A. Schweitzer, A particle-partition of unity method part II: efficient cover construction and reliable integration. SIAM J. Sci. Comput. **23**, 1655–1682 (2002)
18. P. Grisvard, *Singularities in Boundary Value Problems* (Masson, Paris, 1992)
19. V. Gupta, C.A. Duarte, I. Babuŝka, U. Banerjee, A stable and optimally convergent generalized FEM (SGFEM) for linear elastic fracture mechanics. Comput. Methods Appl. Mech. Eng. **266**, 23–39 (2013)
20. H. Lewy, G. Stampacchia, On the regularity of the solution of a variational inequality. Commun. Pure Appl. Math. **22**, 153–188 (1969)
21. J.-L. Lions, G. Stampacchia, Variational inequalities. Commun. Pure Appl. Math. **20**, 493–519 (1967)
22. J.M. Melenk, I. Babuška, The partition of unity finite element method: basic theory and applications. Comput. Methods Appl. Mech. Eng. **139**, 289–314 (1996)
23. S.E. Mousavi, N. Sukumar, Generalized Duffy transformation for integrating vertex singularities. Comput. Mech. **45**(2–3), 127–140 (2010)
24. H.-S. Oh, C. Davis, J.W. Jeong, Meshfree particle methods for thin plates. Comput. Methods Appl. Mech. Eng. **209–212**, 156–171 (2012)
25. H.-S. Oh, J.G. Kim, W.-T. Hong, The piecewise polynomial partition of unity functions for the generalized finite element methods. Comput. Methods Appl. Mech. Eng. **197**, 3702–3711 (2008)

Particle Method Modeling of Nonlocal Multiresolution Continua

Zili Dai, Miguel A. Bessa, Shaofan Li, and Wing Kam Liu

Abstract This work is concerned with the application of two different particle methods, the state-based peridynamics and the reproducing kernel particle method, to model the nonlocal multiresolution continuum. It is shown that both methods lead to the same results in uniform grids, and that they can offer an alternative to the finite element method based multiscale analysis. The equivalence between the two methods is explained in a comparison study, which shows that the state-based peridynamics may have better computational efficiency, but the RKPM synchronized derivative approach may have the possibility of faster convergence.

Keywords State-based peridynamics • Reproducing kernel particle method • Multiresolution continuum • Micromorphic continuum

1 Introduction

The aim of this article is twofold: present the recent findings that connect State-based Peridynamics [25] and the Reproducing Kernel Particle Method (RKPM) [18]; and apply these meshfree particle methods to multiscale analysis of multiresolution continua [20]. The first part provides insight on the underlying numerics of the two meshfree particle methods by concluding that the state-based peridynamics is equivalent to RKPM in uniform grids away from the boundaries, with the advantage that it has better computational efficiency. This conclusion motivates the application of state-based peridynamics to multiresolution analysis,

Z. Dai • M.A. Bessa • W.K. Liu (✉)
Department of Mechanical Engineering, Northwestern University, 2145 Sheridan Rd., Evanston, IL 60208, USA
e-mail: w-liu@northwestern.edu

S. Li
Department of Civil and Environmental Engineering, University of California, Berkeley, CA 94720, USA
e-mail: li@ce.berkeley.edu

M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations VII*, Lecture Notes in Computational Science and Engineering 100,
DOI 10.1007/978-3-319-06898-5_3

since the mesh-based implementation of multiresolution analysis, e.g. in finite element implementation, is significantly inefficient [20, 27, 28].

We will start by connecting the two methods in the next section with explanations and discussions. The following section presents the multiresolution discretizations for these methods and a one-dimensional illustration.

2 Connecting Peridynamics and the Reproducing Kernel Particle Method

Recently, Bessa et al. [3] have found that there is a direct connection and resemblance between the state-based peridynamics and the reproducing kernel particle method, if it is used as an a nonlocal integral formulation for instance as a modified smooth particle hydrodynamics. The discovery by Bessa et al. [3] of the connection between state-based peridynamics and classical meshfree methods is significantly important, because Peridynamics has a low computational cost when compared to the Element-free Galerkin (EFG) method [2] or the Reproducing Kernel Particle Method (RKPM) [17, 18], when it is used as a meshfree interpolant in the solution of the Galerkin weak formation for partial differential equations, while methods like EFG or weak form RKPM provide a framework to solve partial differential equations that can have higher than second order derivatives. Understanding the link between these methods may lead to advancement of particle methods.

2.1 State-Based Peridynamics

The state-based peridynamics was proposed by Silling et al. [25], and it extends the bond-based peridynamic theory, to a general class of continuum media, which was introduced by the same author [24], to a general class of continuum media. Here we will not provide a comprehensive review of the method, and the reader is referred to the original papers as well as some subsequent contributions [8, 10, 26, 29].

Figure 1 shows a schematic of a peridynamic continuum where a point $\mathbf{X}_I$ in the reference configuration is considered to be influenced by its neighboring points $\mathbf{X}_J$, within a certain distance δ that is called as the horizon or the smoothing length. Note that the local domain formed by the horizon around each point is a compact support of that point $\mathcal{H}_{\mathbf{X}_I}$, and that the relative position vector from material point $\mathbf{X}_I$ to the material point $\mathbf{X}_J$ in the reference configuration is called a reference bond, denoted by $\mathbf{X}_{I:J} = \mathbf{X}_J - \mathbf{X}_I$.

To begin with, we first consider the local balance equations of linear momentum,

$$\rho\ddot{\mathbf{u}}\left[\mathbf{X}_I, t\right] = \nabla \cdot \boldsymbol{\sigma}\left[\mathbf{X}_I, t\right] + \mathbf{b}\left[\mathbf{X}_I, t\right] \quad \longrightarrow \quad \rho\ddot{\mathbf{u}}_I = \nabla \cdot \boldsymbol{\sigma}_I + \mathbf{b}_I \tag{1}$$

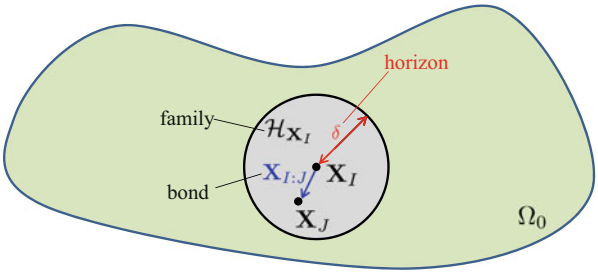


Fig. 1 State-based peridynamics continuum in the reference configuration [3]

where ρ is the density of the material, $\boldsymbol{\sigma}_I$ is the Cauchy stress at point $\mathbf{X}_I$ in the continuum, and $\mathbf{b}_I$ is the external body force applied to the body, following the usual notation.

Adopting a nonlocal approximation, one may replace the divergence of stress, $\nabla \cdot \boldsymbol{\sigma}_I$, by an integral relation,

$$\rho \ddot{\mathbf{u}}_I = \int_{\mathcal{H}_{\mathbf{X}_I}} \left(\underline{\mathbf{T}}_I \langle \mathbf{X}_{I:J} \rangle - \underline{\mathbf{T}}_J \langle \mathbf{X}_{J:I} \rangle \right) dV_{\mathbf{X}_I} + \mathbf{b}_I \tag{2}$$

where the term $\underline{\mathbf{T}}_A \langle \mathbf{X}_{A:B} \rangle$ is called the force state at point $\mathbf{X}_A$ operating on the reference bond $\mathbf{X}_{A:B} = \mathbf{X}_B - \mathbf{X}_A$.

The novelty of the state-based peridynamic theory lies in the introduction of the integral expression of Eq. (2) that approximates the local partial differential equation of balance of linear momentum[1] with a nonlocal integral equation.

In the state-based peridynamics, the definition of the force states $\underline{\mathbf{T}}$ is an important concept in the development of the theory (see [8, 10, 25, 26]). For our purpose, we are only interested in a particular definition for this force state that was proposed in [25], denominated the correspondence principle, which is able to incorporate classical stress-strain models into a peridynamic formulation by establishing an equivalence or correspondence of strain-energy density functionals:

$$\underline{\mathbf{T}}_A \langle \mathbf{X}_{A:B} \rangle = \omega(|\mathbf{X}_{AB}|) \boldsymbol{\sigma}_A \cdot \mathbf{K}_A^{-1} \cdot \mathbf{X}_{A:B} \tag{3}$$

where $\boldsymbol{\sigma}_A$ is the stress calculated at point $\mathbf{X}_A$, and $\mathbf{X}_{A:B}$ is once again the reference bond between the points, and $\omega_{|\mathbf{X}_{AB}|}$ is the smoothing function that is defined exactly in the same way as other meshfree methods, i.e. it is a scalar window function that depends on the distance between the two points, $|\mathbf{X}_{AB}| = |\mathbf{X}_B - \mathbf{X}_A|$. The tensor $\mathbf{K}_A$ is called the reference shape tensor at point $\mathbf{X}_A$ and is defined as:

[1] Note that this equation is presented by using an abbreviated notation in this paper where the time dependence of each term is suppressed.

$$\mathbf{K}_I = \int_{\mathcal{H}_{\mathbf{X}_I}} \omega(|\mathbf{X}_{IJ}|)\mathbf{X}_{I:J} \otimes \mathbf{X}_{I:J} dV_{\mathbf{X}_I} \tag{4}$$

which is exactly the same as the moment matrix tensor introduced in RKPM method e.g. [15, 19], when the polynomial basis is chosen as the linear polynomial.

The last quantity that needs to be defined in the theory is the deformation gradient such that the stress at each point, $\boldsymbol{\sigma}_I \equiv \boldsymbol{\sigma}(\boldsymbol{\mathcal{F}}_I)$, can be calculated[2]:

$$\boldsymbol{\mathcal{F}}_I = \left(\int_{\mathcal{H}_{\mathbf{X}_I}} \omega(|\mathbf{X}_{IJ}|)\mathbf{x}_{I:J} \otimes \mathbf{X}_{I:J} dV_{\mathbf{X}_I} \right) \cdot \mathbf{K}_I^{-1} . \tag{5}$$

Finally, the integral formulation presented above can be discretized by replacing the integrals by discrete summation over a set of particles over the compact support $\mathcal{H}_{\mathbf{X}_I}$, in which there are L points surrounding point $\mathbf{X}_I$. The discretized equation (2), using the correspondence principle given by Eq. (3), can be expressed as,

$$\rho\ddot{\mathbf{u}}_I = \left(\sum_{J=1}^{L} \omega(|\mathbf{X}_{IJ}|) \left(\boldsymbol{\sigma}_J \cdot \mathbf{K}_J^{-1} + \boldsymbol{\sigma}_I \cdot \mathbf{K}_I^{-1} \right) \cdot \mathbf{X}_{J:I} \Delta V_{\mathbf{X}_J} \right) + \mathbf{b}_I \tag{6}$$

Since $\omega(|\mathbf{X}_{IJ}|) = \omega(|\mathbf{X}_{JI}|)$, and $\mathbf{X}_{I:J} = -\mathbf{X}_{J:I}$. the stress $\boldsymbol{\sigma}_I \equiv \boldsymbol{\sigma}(\boldsymbol{\mathcal{F}}_I)$ can be calculated based on any constitutive law that depends on the discretized nonlocal deformation gradient,

$$\boldsymbol{\mathcal{F}}_I = \left(\sum_{J=1}^{L} \omega(|\mathbf{X}_{IJ}|)\mathbf{x}_{I:J} \otimes \mathbf{X}_{I:J} \Delta V_I \right) \cdot \mathbf{K}_I^{-1}, \tag{7}$$

with $\mathbf{x}_{I:J}$ as the current bond, $\mathbf{X}_{I:J}$ as the reference bond, and $\mathbf{K}_I$ as the discrete shape tensor that is defined as,

$$\mathbf{K}_I = \sum_{J=1}^{L} \omega(|\mathbf{X}_{IJ}|)\mathbf{X}_{I:J} \otimes \mathbf{X}_{I:J} \Delta V_I . \tag{8}$$

2.2 Reproducing Kernel Particle Method

Meshfree methods have been extensively studied since 1990s, and several reviews are available in the literature e.g. [1, 14, 23]. The basic idea of interpolating meshfree methods is to use a polynomial approximation that may be a best fit to

[2]Note that this deformation gradient is a nonlocal quantity unlike the infinitesimal deformation gradient. See [3] for detailed discussions.

a cloud of scattered points (Diffuse element Method [22], Element-free Galerkin [2], Reproducing Kernel Particle Method [18], etc.) or that may be an over-fit of that cloud of points, i.e. the polynomial passes through the points (finite difference methods [7], polynomial point interpolation method [16], etc.). These methods share the same mathematical principles and foundation as discussed in [3]. There are also non-interpolating meshfree particle methods such as molecular dynamics and the smoothed particle hydrodynamics (SPH). The readers are referred to the above literature among others for details.

Here it suffices to recall that the discrete approximation of a spatial function $f(\mathbf{X})$ at a given point $\mathbf{X}'$ can be evaluated by a polynomial approximation, in which the coefficients $\mathbf{a}$ of that polynomial series can be calculated at another point $\mathbf{X}$:

$$f^h(\mathbf{X}, \mathbf{X}') = \mathbf{p}^{\mathrm{T}}(\mathbf{X}' - \mathbf{X})\mathbf{a}(\mathbf{X}) \tag{9}$$

where $\mathbf{p}(\mathbf{x})$ is a complete polynomial basis of degree k with m monomials, and $\mathbf{a}(\mathbf{X})$ is the vector composed by the m coefficients for the polynomial basis that are used in the approximation. These coefficients depend on the position $\mathbf{X}$ where they are being evaluated, and they can be explicitly written as:

$$\mathbf{a}(\mathbf{X}) = \mathbf{M}(\mathbf{X})^{-1} \sum_{J=1}^{L} \omega(\mathbf{X}_J - \mathbf{X})\mathbf{p}(\mathbf{X}_J - \mathbf{X})\, f_J \Delta V_J \tag{10}$$

where the matrix $\mathbf{M}(\mathbf{X})$ is called as the moment matrix, and it is calculated based on the following formula,

$$\mathbf{M}(\mathbf{X}) = \sum_{J=1}^{L} \omega(\mathbf{X}_J - \mathbf{X})\mathbf{p}(\mathbf{X}_J - \mathbf{X})\mathbf{p}^{\mathrm{T}}(\mathbf{X}_J - \mathbf{X})\Delta V_J \tag{11}$$

where L is the number of total particles inside the local support of the particle $\mathbf{X}$, in which the smoothing function has the property $\omega(\mathbf{X}_J - \mathbf{X}) \neq 0$. The term ΔV_J is a measure of volume around point $\mathbf{X}_J$.

If we choose

$$\mathbf{p}(\mathbf{X} - \mathbf{X}_J) = \mathbf{X} - \mathbf{X}_J,$$

Eq. (11) recovers the definition of the shape function of the state-based peridynamics $\mathbf{K}_J(\mathbf{X})$ shown in Eq. (8). From this perspective, the state-based peridynamics is a special case of RKPM, because the RKPM moment matrix is in fact a generalized shape function; and one can use RKPM moment matrix to construct a higher order "peridynamics", which will be discussed in a separated paper.

In computations, different meshfree methods adopt different approaches to approximate the derivatives of the function $f(\mathbf{X})$. The reproducing kernel particle method (RKPM) uses a so-called synchronized derivatives to approximate the derivatives of $f(\mathbf{X})$ [11–13]. The so-called synchronized derivative is the first

non-local derivative operator that have been invented in numerical computation. For a given point $\mathbf{X}$ (fixed), we can approximate the derivative of $f(\mathbf{X})$ by taking the derivative with respect to $\mathbf{X}'$ [11–13]. The derivative of f with respect to X' is given by:

$$\frac{\partial f^h(\mathbf{X}, \mathbf{X}')}{\partial X'} = \frac{\partial \mathbf{p}^{\mathrm{T}}(\mathbf{X}' - \mathbf{X})}{\partial X'} \mathbf{a}(\mathbf{X}) \tag{12}$$

with $\mathbf{a}(\mathbf{X})$ given by Eq. (10).

Moreover, if we consider the strong form of the conservation of linear momentum, then the RKPM approximation for the divergence of stress as well as for the deformation gradient are only needed at each particle I. This means that the general equation (12) can be simplified, because we have $\mathbf{X}' = \mathbf{X} = \mathbf{X}_I$:

$$\frac{\partial f^h(\mathbf{X}_I)}{\partial \mathbf{X}} = \frac{\partial \mathbf{p}^{\mathrm{T}}(\mathbf{0})}{\partial \mathbf{X}} \mathbf{a}(\mathbf{X}_I) \tag{13}$$

where $\mathbf{a}(\mathbf{X}_I)$ is determined by:

$$\begin{aligned} \mathbf{a}(\mathbf{X}_I) &= \mathbf{M}(\mathbf{X}_I)^{-1} \sum_{J=1}^{L} \omega(\mathbf{X}_J - \mathbf{X}_I) \mathbf{p}(\mathbf{X}_J - \mathbf{X}_I) f_J \Delta V_J \\ &= \mathbf{M}(\mathbf{X}_I)^{-1} \sum_{J=1}^{L} \omega(|\mathbf{X}_{IJ}|) \mathbf{p}(\mathbf{X}_{I:J}) f_J \Delta V_J \end{aligned} \tag{14}$$

in which $\mathbf{M}(\mathbf{X}_I)$ is calculated based on the following formula,

$$\begin{aligned} \mathbf{M}(\mathbf{X}_I) &= \sum_{J=1}^{L} \omega(\mathbf{X}_I - \mathbf{X}_J) \mathbf{p}(\mathbf{X}_I - \mathbf{X}_J) \mathbf{p}^{\mathrm{T}}(\mathbf{X}_I - \mathbf{X}_J) \Delta V_J \\ &= \sum_{J=1}^{L} \omega_{I:J} \mathbf{p}(\mathbf{X}_{I:J}) \otimes \mathbf{p}(\mathbf{X}_{I:J}) \Delta V_J \end{aligned} \tag{15}$$

Note that the second line in Eqs. (14) and (15) was included to show the remarkable similarity between the derivatives calculated using RKPM with synchronized derivatives and state-based peridynamics. In fact, observing Eqs. (15) and (8) it is clear that the moment matrix used in RKPM includes the shape tensor as a submatrix. This can be illustrated by considering a specific example; let's consider a quadratic polynomial basis in a 2D problem:

$$\mathbf{p}^{\mathrm{T}}(\mathbf{X}_{I:J}) = \left[1 \; X_{I:J} \; Y_{I:J} \; (X_{I:J})^2 \; X_{I:J} Y_{I:J} \; (Y_{I:J})^2\right] \tag{16}$$

We now compute the derivative of the polynomial basis $\frac{\partial \mathbf{p}^{\mathrm{T}}(\mathbf{0})}{\partial \mathbf{X}}$ that is used to calculate the derivatives in RKPM, and we obtain,

$$\frac{\partial \mathbf{p}^{\mathrm{T}}(\mathbf{0})}{\partial X} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{17}$$

and

$$\frac{\partial \mathbf{p}^{\mathrm{T}}(\mathbf{0})}{\partial Y} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{18}$$

Hence, it is intuitive to understand that due to Eqs. (17) and (18) only a few terms of the matrix $\mathbf{a}(\mathbf{X}_I)$ will be used to calculate the derivative of $f(\mathbf{X})$, and consequently the same happens to the moment matrix $\mathbf{M}(\mathbf{X}_I)$. More importantly, the terms that are kept are the ones corresponding to the sub-vector in the polynomial basis $\begin{bmatrix} X_{I:J} & Y_{I:J} \end{bmatrix}$.

In other words, if we take $f(\mathbf{X})$ as the current position of particle x, we can then compute the deformation gradient

$$F_{ij} = \frac{\partial x_i}{\partial X_j}$$

by using the synchronized derivative approach, which leads to the exact same result that is obtained by using state-based peridynamics equation (7).

Obviously, the above explanation is not a full proof of equivalence, neither it is intended to be because the full proof was recently published by Bessa et al. [3]. Instead, the objective is to provide a simple explanation where the equivalence can be intuitively understood and where it is visible why the number of operations performed in the state-based peridynamics is significantly lower than the number of operations in RKPM. A final note in relation to the above explanation: it explains why the deformation gradient is exactly equivalent between the two methods but it does not explain why the divergence of stress is also equivalent. In fact, as can be seen in [3], the equivalence of the divergence of stress approximation is only true for uniform grids.

In summary, we conclude that the state-based peridynamics and RKPM have a strong connection. In specific, we have discovered that the state-based peridynamics derivation formulation is essentially equivalent with the RKPM synchronized derivative approach, but it has higher computational efficiency. This finding leads to the derivation or formulation of a reproducing kernel peridynamics method [3], such that the convergence analysis and boundary treatments of RKPM can be applied to the state-based peridynamics to improve its accuracy.

To demonstrate the equivalency between the state-based peridynamics and RKPM. We implement both methods to solve dynamic problems in a multiresolution continuum.

3 Multiresolution Particle Method

Recently, a multiresolution continuum theory was introduced by McVeigh et al. [20] after generalizing the micromorphic theory [6,21] to include multiscale nested micro-cells instead of single scale cell approach adopted in the early works of Erigen or Mindlin [6, 21]. The early micromorphic theory was reviewed and rigorously presented by Eringen [5], and the readers are referred to the original publications for detailed information.

Generally speaking, the micromorphic continuum is a generalization of classical continuum mechanics so that a material is considered to be a collection of *deformable* particles that have a *finite* size, rather than the usual consideration of having a collection of infinitesimal and undeformable particles. Figure 2 shows a schematic illustration of this continuum. Eringen [5,6] and Mindlin [21] derived the governing equations for the micromorphic continuum:

$$\frac{\partial}{\partial X_i^{(0)}}(\sigma_{(ij)}^{(0)} + \sigma_{ij}^{(1)}) + \rho^{(0)} b_j^{(0)} = \rho^{(0)} \dot{v_j}^{(0)} \tag{19}$$

and

$$\frac{\partial}{\partial X_i^{(0)}}(\beta_{ijk}^{(1)}) + \sigma_{jk}^{(1)} + \rho^{(1)} B_{jk}^{(1)} = I_{lj}^{(1)} \gamma_{lk}^{(1)} \tag{20}$$

where the quantities with the superscript (0) indicate macro-quantities that are present in classical continuum mechanics, while the quantities with the superscript (1) are the micro-quantities that arise from the finite deformable particles of the micromorphic continuum.

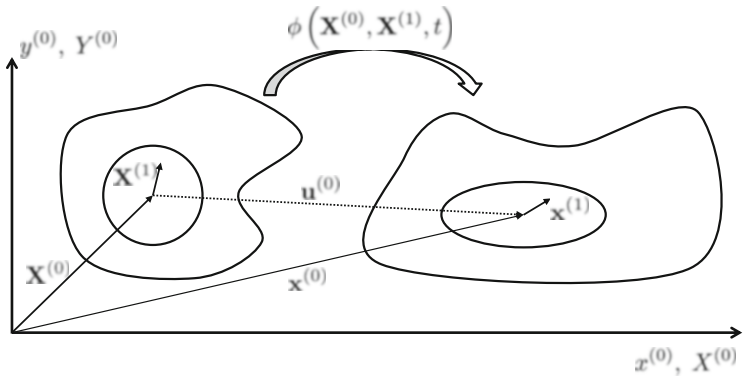


Fig. 2 Schematic of a micromorphic continuum

The macro-quantities are the (classical) macro-stress $\sigma^{(0)}_{(ij)}$, which is a symmetric tensor represented by the parentheses in the indices, the macro-density $\rho^{(0)}$, the macro body force $b^{(0)}_j$ and the macro-acceleration $\dot{v_j}^{(0)}$. The micro-quantities in the above equations are the micro-stress $\sigma^{(1)}_{ij}$ that are non-symmetric tensor in general.[3] The micro double-stress is denoted as $\sigma\sigma^{(1)}_{ijk}$, which is a higher-order stress at the fine scale, and it is conjugate to the strain-gradient; the micro-density is denoted as $\rho^{(1)}$; the micro double-body force is denoted as $B^{(1)}_{jk}$; the micro moment of inertia is denoted as $I^{(1)}_{lj}$ for each finite deformable particle, and finally the micro-acceleration is denoted as $\gamma^{(1)}_{lk} = \dot{L}^{(1)}_{lk} + L^{(1)}_{lj} L^{(1)}_{jk}$.

The corresponding boundary conditions are expressed as:

$$n^{(0)}_i(\sigma^{(0)}_{ij} + \sigma^{(1)}_{ij}) = \bar{t}_j \qquad \text{on } \Gamma^{(0)} \tag{21}$$

and

$$n^{(0)}_i \beta^{(1)}_{ijk} = T^{(1)}_{jk} \qquad \text{on } \Gamma^{(0)} \tag{22}$$

where $\bar{t}_j$ is the applied traction, and $T^{(1)}_{jk}$ is the applied double traction.

McVeigh et al. [20] proposed a finite deformation multiresolution continuum theory, in which each coarse scale particle is modeled by a nested or successive fine scale particle representative volume elements (RVEs), which creates a multi-scale continuum theory where each set of extra degrees-of-freedom could be associated to a different physical mechanism during the deformation of the material. The derivation of the strong form of the equations of motion of the multiresolution medium is straightforward, because one can follow the multiscale virtual work approach proposed by Germain [9]. It may be noted that when deriving the micromorphic equations from a postulated virtual power principle, care must be taken so that extra kinematic quantities are assumed to be conjugate to the extra kinetic quantities. In doing so, it will result a set of partial differential equations (see [20]), which as expressed in indicial notation as follows,

$$\frac{\partial}{\partial X^{(0)}_i}(\sigma^{(0)}_{(ij)} + \sum_{n=1}^{N} \sigma^{(n)}_{ij}) + \rho^{(0)} b^{(0)}_j = \rho^{(0)} \dot{v}^{(0)}_j \qquad \text{in } \Omega^{(0)}, \tag{23}$$

and

$$\frac{\partial}{\partial X^{(0)}_i}(\beta^{(n)}_{ijk}) + \sigma^{(n)}_{jk} + \rho^{(n)} B^{(n)}_{jk} = I_{lj} \gamma^{(n)}_{lk} \qquad \text{for }, n = 1, \ldots, N \qquad \text{in } \Omega^{(0)} \tag{24}$$

[3] The micro-stress $\sigma^{(1)}_{ij}$ is non-symmetric if couple-stresses (moments per unit area) are present. (See Cosserat continuum [4] or micro-polar continuum [5] for details.)

with the corresponding boundary conditions,

$$n_i^{(0)}(\sigma_{ij}^{(0)} + \sum_{n=1}^{N} \sigma_{ij}^{(n)}) = \bar{t}_j \qquad \text{on } \Gamma^{(0)} \tag{25}$$

and

$$n_i^{(0)} \beta_{ijk}^{(n)} = T_{jk}^{(n)} \qquad \text{for }, n = 1, \ldots, N \qquad \text{on } \Gamma^{(0)} \,. \tag{26}$$

The objective of this work is not to review peridynamics theory, but rather to show the simplicity of using particle methods to formulate a computational multiresolution continuum theory, as opposed to the finite element implementation of multiresolution continuum theory that requires significant computational cost and effort as indicated in [20, 27, 28].

After discretizing the multiresolution continuum equations (23)–(24) by using strong form particle methods, each physical quantity is only evaluated at the site of particle positions. Therefore, the equations of motion can be rewritten in tensorial notation as follows,

$$\nabla \cdot \left(\boldsymbol{\sigma}_I^{(0)} + \sum_{n=1}^{N} \boldsymbol{\sigma}_I^{(n)} \right) + \mathbf{b}_I^{(0)} = \rho \ddot{\mathbf{u}}_I^{(0)} \tag{27}$$

and

$$\nabla \cdot \boldsymbol{\beta}_I^{(n)} + \boldsymbol{\sigma}_I^{(n)} + \mathbf{b}_I^{(n)} = \mathbf{I} \cdot \left(\boldsymbol{\gamma}^{(n)} \right)^T \,. \tag{28}$$

Depending on the specific particle method adopted, the specific derivatives are approximated differently. In the next subsections, we shall present the two different discretizations using both the state-based peridynamics and the reproducing kernel particle method with synchronized or wavelet derivatives.

3.1 Multiresolution vs. the State-Based Peridynamics

From Sect. 2.1 and by inspecting Eq. (6) one may find that it is straightforward to determine the state-based peridynamics approximation for the divergence of the stress measures at different scales (macro-, micro- and double-stresses) as presented in the multiresolution equations (27)–(28),

$$\nabla \cdot \left(\boldsymbol{\sigma}_I^{(0)} + \sum_{n=1}^{N} \boldsymbol{\sigma}_I^{(n)} \right) \rightarrow \sum_{J=1}^{L} \omega(|\mathbf{X}_{IJ}|) \left[\left(\boldsymbol{\sigma}_J^{(0)} + \sum_{n=1}^{N} \boldsymbol{\sigma}_J^{(n)} \right) \cdot \mathbf{K}_J^{-1} + \left(\boldsymbol{\sigma}_I^{(0)} + \sum_{n=1}^{N} \boldsymbol{\sigma}_I^{(n)} \right) \cdot \mathbf{K}_I^{-1} \right] \cdot \mathbf{X}_{J:I} \Delta V_{\mathbf{X}_J} \tag{29}$$

and

$$\nabla \cdot \boldsymbol{\beta}_I^{(n)} \rightarrow \sum_{J=1}^{L} \omega_{I:J} \left[\boldsymbol{\beta}_J^{(n)} \cdot \mathbf{K}_J^{-1} + \boldsymbol{\beta}_I^{(n)} \cdot \mathbf{K}_I^{-1} \right] \cdot \mathbf{X}_{J:I} \Delta V_{\mathbf{X}_J} \tag{30}$$

where the stress measures at different scales are calculated according to given constitutive laws that are usually functions of the macro deformation gradient and the micro deformation gradients and their gradients:

$$\boldsymbol{\sigma}_I^{(0)} \equiv f \left(\boldsymbol{\mathcal{F}}_I^{(0)} \right), \tag{31}$$

$$\boldsymbol{\sigma}_I^{(n)} \equiv g^{(n)} \left(\boldsymbol{\mathcal{F}}_I^{(n)} - \boldsymbol{\mathcal{F}}_I^{(0)} \right), \tag{32}$$

and

$$\boldsymbol{\beta}_I^{(n)} \equiv h^{(n)} \left(\nabla \boldsymbol{\mathcal{F}}_I^{(n)} \right) \tag{33}$$

where f, $g^{(n)}$, and $h^{(n)}$ are the constitutive relations expressed as the functions of the strain measure adopted. Note that the micro-stresses $\boldsymbol{\sigma}_I^{(n)}$ at particle I are conjugate to a relative strain measure.

The spatial gradients at different scales are needed to compute the stresses at the different scales, which can be approximated by using the state-based peridynamics approach as follows,

$$\nabla \mathbf{x}_I^{(0)} \rightarrow \boldsymbol{\mathcal{F}}_I^{(0)} = \left(\sum_{J=1}^{L} \omega(|\mathbf{X}_{IJ}|) \mathbf{x}_{I:J}^{(0)} \otimes \mathbf{X}_{I:J}^{(0)} \Delta V_I \right) \cdot \mathbf{K}_I^{-1} \tag{34}$$

and

$$\nabla \mathbf{F}_I^{(n)} \rightarrow \nabla \boldsymbol{\mathcal{F}}_I^{(n)} = \left(\sum_{J=1}^{L} \omega(|\mathbf{X}_{IJ}|) \boldsymbol{\mathcal{F}}_{I:J}^{(n)} \otimes \mathbf{X}_{I:J}^{(0)} \Delta V_I \right) \cdot \mathbf{K}_I^{-1} \tag{35}$$

with the shape tensor defined as before, i.e.

$$\mathbf{K}_I = \sum_{J=1}^{L} \omega(|\mathbf{X}_{IJ}|)\mathbf{X}^{(0)}_{I:J} \otimes \mathbf{X}^{(0)}_{I:J} \Delta V_I \ . \tag{36}$$

We recall that in the multiresolution continuum theory the extra degrees-of-freedom are the micro deformation gradients $\boldsymbol{\mathcal{F}}^{(n)}_I$. Therefore, only the first derivative of the extra degrees-of-freedom is required when computing the gradient of the micro deformation gradient $\nabla \boldsymbol{\mathcal{F}}^{(n)}_I$ (see [20, 27] for details).

With some additional manipulations, we can derive a non-local multiscale peridynamics formulation. For example, the two-scale micromorphic peridynamics integral equations may be written as follows,

$$\rho^{(0)}\ddot{\mathbf{u}} = \int_{H^0_X} \Big(\mathbf{T}[\mathbf{X}, t] < \mathbf{X}' - \mathbf{X} > - \mathbf{T}[\mathbf{X}', t] < \mathbf{X} - \mathbf{X}' > \Big) dV_{X'}$$

$$- \int_{H^0_X} \Big(V[\boldsymbol{\psi}, t] < \mathbf{X}' - \mathbf{X} > - V[\boldsymbol{\psi}', t] < \mathbf{X}' - \mathbf{X} > \Big) dV_{X'} + \mathbf{b}^{(0)}(\mathbf{X}) \tag{37}$$

$$\boldsymbol{\gamma}^{(1)} \cdot \mathbf{I}^{(1)} = \int_{H^1_\Xi} \Big\{ \Big(W[\boldsymbol{\lambda}, t] < \boldsymbol{\Xi}' - \boldsymbol{\Xi} > - W[\boldsymbol{\lambda}', t] < \boldsymbol{\Xi}' - \boldsymbol{\Xi} > \Big)$$

$$- \Big(U[\boldsymbol{\psi}, t] < \boldsymbol{\Xi}' - \boldsymbol{\Xi} > - U[\boldsymbol{\psi}', t] < \boldsymbol{\Xi}' - \boldsymbol{\Xi} > \Big) \Big\} dV_\Xi + \mathbf{b}^{(1)}(\boldsymbol{\Xi}) \tag{38}$$

where $\mathbf{X} := \mathbf{X}^{(0)}$ is the coarse scale coordinate; $\boldsymbol{\Xi} := \mathbf{X}^{(1)}$ is the fine scale coordinate; $\boldsymbol{\psi} := \mathbf{F}^{(1)} - \mathbf{F}^{(0)}$ is the micro deformation at the fine scale, and $\boldsymbol{\lambda} := \nabla_\Xi \otimes \mathbf{F}^{(1)}$ is the gradient of the mciro deformation at fine scale.

In Eq. (37), $V[\boldsymbol{\psi}, t]$ is a non-local micro force state, and In Eq. (38), $W[\boldsymbol{\lambda}, t]$ is a non-local micro couple state, and $U[\boldsymbol{\psi}, t]$ is a non-local micro stress state. The particle method implementations of Eqs. (37) and (38) in three-dimensional space will be reported in a separate paper. In the following sections, only an one-dimensional example will be discussed in order to demonstrate the validity of the above equations.

3.2 *Multiresolution Analysis vs. RKPM Synchronized Derivative Approach*

The RKPM approximation for the divergence of different stress measures and their gradients are determined by strains and strain gradients, which are obtained from Eqs. (13), (14), and (15) presented in Sect. 2.2. The fact that we have different quantities in the multiresolution continuum theory does not change the way the derivatives are calculated. Hence, if we need to approximate the divergence of the double-stress of the first extra scale, we need to calculate the derivatives of each component according to that formula, e.g. the derivative of component β_{111} with

respect to Y is obtained as follows,

$$\frac{\partial \beta_{111}(\mathbf{X}_I^{(0)}, t)}{\partial Y^{(0)}} = \frac{\partial \left(\beta_{111}\right)_I}{\partial Y^{(0)}} = \frac{\partial \mathbf{p}^{\mathrm{T}}(\mathbf{0})}{\partial Y^{(0)}} \tag{39}$$

where $\mathbf{a}(\mathbf{X}_I^{(0)})$ is determined by:

$$\mathbf{M}(\mathbf{X}_I^{(0)})^{-1} \sum_{J=1}^{L} \omega(\mathbf{X}_J^{(0)} - \mathbf{X}_I^{(0)}) \mathbf{p}(\mathbf{X}_J^{(0)} - \mathbf{X}_I^{(0)}) \left(\beta_{111}\right)_J \Delta V_J \tag{40}$$

with $\mathbf{M}(\mathbf{X}_I^{(0)})$ being given as before in Eq. (15).

The remaining derivations of the double-stress components and the other stress measures can be proceeded in the similar way as above, i.e. by replacing $f(\mathbf{X})$ in Eqs. (13)–(15) by the desired components of each quantity.

3.3 *Illustrative Example*

Since we are still in the early stages of using particle methods to study multiresolution problems in the micromorphic continuum, we only present preliminary results obtained for a one-dimensional example by using both particle methods discussed above, i.e. the state-based peridynamics and the reproducing kernel particle method using the synchronized derivative approach. We hope to accomplish two objectives with this simple illustration: (1) To demonstrate that these methods can be used as an alternative to the finite element method, and (2) To show that both methods the difference as well as similarity of the two methods by comparing their solutions.

We consider an one-dimensional bar shown in Fig. 3, where each particle in the continuum is considered to be a micro cell that is capable of finite deformation. Since only one micro scale is being modeled here, and no other nested microscales are being considered in this example, this is is a multiresolution continuum of

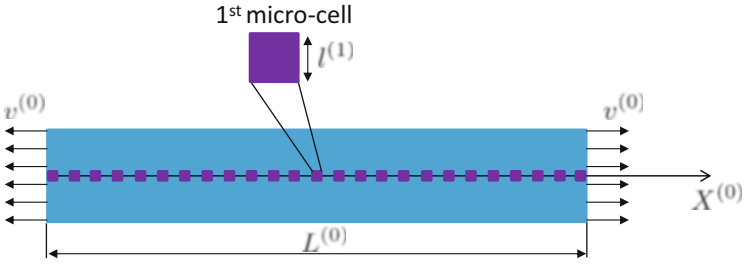


Fig. 3 Uniaxial tensile bar with a multiresolution continuum of grade 1

grade 1, i.e. this is a micromorphic continuum. In other words, the material being considered is a two-scale microstructured material. A similar problem was also considered by Vernerey et al. [27].

For this particular problem, the equations of motion can be derived from Eqs. (23) and (24) as,

$$\frac{\partial}{\partial X^{(0)}}(\sigma^{(0)} + \sigma^{(1)}) = \rho^{(0)}\dot{v}_j^{(0)} \qquad \text{in } \frac{L^{(0)}}{2} \leq X^{(0)} \leq \frac{L^{(0)}}{2} \tag{41}$$

and

$$\frac{\partial}{\partial X^{(0)}}(\beta^{(1)}) + \sigma^{(1)} = I^{(1)}\gamma^{(1)} \qquad \text{in } \frac{L^{(0)}}{2} \leq X^{(0)} \leq \frac{L^{(0)}}{2} \tag{42}$$

with the corresponding boundary conditions,

$$\begin{aligned} v^{(0)} &= -\bar{v}_0 \quad D^{(1)} = 0 \quad \text{for } X^{(0)} = -\tfrac{L^{(0)}}{2} \\ v^{(0)} &= \bar{v}_0 \quad D^{(1)} = 0 \quad \text{for } X^{(0)} = \tfrac{L^{(0)}}{2} \end{aligned} \tag{43}$$

Considering that the micro-cell is a cube with length $l^{(1)}$, we can calculate the moment inertia density of the micro-cell as,

$$I^{(1)} = \rho^{(1)} \frac{\left(l^{(1)}\right)^2}{6} \tag{44}$$

Finally, we can define the micro-acceleration $\gamma^{(1)}$ as,

$$\gamma^{(1)} = \dot{D}^{(1)} + \left(D^{(1)}\right)^2 . \tag{45}$$

In this example, the total length of the bar is $L^{(0)} = 100\,\text{mm}$, and the material properties of the bar are listed in Table 1. In this work, the finite deformation of the multiresolution continua is considered, and the constitutive update for each stress measure is performed by using the rate form of the constitutive equation, similar to what was done in [27]:

$$\frac{d\sigma^{(0)}}{dt} = E^{(0)} D^{(0)}, \tag{46}$$

Table 1 Material parameters for macro- and micro-scales

$E^{(0)}$ (GPa)	$\rho^{(0)}$ (kg/m^3)	$\sigma_y^{(0)}$ (GPa)	$E^{(1)}$ (GPa)	$\rho^{(1)}$ (kg/m^3)	$l^{(1)}$ (mm)
200	7,850	20	20	7,065	2, 4, 8

$$\frac{d\sigma^{(1)}}{dt} = E^{(1)}\left(D^{(1)} - D^{(0)}\right), \tag{47}$$

and

$$\frac{d\beta^{(1)}}{dt} = E^{(1)}\frac{\left(l^{(1)}\right)^2}{6}D^{(0)}_{,x} \tag{48}$$

Note that since the constitutive laws are written in the rate form, the kinematic conjugate to each stress measure is the respective rate-of-deformation or its gradient. Therefore, in this example the extra degree-of-freedom is $D^{(1)}$ instead of the deformation gradient as what was discussed in the previous sections. Also note that in the multiresolution theory the double-stress $\sigma\sigma^{(1)}$ does not needs to be a function of the elasticity constants used for the micro-cell, and this decreases the number of constants in the additional constitutive laws.

Generally speaking, finding the constitutive laws for micromorphic or multiresolution continua is still an open research topic, and it is still outstanding challenge in modeling of multiresolution continua. This work is an early step towards solving this complex problem, which we hope to serve as a proof of concept in the subsequent developments.

Finally, since the plastic yielding for the macro-scale is considered, it triggers a perfect plastic behavior at the center of the bar, and in turn the bar will localize at the center when the stress reaches to the value of the yield stress. Three different simulations have been performed using three different micro-cell lengths $l^{(1)}$ as shown in Table 1. The simulation parameters used in the computation are as follows: the smoothing length is 15 mm, the number of particles is 300, the time step was 10^{-6} ms, and the total simulation time was 3×10^{-2} ms. The simulations have been carried out for both particle methods: the state-based peridynamics and RKPM using synchronized derivative approach.

Figure 4 shows the result obtained with any of the two methods. First and foremost, we verify for this simple example the analytic proof provided by Bessa et al. [3] and intuitively reached in Sect. 2 of this paper: state-based peridynamics and RKPM are equivalent for uniform grids. Secondly, this simple example shows that it is fairly straightforward to discretize the multiresolution continuum equations using strong form particle methods. In fact, it is clear to observe the effect of the extra quantities of the multiresolution continuum that regularize the strain localization. This regularization is comparable to a similar study performed by Vernerey et al. [27], although the study presented here is dynamic problem instead of quasi-static problem, and the material parameters used in the two simulations are slightly different. In addition, it can be seen that for different micro-cell lengths the size of the localization zones are different as expected.

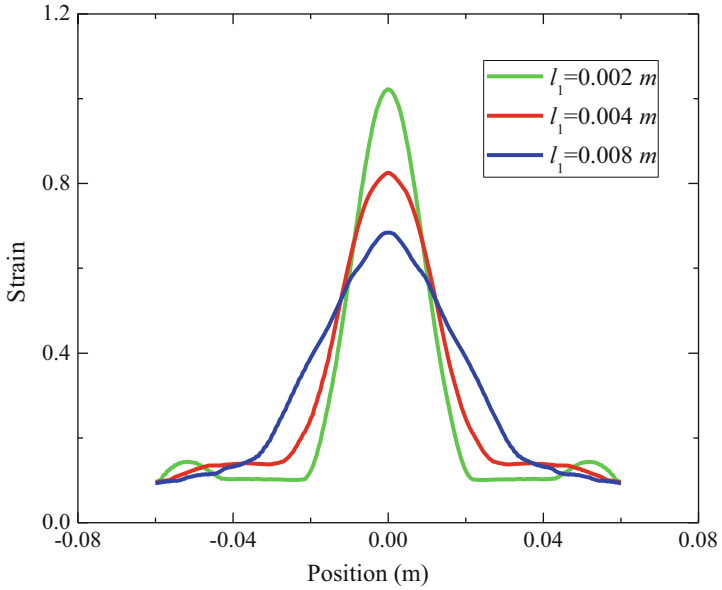


Fig. 4 Tensile strain along the bar at time 3×10^{-2} ms for three different micro-cell lengths. Both methods, state-based peridynamics and RKPM obtained the exact same result

4 Discussions

In this short paper, we present an intuitive explanation on the equivalence between state-based peridynamics and the reproducing kernel particle method. Even though an analytical proof was reported by Bessa et al. [3], and it was stated that the two are equivalent, but the state-based peridynamics formulation has more computational efficiency, we would like to point out that the RKPM synchronized derivative approach is proven to be convergent (see [12]), and when the order of the polynomial basis $\mathbf{P}(\mathbf{X})$ is greater than one it will enjoy faster convergence rate than that of the peridynamics simulations.

Moreover, in this work we have shown some preliminary results of application of these two particle methods to solve dynamics problems of multiresolution media. The simplicity of the approach developed in this work is attractive to solve the complex multiscale coupling problem of micromorphic media, but further research needs to be carried out in order to prove the stability of the method for multidimensions and non-uniform particle distribution since strong form particle methods based on polynomial basis functions suffer from instabilities for Neumann boundary conditions.

A forthcoming publication will address this matter in details, and the viability of the method will be assessed through more general and realistic examples.

References

1. T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, P. Krysl, Meshless methods: an overview and recent developments. Comput. Methods Appl. Mech. Eng. **139**(14), 3–47 (1996)
2. T. Belytschko, Y.Y. Lu, L. Gu, Element-free Galerkin methods. Int. J. Numer. Methods Eng. **37**(2), 229–256 (1994)
3. M.A. Bessa, J.T. Foster, T. Belytschko, W.K. Liu, A meshfree unification: reproducing kernel peridynamics. Comput. Mech. **53**(6), 1251–1264 (2014)
4. E. Cosserat, F. Cosserat, *Théorie des corps déformables* (A. Hermann et fils, Paris, 1909)
5. A.C. Eringen, *Nonlocal Continuum Field Theories* (Springer, New York, 2002)
6. A.C. Eringen, E.S. Suhubi, Nonlinear theory of simple micro-elastic solids—i. Int. J. Eng. Sci. **2**(2), 189–203 (1964)
7. B. Fornberg, Generation of finite difference formulas on arbitrarily spaced grids. Math. Comput. **51**(184), 699–706 (1988)
8. J.T. Foster, S.A. Silling, W.W. Chen, Viscoplasticity using peridynamics. Int. J. Numer. Methods Eng. **81**(10), 1242–1258 (2010)
9. P. Germain, The method of virtual power in continuum mechanics. Part 2: microstructure. SIAM J. Appl. Math. **25**(3), 556–575 (1973). (English)
10. R.B. Lehoucq, S.A. Silling, Force flux and the peridynamic stress tensor. J. Mech. Phys. Solids **56**(4), 1566–1577 (2008)
11. S. Li, W.K. Liu, Synchronized reproducing kernel interpolant via multiple wavelet expansion. Comput. Mech. **21**(1), 28–47 (1998). (English)
12. S. Li, W.K. Liu, Reproducing kernel hierarchical partition of unity part i: formulations. Int. J. Numer. Methods Eng. **45**, 251–288 (1999)
13. S. Li, W.K. Liu, Reproducing kernel hierarchical partition of unity part ii: applications. Int. J. Numer. Methods Eng. **45**, 289–300 (1999)
14. S. Li, W.K. Liu, Meshfree and particle methods and their applications. Appl. Mech. Rev. **55**, 1–34 (2002)
15. S. Li, W.K. Liu, *Meshfree Particle Methods* (Springer, Berlin/New York, 2004)
16. G.R. Liu, Y.T. Gu, A point interpolation method for two-dimensional solids. Int. J. Numer. Methods Eng. **50**(4), 937–951 (2001)
17. W.K. Liu, S. Jun, S. Li, J. Adee, T. Belytschko, Reproducing kernel particle methods for structural dynamics. Int. J. Numer. Methods Eng. **38**(10), 1655–1679 (1995)
18. W.K. Liu, S. Jun, Y.F. Zhang, Reproducing kernel particle methods. Int. J. Numer. Methods Fluids **20**(8–9), 1081–1106 (1995)
19. W.K. Liu, S. Li, T. Belytschko, Moving least-square reproducing kernel methods (i) methodology and convergence. Comput. Methods Appl. Mech. Eng. **143**(1–2), 113–154 (1997)
20. C. McVeigh, F. Vernerey, W.K. Liu, L.C. Brinson, Multiresolution analysis for material design. Comput. Methods Appl. Mech. Eng. **195**, 5053–5076 (2006). John H. Argyris Memorial Issue. Part I
21. R.D. Mindlin, Micro-structure in linear elasticity. Arch. Ration. Mech. Anal. **16**(1), 51–78 (1964). (English)
22. B. Nayroles, G. Touzot, P. Villon, Generalizing the finite element method: diffuse approximation and diffuse elements. Comput. Mech. **10**(5), 307–318 (1992). (English)
23. V.P. Nguyen, T. Rabczuk, S. Bordas, M. Duflot, Meshless methods: a review and computer implementation aspects. Math. Comput. Simul. **79**(3), 763–813 (2008)
24. S.A. Silling, Reformulation of elasticity theory for discontinuities and long-range forces. J. Mech. Phys. Solids **48**(1), 175–209 (2000)
25. S.A. Silling, M. Epton, O. Weckner, J. Xu, E. Askari, Peridynamic states and constitutive modeling. J. Elast. **88**(2), 151–184 (2007). (English)
26. S.A. Silling, R.B. Lehoucq, Convergence of peridynamics to classical elasticity theory. J. Elast. **93**(1), 13–37 (2008). (English)

27. F. Vernerey, W.K. Liu, B. Moran, Multi-scale micromorphic theory for hierarchical materials. J. Mech. Phys. Solids **55**(12), 2603–2651 (2007)
28. F.J. Vernerey, W.K. Liu, B. Moran, G. Olson, A micromorphic model for the multiple scale failure of heterogeneous materials. J. Mech. Phys. Solids **56**(4), 1320–1347 (2008)
29. K. Zhou, Q. Du, Mathematical and numerical analysis of linear peridynamic models with nonlocal boundary conditions. SIAM J. Numer. Anal. **48**(5), 1759–1780 (2010)

Co-simulations of Discrete and Finite Element Codes

Carsten Dehning, Claas Bierwisch, and Torsten Kraft

Abstract This paper describes methods and algorithms implemented for the co-simulation between a mesh-free discrete element method (DEM) code and a finite element analysis (FEA) code under control of a co-simulation middleware software environment.

The involved software packages, the DEM code SimPARTIX, the FEA code Abaqus and the co-simulation environment MpCCI are briefly described. The codes were extended to account for the exchange of quantities (positions, velocities, forces, heat) between the individual particles on the DEM side and the contact surfaces of the FE application, which now allows the joint simulation of the structure-structure interaction between a particulate phase and arbitrary elastic structures. The MpCCI software package was improved with respect to the massive parallel computation of the DEM code and high-speed communication on high performance computing (HPC) clusters.

A few validation examples as well as a more complex simulation of the motion of a tire on sticky soil will be presented.

Keywords Co-simulation • Discrete element method (DEM) • Finite element method (FEM) • Tire traction

1 Introduction

Lagrangian particle tracking methods, implemented in many computational fluid dynamics (CFD) codes, are used since a long time to simulate the particle transport and dispersion in multiphase flows. The particle concentration is at most low and particle-particle interaction due to collision is often neglected due to the computational effort. In addition, the friction due to the relative motion between

C. Dehning (✉)
Fraunhofer SCAI, Schloss Birlinghoven, 53754 Sankt Augustin, Germany
e-mail: carsten.dehning@scai.fraunhofer.de

C. Bierwisch • T. Kraft
Fraunhofer IWM, Wöhlerstr. 11, 79108 Freiburg, Germany

M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations VII*, Lecture Notes in Computational Science and Engineering 100,
DOI 10.1007/978-3-319-06898-5_4

the particle and its surrounding fluid (Stokes' law or the Basset-Boussinesq-Oseen equation [1]) dominates the particle inertia and the motion of each particle.

Although the equations used to describe the physical phenomena of the fluid flow and the particles are quite different and require different solution schemes, both methods can easily be implemented into a single monolithic code.

In distinction to the (multiphase) flow driven particle motion, the flow of granular matter like of powders, dry sand or even sticky pastes is dominated by direct particle-particle interaction (collision, contact friction) or by short distance forces (cohesion) as a result of the high particle concentration. Effects of the surrounding fluid may be neglected as long as the particle momentum is high compared to the momentum of the fluid. The driving force is not described by the motion of the surrounding fluid but by external forces (gravitation), since the particle-motion itself drives the flow field. Further interaction between particles and structures takes place at solid or flexible boundaries. DEM codes are usually most suitable for these kinds of simulation.

Coupling resp. co-simulation of different physical phenomena, each described by separate independent and quite different equations and solved with independent simulation codes becomes attractive in order to model contact between granular matter and elastic structures. The motion of a tire on dry sandy ground, mud or even snow is an exemplary application. It might, in a similar way, be applied to mining or drilling applications, simulations of excavating machinery (interaction of the excavator shovel with soil), farming machinery (plow soil interaction) or the plug flow in floor seed pumps and pipes.

Based on the MpCCI co-simulation environment, the popular DEM code SimPARTIX can be coupled to a structural dynamics FEA code (in this paper Abaqus is used) or any fluid dynamics code.

Besides some theoretical background, this paper demonstrates the possibilities of such co-simulations. As an example, the traction of a tire on a snow-covered road is studied numerically to avoid on the one hand expensive experiments and, more importantly, to investigate the different effects like snow characteristics, tire deformation, tire pressure or tread pattern on the traction individually.

2 SimPARTIX

The DEM code presented in the paper is SimPARTIX§. SimPARTIX is developed by the Fraunhofer IWM [2].

In a DEM simulation, each individual physical grain is represented by a discrete element, which in the simplest case is a sphere, but may also have a more complex geometry [3]. Different grains interact by physical force laws such as Hertzian repulsion, Johnson-Kendall-Roberts cohesion [6], viscous dissipation or an Amonton friction law (see Fig. 1).

The combination of grain shape, grain size (distribution) and micromechanical interaction model defines the behavior of the granular bulk material. For example,

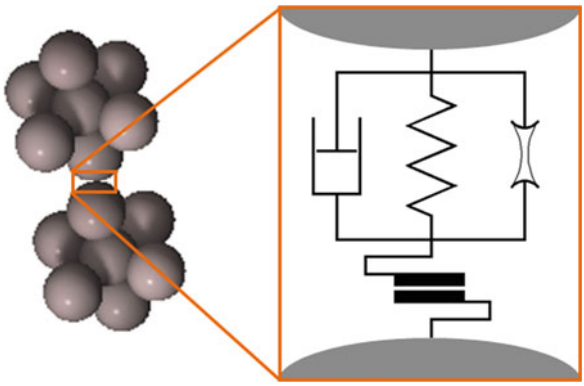


Fig. 1 Schematic of typical force laws from DEM particle contact

effective rheological properties can vary from freely flowing (powders) to sticky (pastes).

The underlying numerical scheme of the DEM code is explicit time integration of Newton's law of motion for an ensemble of discrete particles. Within SimPARTIX the spheres have a constant radius, which can be different for each sphere. A velocity Verlet algorithm [4] is used for the integration, yielding updates for the position $\mathbf{r}_i$, velocity $\mathbf{v}_i$ and angular velocity $\mathbf{w}_i$ of a certain particle i with mass m_i and moment of inertia I_i after each time step Dt in the following form:

$$\boldsymbol{r}_i(t+\Delta t) = \boldsymbol{r}_i(t) + \Delta t\, \boldsymbol{v}_i(t) + \frac{\Delta t^2}{2m_i}\boldsymbol{F}_i(t) \tag{1}$$

$$\boldsymbol{v}_i(t+\Delta t) = \boldsymbol{v}_i(t) + \frac{\Delta t}{2m_i}\left(\boldsymbol{F}_i(t) + \boldsymbol{F}_i(t+\Delta t)\right) \tag{2}$$

$$\boldsymbol{w}_i(t+\Delta t) = \boldsymbol{w}_i(t) + \frac{\Delta t I_i^{-1}}{2}\left(\boldsymbol{T}_i(t) + \boldsymbol{T}_i(t+\Delta t)\right) \tag{3}$$

The force $\mathbf{F}_i$ and torque $\mathbf{T}_i$ acting on each particle determine its motion.

Thus, the definition of appropriate force laws is a key ingredient within DEM simulations. The total force on particle i with radius R_i is the sum of forces due to interactions with other particles j and the contributions of gravity and Stokes' drag, latter only if there is an interaction with a surrounding fluid with dynamic viscosity μ_a.

$$\mathbf{F}_i = \sum_{j \neq i}\left(\boldsymbol{f}_{ij}^{\text{rep}} + \boldsymbol{f}_{ij}^{\text{damp}} + \boldsymbol{f}_{ij}^{\text{slide}} + \boldsymbol{f}_{ij}^{\text{coh}}\right) + \boldsymbol{f}_i^{\text{grav}} + \boldsymbol{f}_i^{\text{stokes}} \tag{4}$$

Four interaction forces were used. They are only activated when particles are in contact, i.e. $d_{ij} > 0$.

$$d_{ij} = R_i + R_j - \left|\boldsymbol{r}_{ij}\right| \tag{5}$$

$$\boldsymbol{r}_{ij} = \boldsymbol{r}_i - \boldsymbol{r}_j \tag{6}$$

$$R_{\text{eff}} = R_i R_j / (R_i + R_j) \tag{7}$$

Hertzian repulsion $\mathbf{f}^{rep}$ prevents particle penetration. The repulsion strength is controlled by the effective Young's modulus E and the Poisson number ν.

$$\boldsymbol{f}_{ij}^{\text{rep}} = \left(\frac{2}{3}\frac{E}{1-\nu^2}\sqrt{R_{\text{eff}}}d_{ij}^{3/2}\right)\boldsymbol{r}_{ij}/\left|\boldsymbol{r}_{ij}\right| \tag{8}$$

A viscous damping term $\mathbf{f}^{damp}$ accounts for inelastic collisions due to a viscosity γ_n.

$$\boldsymbol{f}_{ij}^{\text{damp}} = -\left(\gamma_n\sqrt{R_{\text{eff}}d_{ij}}\left(\boldsymbol{v}_i - \boldsymbol{v}_j\right)\boldsymbol{r}_{ij}/\left|\boldsymbol{r}_{ij}\right|\right)\boldsymbol{r}_{ij}/\left|\boldsymbol{r}_{ij}\right| \tag{9}$$

Amonton like sliding friction is modelled by a tangential spring contact $\mathbf{f}^{slide}$ as introduced in [5] where $\mathbf{s}_{ij}$ is the tangential displacement of the initial contact points.

$$\boldsymbol{f}_{ij}^{\text{slide}} = -\min\left[\frac{8}{3}\frac{G}{2-\nu}\sqrt{R_{\text{eff}}d_{ij}}\left|\boldsymbol{s}_{ij}\right|, \mu\left|\boldsymbol{f}_{ij}^{\text{rep}} + \boldsymbol{f}_{ij}^{\text{damp}}\right|\right]\boldsymbol{s}_{ij}/\left|\boldsymbol{s}_{ij}\right| \tag{10}$$

The parameter μ is the coefficient of friction and κ_t is an effective shear modulus.

Cohesion is represented in terms of the Johnson-Kendall-Roberts theory [5]. The magnitude of the cohesive force $\mathbf{f}^{coh}$ is determined by the work of adhesion per w.

$$\boldsymbol{f}_{ij}^{\text{coh}} = -\left(\sqrt{4\pi w\frac{E}{1-\nu^2}}R_{\text{eff}}^{3/4}d_{ij}^{3/4}\right)\boldsymbol{r}_{ij}/\left|\boldsymbol{r}_{ij}\right| \tag{11}$$

The gravitational force $\mathbf{f}^{grav}$ and the stoke force $\mathbf{f}^{stokes}$ are independent from particle interactions.

$$\boldsymbol{f}_i^{\text{grav}} = -m_i g\boldsymbol{e}_z, \tag{12}$$

$$\boldsymbol{f}_i^{\text{stokes}} = -6\pi\mu_a R_i \boldsymbol{v}_i \tag{13}$$

The total torque on particle i is the sum of torques $\mathbf{t}^{slide}$ proportional to the sliding friction force.

$$\boldsymbol{T}_i = \sum_{j\neq i}\boldsymbol{t}_{ij}^{\text{slide}} \tag{14}$$

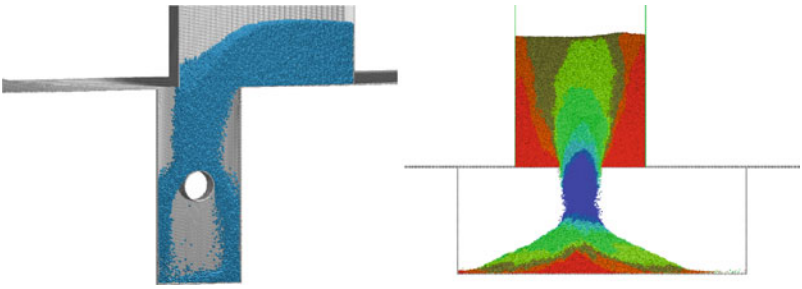


Fig. 2 Examples of granular media flow simulations with SimPARTIX

$$\boldsymbol{t}_{ij}^{\text{slide}} = R_i \, \boldsymbol{f}_{ij}^{\text{slide}} \times \boldsymbol{r}_{ij} / \left| \boldsymbol{r}_{ij} \right| \tag{15}$$

Apparently, the viscous damping force depends explicitly on the particle velocity, which according to the velocity Verlet scheme can only be updated at the end of the time step when $\mathbf{F}_i\ (t + \mathrm{D}t)$ is known. This problem is circumvented by using a predictor velocity $\mathbf{u}_i\ (t + \mathrm{D}t) = \mathbf{v}_i\ (t) + \mathbf{F}_i\ (t)\ \mathrm{D}t / m_i$ for the force calculation.

Figure 2 shows some exemplary granular flow applications for DEM simulations. On the left hand side, the filling of a slender cavity with a circular obstacle from a feeding shoe is displayed. On the right hand side, the instantaneous velocity magnitude during outflow from a hopper is shown. Red represents low velocities and blue stands for large velocities.

3 Abaqus

Abaqus is a structural mechanics FEA code, which is part of the SIMULIA software package developed by Dassault Systemes [7]. Abaqus is an industry wide well-known code with a high reputation regarding accuracy and robustness. Abaqus/Standard is an implicit (transient) nonlinear finite element solver. Its noteworthy strength is its capability to deal with sophisticated nonlinear material models and yield. A second type of solver is Abaqus/Explicit, which is especially suited for highly nonlinear systems with many complex contacts under transient loads like crash simulations.

Multiple tire companies worldwide use the FEA-code Abaqus for the prediction of the traction and wear of tires on dry solid roads. It includes already various composite material models suitable for describing tire materials (Fig. 3).

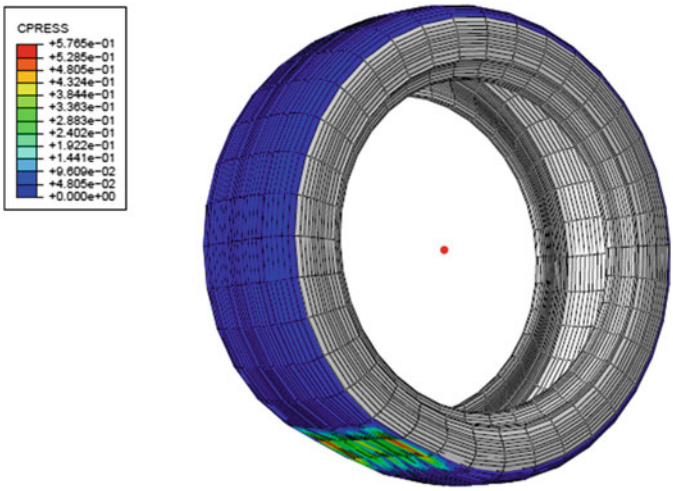


Fig. 3 Calculated pressure on the tire-road contact area

4 The Co-simulation Concept

Co-simulation in the case presented here means the controlled exchange of quantities on the coupled walls between two different simulation codes at each time step. In this surface coupling method these walls are the domain boundaries of each numerical model. The coupled walls on the structural side are in fact all parts, which may get in contact with the particles during the simulation at any time. In case of a tire, it is the outer surface of the tire, outer rim, thread and groves.

At the coupled walls, elastic and kinetic energy as well as momentum are exchanged between the structure and the particles. The DEM-code particles, hitting the wall, cause reaction forces at the wall due to repulsion, cohesion and damping as well as tangential friction. These forces are the structural load applied to the FEA model. The FEA results are new nodal displacements and the nodal velocities, which on their part are the input to the DEM-code describing the motion of the coupled wall. The velocities at the boundaries of the DEM-code account for a proper calculation of the Hertz repulsion during the particle-surface contact.

Displacements and velocity are therefore sent back from the FEA-code to the DEM-code (see Fig. 4).

Since displacements repeatedly need to be added to the true vertex coordinates on the DEM side (per time step) to keep the coordinates of the coupled surfaces in sync in both partner codes, this procedure leads to summation errors in the case of a large number of exchanges – in our case multi-million time steps. Therefore, in fact the true nodal coordinates of the deformed FEA surface are exchanged instead of the displacements.

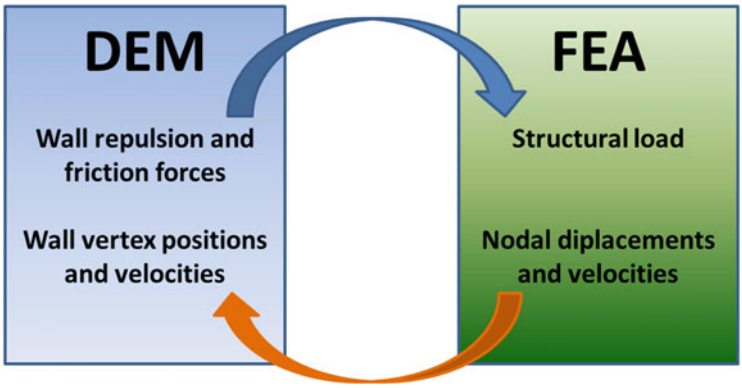


Fig. 4 Exchanged quantities during a time step cycle

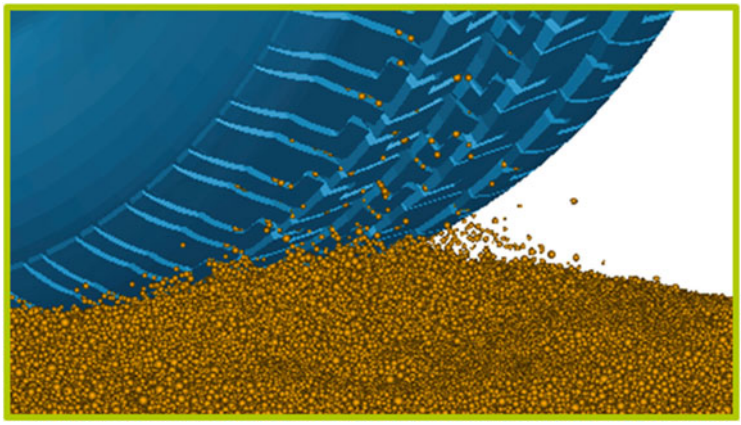


Fig. 5 Example of a tire in contact with soil

From the viewpoint of each code, there is no coupling at all. A code just sees updated transient boundary conditions at each time step. This coupling concept allows the simulation of e.g. the tire-soil interaction (see Fig. 5).

5 MpCCI

Instead of having a monolithic software which integrates different kind of solvers under one hat, the co-simulation between the introduced two codes runs under the control of the MpCCI coupling environment. Whether the DEM-solver and the FEA-solver are fully integrated into a single code or are used as separate processes

Table 1 List of codes supported by MpCCI

Code	Releases
Abaqus	6.12–6.14
Ansys/FEA	12.0–15.0
Ansys/Fluent	12.0–15.0
Ansys/ICEPAK	13.0–15.0
Elmer/CSC	Current release
FINE/Hexa	2.10–4
FINE/Open	2.11-1–2.12-3
FINE/Turbo	8.9-1–9.0-3
Flowmaster	7.6–9.0
Flux	10.3
JMAG	11.0–13.0
MATLAB	R2010b–R2012b
MD Nastran	2010.1–2013.1
MSC.Adams	2010–2013.2
MSC.Marc	2010–2013.1
OpenFOAM	1.6–2.1.1, 3.0-extend
RadTherm	10.0–11.1
SIMPACK	9.1.1–9.5
STAR-CCM+	7.02–9.02
STAR-CD	4.06–4.16
SimPARTIX	New since 2013

controlled by the middleware MpCCI is not a question of the numeric, but only of the software design.

MpCCI is a co-simulation software environment developed by the Fraunhofer SCAI [8], which handles a large number of simulation codes for a co-simulation. MpCCI is not limited to pure mesh-based codes like FEA or CFD, but includes system codes (mechanical or electrical networks) as well. System codes may e.g. act as an additional control process to simulate the steering of a vehicle or the anti-blocking system of a brake. Table 1 shows the list of codes linked to MpCCI.

For pure mesh based codes, (FEA/CFD) MpCCI provides mesh-mapping algorithms to account for the exchange of quantities on non-conformal meshes with different discretizations [8].

Figure 6 shows the basic software architecture of MpCCI.

The simulation-codes never communicate with each other directly, they even do not really know about each other. Instead, they only communicate with a central service process, the MpCCI coupling server. The communication functionality is implemented in the MpCCI code adapter, which is either statically linked to or dynamically loaded by the simulation-code.

The configuration of the coupled system is set up via a GUI. The configuration is stored in a project file, which later on acts as the input to the MpCCI server.

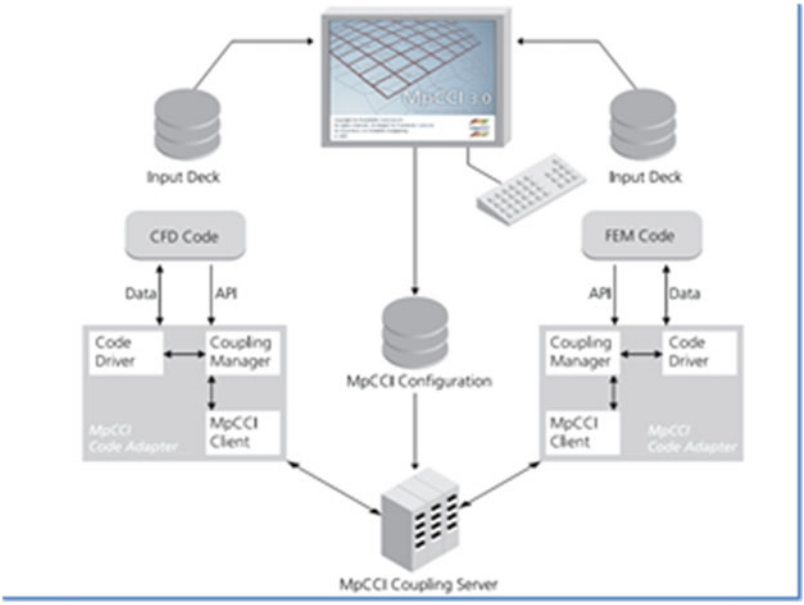


Fig. 6 The MpCCI co-simulation software architecture

After the launch of all participating processes, the two simulation-codes connect to the MpCCI server process via the adapter interface. During a first setup phase the MpCCI server delivers the configuration information about the coupled meshes and quantities and the simulation codes in turn return the initial computational grid description or particle cloud characteristics required for the quantity mapping back to the server process.

The central MpCCI server process acts as a communication manager as well as a mapping process and is responsible for the receipt, mapping and the delivery of requested data in time. It permanently controls the synchronization between the simulation-codes.

For the objectives targeted in this paper, the basic version of the MpCCI software has been widely extended:

1. Mapping of point-cloud quantities on surfaces or volumes.
2. Extensions related to the time-synchronization schemes for multi time scale problems.
3. Algorithms used for interpolations in time.
4. Performance on HPC-clusters.

5.1 Principle Code Synchronizations Methods

A simulation-code solves its individual set of equations either transient (explicit or implicit), stationary (iterative), or it just calculates a series of independent states without any information about time or an iteration counter. A trivial synchronization scheme for the co-simulation between two transient codes is the exchange of quantities at all of the coupled regions at each time step. Both simulation-codes need a perfect synchronization of their, actually individual, time step size. MpCCI however also deals with mixed types of solvers. This requires a more flexible and thus complex logic.

First, during the delivery of data from a simulation-code to the MpCCI server process the duplicated dataset within the MpCCI server is marked by an individual tag ID, an increasing only step-counter, plus optional time information and/or iteration counters in case of a transient-implicit simulation-code. The full exchange of quantities between at least two partner codes happens at certain synchronization points, called a "coupling step".

The initial exchange procedure of the quantities between the both partner codes determines on the parallel or serialized runs. The two principles of a fully synchronized computation are shown in Fig. 7.

Serial coupling is a semi-implicit co-simulation procedure, whilst parallel co-simulation is fully explicit. Parallel coupling is especially attractive to avoid computational overhead and delays.

Depending on the kind of the four solver types, we have to consider the following co-simulation procedures between them (see Table 2).

Except for the "explicit-transient" co-simulation, the synchronization between the codes is trivial, since the mapped quantity values are basically identified by either their ID, time and/or iteration counter. All codes must be in perfect sync.

If any "state-based code" is involved in a co-simulation, the synchronization procedure is named event based coupling, since the participating state-based code cannot be time – or iteration – synchronized with its partners (except via the tag ID). A typical state-based application is the determination of material properties in thermodynamic equilibrium. The properties depend on temperature and pressure

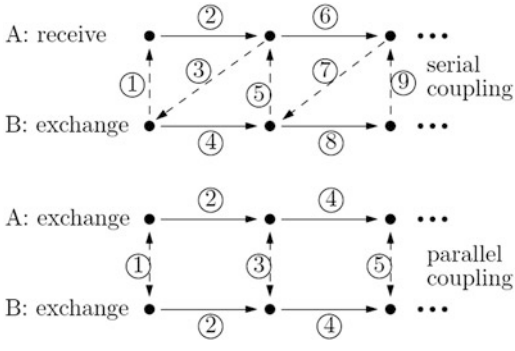


Fig. 7 Synchronous time steps or iterations, *Top*: serialized computation, *Bottom*: parallel computation

Table 2 Symmetric code synchronisation schemes

Code-1 Code-2	Transient/ implicit	Transient/ explicit	Stationary/ iterative	State
Transient/ implicit	Transient with inner iteration	Explicit transient	Iteration only	Tag ID or event based
Transient/explicit		Explicit transient	Iteration only	Tag ID or event based
Stationary/ iterative			Iteration only	Tag ID or event based
State				Tag ID or event based

only, but not on the time. This concept is not more than just a split of the complete solution into separate processes.

The event based co-simulation procedure has successfully been applied to predict the voltage drop inside the electric arch in switches and power circuit breakers [9]. In the electric arch application the reason not to synchronize the quantities exchange at prescribed time steps was caused by the fact that the transient CFD code and the steady state electromagnetic solver require extremely different processor times and the heterogeneous application could not be load balanced otherwise. The event triggering the exchange of quantities is implemented inside an application specific user function.

In our FEM/DEM co-simulation scenario presented here, both codes are transient explicit codes and the coupling is transient explicit also. An inner iteration on the boundary conditions is not possible. This is not a limitation due to the design of the MpCCI software, but due to the fact that none of the codes used here is able to repeat the same time step after an update/exchange of the boundary conditions.

5.2 Synchronization and Exchange Algorithms for the Transient Case

If, in a transient-transient case, the coupling is not event-based, both partner codes need to agree on what is called a “coupling time step size”. The full exchange of quantities (send and receive) happens at exactly these time steps. The “coupling time step size” is negotiated between the codes in various ways explained below. It must not necessarily be identical with the numerical “physical time step size” used by the participating codes to solve their individual equations. It might even be much smaller. Some of the partner codes are then forced to reduce their physical time-step-size to avoid an overshot in time at the next exchange. The time-step-size reduction should normally not be any problem from the numerical point of view. If the “coupling step size” is identical with the time-step-size of each code, the co-simulation procedure is called “strong transient coupling”.

If the “physical time step size” of a code is less than the coupling time due to a large “coupling time step size”, the procedure is called “sub-cycling”. A code may do multiple internal time steps between two coupling steps with unchanged boundary conditions. This procedure is quite useful in cases with very smooth and tiny changes of the BCs in time (between two coupling steps) and the fluctuation frequency of the BC’s is far less than coupling frequency. It might also help to reduce the computer time required for the co-simulation run, since the exchange of quantities always causes delays due to data (network) traffic and the at most required mesh mapping.

A better “sub-cycling” procedure is the full exchange of quantities between the partner codes at each “coupling time step”, and the receipt only of interpolated or extrapolated values at each numerical time step. This procedure is foreseen with strongly unbalanced partner codes, unbalanced in the sense of the computational effort required to solve a time step (see below).

5.3 Synchronization in Time

At each “coupling step” a simulation-code first sends its current physical time plus a prospective next target time (equals the current physical time plus the coupling time-step-size) to the MpCCI server. The reason for using absolute time values instead of time steps is the prevention of numerical summation errors within the MpCCI server process. At the end of a “coupling step”, after all quantities at the current physical time have been exchanged among the partners, each code receives back the negotiated new target coupling time, which might not necessarily be identical with the prospective target time initially delivered to the server.

If all transient partner codes have to be in perfect sync regarding the current time and the target coupling time, we need consider two different synchronization scenarios:

5.3.1 (I) The Master-Slave Concellpt

Exactly one of the codes is the “time-master” and all its partner codes are called “time-slaves”. The time-master defines the coupling target time and the time slaves are forced to accept the target time received from the MpCCI server.

5.3.2 (II) Time Negotiations

All codes negotiate on a new “coupling time step size”. This negotiation is implemented in the server. The server simply returns the minimum or maximum of the prospective target time of all codes. Like in scenario (I), even the (minimal) “coupling step size” might be larger than the “physical time step size” of each code.

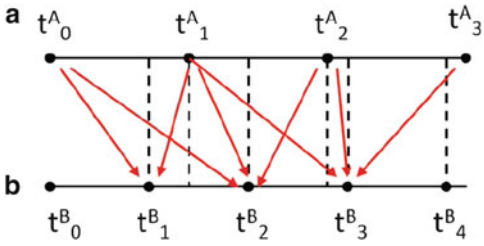


Fig. 8 Asynchronous time stepping

In opposite to this it gives each code a chance to exchange quantities with its partners at exactly each numerical time step.

5.4 *Asynchronous Transient Coupling (Independent Marching in Time)*

Two simulation-codes may advance in time independent of each other. There is no synchronization point any longer. A code may deliver or request mapped quantities at any time. The MpCCI server then keeps track of the history, and stores and maps into multiple buffers (Fig. 8).

This kind of coupling procedure is practically applied to solve "multi time scale" co-simulation problems. A typical scenario of a multi time scale problem would be the co-simulation between a transient solver advancing with tiny time step sizes, but a large number of time steps per second wall-clock-time, while its partner code requires far more computational effort per time step, but may advance with a larger time-step-size, thus far fewer steps.

An example are two phase flows with lagrangian particle tracking, in which the particle relaxation time is at most far less than the numerical time-step-size required to solve the flow field. The coupling time-step-size is then controlled by the flow solver or a structural mechanics solver, and the solution of the particle tracks is independently done. This asynchronous coupling reduces latencies (one code might be waiting for its partner code) and may be applied to achieve a "load balanced" co-simulation.

5.5 *Interpolation and Integration in Time*

The solution of multi time scale problems requires an interpolation as well as integration in time. Any space interpolation is automatically achieved due to the repeated neighborhood search either within the MpCCI server or within the DEM code.

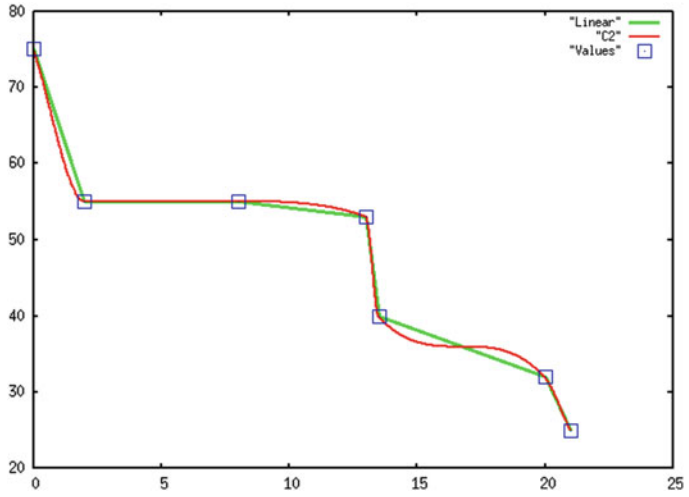


Fig. 9 Implemented linear and monotone cubic interpolation of an example dataset

The code advancing with the smaller time-step-size (in the tire-on-soil application the DEM-code) may, independently of the coupling time-step-size, exchange quantities at each time step and is than no longer in sync with its partner code. The DEM-code requests new positions and velocities at the coupled wall vertices at each physical time step (interpolation) and delivers new contact forces at the coupled vertices (require integration).

An interpolation scheme must account for the fact that coordinates and velocities are separate but dependent quantities, thus at least a C2-continuity is required. Within MpCCI a monotone cubic spline interpolation [10] is implemented. This cubic interpolation scheme allows a code also to request for the first and second time derivate of a quantity consistently (see Fig. 9).

The DEM-code sends new contact forces at the coupled vertices at each physical time step – a true exchange of quantities is only possible at the coupling time steps. Since forces are picked up from the partner code at larger time steps, the MpCCI server has to keep track of all force spikes. For integral quantities (not fields like temperature) MpCCI sums up any received integral values into an "time integral buffer", starting at time t_i. Whenever the partner code has received the integral values at time t_j, MpCCI restarts the summation.

$$\boldsymbol{f}^{\text{Abaqus}}(t_i, t_j) = \frac{1}{t_j - t_i}\left(\int_{t_i}^{t_j} \boldsymbol{f}^{\text{SimPARTIX}} dt\right) \tag{16}$$

5.6 Load Balancing with Asynchronous Coupling

Given $\triangle t_i$ as the time-step-size of a code and n time steps per coupling step, the coupling time-step-size $\triangle t_c$ for an individual code is

$$\triangle t_c = \sum_{1}^{n} \triangle t_i, \quad n \geq 1 \tag{17}$$

In case of a multi-time-scale problem, the largest time-step-size is identical with the coupling time-step-size $\triangle t_c$ with n equals 1. For the other simulation-codes, the number of time steps n and the time-step-size $\triangle t_i$ is adjusted to ensure a perfect match at each $\triangle t_c$.

The elapsed wall-clock-time $\triangle t_w$ per time-step $\triangle t_i$ of a simulation code is in general independent of $\triangle t_i$, but dominated by the complexity of the physical equations and the soft- and hardware performance. The optimal coupling procedure, from the performance point of view, requires a nearly identical wall-clock-time $\triangle t_{cw} = n \triangle t_w$ per coupling-time-step for all participating codes. Due to measurement of $\triangle t_w$ within the MpCCI server (communication delays of each code between two communication requests), the MpCCI server can give an advice on the selection of the number of steps n and the physical time-step-size to each code.

5.7 Co-simulation Test Case: Particles Dropped on an Elastic Panel

A DEM-FEA co-simulation demonstrating the principle effects and the correctness of the co-simulation setup was carried out. A bulk of particles with different radii is placed above a horizontal elastic panel clamped on one edge. The freely moving particles are subject to gravitational acceleration pointing downwards. The elastic panel modelled in Abaqus is not directly affected by gravity. A visualization of the simulation of the DEM part is shown in Fig. 10. The particles are colored randomly. The particles representing the panel in the DEM part are colored according to their horizontal position from front to back. Results of the banded panel in the FEA part are not shown but behave as expected. The time series shows how the panel first bends due to the particle impacts and swings back later because of stored elastic energy.

Note that this is a synchronous coupling and was not considered as a multi-time-scale problem, although the differences of the time-step-sizes between the DEM- and the FEA-code were at least one order of magnitude. In addition, there was no demand for load balancing the simulation, since in this example the wall-clock-time per physical time step of both codes, SimPARTIX and Abaqus/Explicit, was similar. Then the DEM-code dominates the time-step-size.

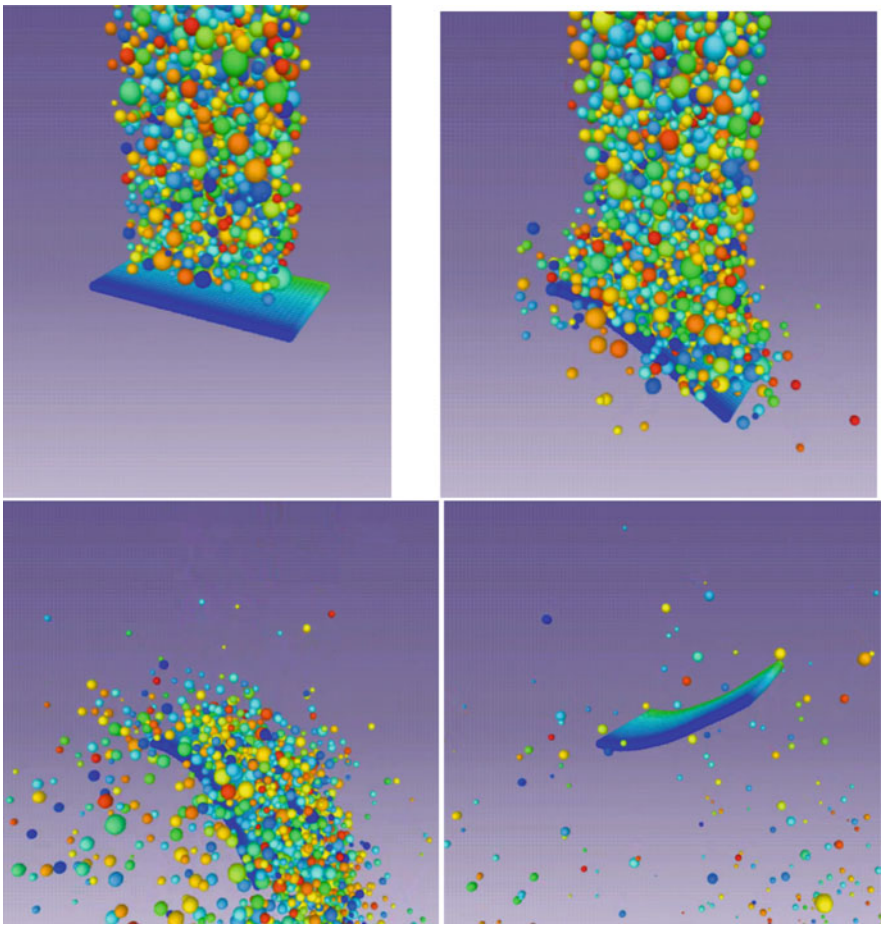


Fig. 10 Simulation snapshot of a bulk of particles falling on a panel clamped on one edge

6 First Results of a Tire-Soil Contact Simulation

The initial conditions depend on whether a car accelerates or retards. The first test case presented here is an accelerating car starting from rest.

The process is split up into three different steps, each requiring a transient simulation. At first the particles are evenly distributed within a sandbox and allowed to settle down until a static equilibrium is achieved. Thus, a soil or snow covered pavement is prepared. During the separate second (now coupled) step, the tire is "pressed" into the sandbox. The vertical load on the tire is the equivalent weight force of a typical car. Again, the objective is a static equilibrium state. As a result of the particles compaction the tire creates a footprint on the surface of the granular bed (see Fig. 11).

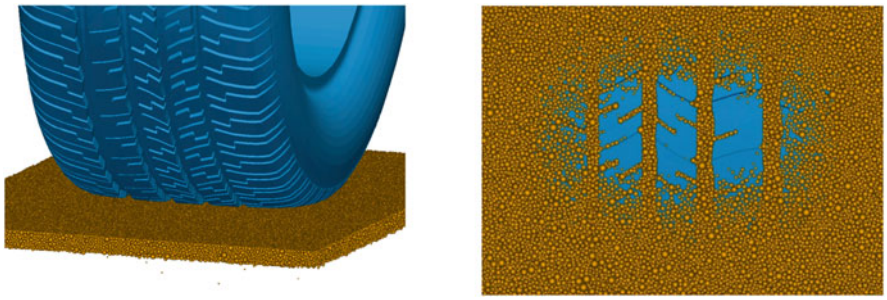


Fig. 11 Tire dropped into a sandbox. The footprint is shown on the *right picture*

The last step of this workflow is the simulation of the spinning/accelerating tire. Two different load cases have to be considered:

1. *Dynamic*: a torque is applied on the axle of the tire (plus the vertical gravitational load and a virtual mass of the car according to D'Alembert's principle).
2. *Kinematic*: an angular acceleration is applied, resulting in a reaction torque on the axle.

In both cases, the reaction-torque is a measure for the traction. Additional different kinematic constraints are possible:

1. *Realistic*: The rotation around the vertical axis (steering) is fixed, otherwise the tire may freely move.
2. *Experimental*: the rotational axis of the tire is fixed, at least in the horizontal direction.

If the axis is not fixed and the tire can freely move, it may, after a short period of simulation time, leave the sandbox (the computational domain). In this more realistic case an automated periodic replication of the sandbox in the driving direction would help to simulate a nearly unlimited track and keep the number of particles constant and thus the computational effort. This feature has not yet been implemented within the code SimPARTIX.

The present example shown below is a first more complex test, but still not a fully realistic simulation due to two restrictions:

1. The tire is modeled as a rigid body. It is not a co-simulation between SimPARTIX and Abaqus. Studies using an elastic tire model are currently under way.
2. A totally fixed axis was chosen, which firstly prevents the tire from leaving the sandbox and secondly avoids the tire to slump into the granular bed because of the vertical load.

A constant torque of 100 Nm is applied on the axis. Rather large cohesion is used to make the particles very sticky and to emulate mud or snow like characteristics. Heat exchange between tire and soil is neglected. Roughly 300,000 particles were used.

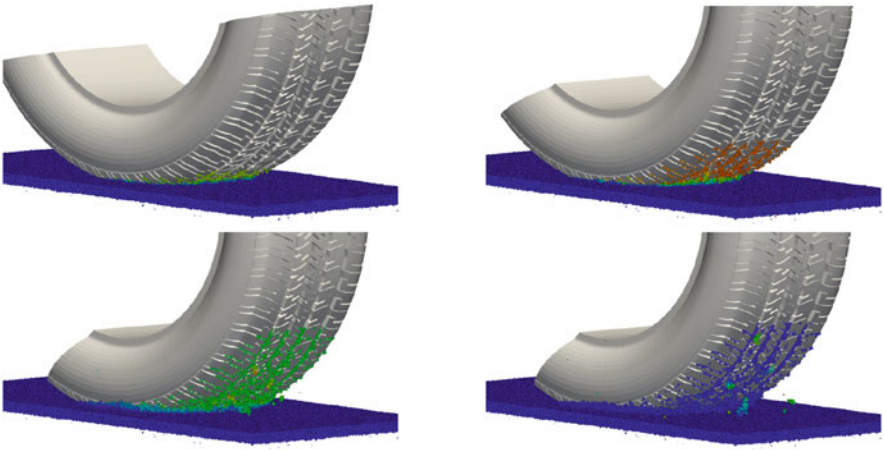


Fig. 12 Simulation snapshots

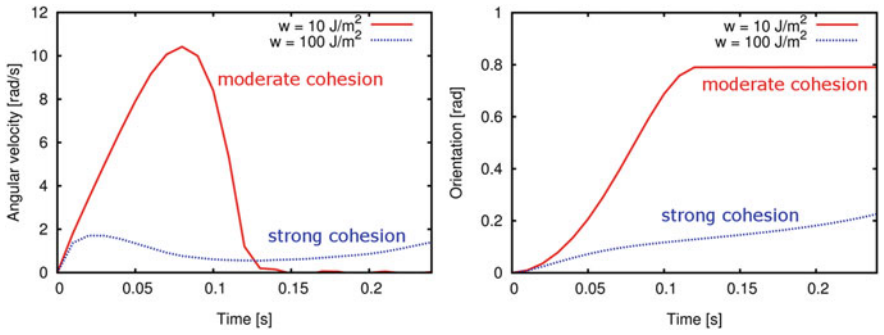


Fig. 13 Influence of the strength of granular cohesion on the tire dynamics

Figure 12 shows a series of simulation snapshots. The particles are color-coded according to their velocity magnitude ranging from blue (zero velocity), to green (medium velocity), and up to red (highest velocity). The principle effects on the behavior of a sticky medium can be reproduced. Some granular material remains in the groves of the tire tread.

The effect of cohesion within the granular bed is shown in Fig. 13. Even though the progression of the traction is quite different in both cases, there are some similarities. Initially the traction is nearly constant (constant angular acceleration) and the tire begins to slip due to a lack of traction. After about 10 ms with strong cohesion and roughly 60–70 ms with moderate cohesion the compaction of the granular material begins at the rear end of the footprint. The traction torque is larger than the axle torque and the tire decelerates. In the case of moderate cohesion the tire stops rotating at about 120 ms. At that point, the applied torque is not sufficient to overcome the traction caused by the compacted granular material. In the case of

strong cohesion the traction torque reaches the same magnitude as the applied torque (constant angular velocity), which indicates a nearly perfect traction. Apparently, the cohesion of the granular bed has a strong impact on the tire dynamics.

7 Conclusion

The simulation of the traction of a tire on soil or a snow covered road is a typical multidisciplinary application. One has to model the characteristics of soil or snow as well as the deformation of the spinning tire. While the modelling of an elastic tire, even made with complex rubber-material, seems no longer a major problem today, the mathematical models of the physics of soil and snow are still under development. The simulation of both physical systems is not covered by a single simulation code.

The presented solution of a co-simulation offers a way to overcome these deficiencies and to open new ways into a simulation world which hasn't been addressed up to now.

The presented test examples for the co-simulation between SimPARTIX and Abaqus as well as the application of the code SimPARTIX to the simulation of a tire starting on a sticky granular bed show promising results.

The future objective is a fully coupled simulation of an accelerating or retarding elastic tire. A long term vision is the integration of an antilock braking system modelled by a system code into the co-simulation process of a braked tire.

References

1. A.B. Basset, On the descant of a sphere in a viscous liquid. Q. J. Math. Lond. **41**, 369 (1910)
2. SimPARTIX – a particle based simulation software. www.simpartix.com, Mar 2014
3. C. Bierwisch, T. Kraft, H. Riedel, M. Moseler, Three-dimensional discrete element models for the granular statics and dynamics of powders in cavity filling. J. Mech. Phys. Solids **57**, 10–31 (2009)
4. M.P. Allen, D.J. Tildesley, *Computer Simulation of Liquids* (Oxford Science Publications, New York, 2006)
5. P.A. Cundall, O.D.L. Strack, A discrete numerical model for granular assemblies. Geotechnique **29**, 47–65 (1979)
6. K.L. Johnson, K. Kendall, A.D. Roberts, Surface energy and the contact of elastic solids. Proc. R. Soc. Lond. Ser. A Math. Phys. Sci. **324**, 301–313 (1971)
7. Abaqus – a general purpose FEA code, http://www.3ds.com/products-services/simulia/portfolio/abaqus/overview, Mar 2014
8. MpCCI – Mesh-based parallel Code Coupling Interface, http://www.mpcci.de, Mar 2014
9. Chr. Rümpler, Lichtbogensimulation für Niederspannungsschaltgeräte. Ph.D. thesis, Technical University of Ilmenau, 2009
10. F.N. Fritsch, R.E. Carlson, Monotone piecewise cubic interpolation. SIAM J. Numer. Anal. **17**, 238–246 (1980)

Efficient Neighbor Search for Particle Methods on GPUs

Patrick Diehl and Marc Alexander Schweitzer

Abstract In this paper we present an efficient and general sorting-based approach for the neighbor search on GPUs. Finding neighbors of a particle is a common task in particle methods and has a significant impact on the overall computational effort–especially in dynamics simulations. We extend a space-filling curve algorithm presented in Connor and Kumar (IEEE Trans Vis Comput Graph, 2009) for its usage on GPUs with the parallel computing model Compute Unified Device Architecture (CUDA). To evaluate our implementation, we consider the respective execution time of our GPU search algorithm, for the most common assemblies of particles: a regular grid, uniformly distributed random points and cluster points in 2 and 3 dimensions. The measured computational time is compared with the theoretical time complexity of the extended algorithm and the computational time of its reference single-core implementation. The presented results show a speed up of factor of 4 comparing the GPU and CPU run times.

Keywords Neighbor search • GPU • Meshfree methods and particle methods

1 Introduction

Particle methods are widely used today, especially in fluid dynamics. One of the earliest particle methods, smoothed-particle hydrodynamics (SPH) [10], was already developed in 1977 to simulate astrophysics problems and is applied today in many others areas of application. Other prominent examples of particle methods, are e.g. Molecular Dynamics (MD) [11] and Peridynamics (PD) [21], which are typically used in material science problems.

A common task in particle methods is the search for neighboring particles, since the discretization depends on the interactions of particles inside an interaction sphere or so-called horizon. Finding the neighborhood of each particle accounts for a large part of the overall computational time and due to the often weak convergence

P. Diehl (✉) • M.A. Schweitzer
Institute for Numerical Simulation, Wegelerstr. 6, 53115, Bonn, Germany
e-mail: diehl@ins.uni-bonn.de; schweitz@ins.uni-bonn.de

M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations VII*, Lecture Notes in Computational Science and Engineering 100,
DOI 10.1007/978-3-319-06898-5_5

properties of particle methods, very large particle clouds need to be employed in high fidelity simulations. To overcome the computational complexity $\mathcal{O}(n^2)$ of the naïve neighbor search, where we compute distances $||x_i - x_j||$ of each particle x_i to all other particles x_j, a number of approaches and numerical libraries exist.

One of the most renowned single-core libraries is ANN [14]. However, to deal with very large particle clouds, necessary for accurate simulations results today, the particle neighborhoods need to be computed in parallel. Here, also a number of libraries exist for shared-memory [2, 15, 24] and distributed-memory [7, 17] parallel computers. On Graphics Processing Units (GPU), however, this task is essentially an open question. Some algorithms [8, 12] and the "knn Cuda" [9] library is available for General Purpose Computation on GPU(GGPU). This library supports GGPU using the Compute Unified Device Architecture (CUDA), but is limited to 65,535 particles, which renders the library essentially useless in our setting, where we are concerned with particle numbers of $\mathcal{O}(10^6)$.

In this paper we present a generic GPU accelerated nearest neighbor search algorithm, which can be utilized in any particle method. Our algorithm rests upon the approach presented in [7]. According to [7] the algorithm is designed for better efficiency on multi-core machines, due to low memory usage and good cache efficiency. For GGPU low memory usage and good cache efficiency are important indicators of the performance on the GPU. These benefits of the algorithm and its complexity $\mathcal{O}(\lceil \frac{n}{p} \rceil m \log(m))$, where n denotes the number of particles, p the number of threads and m the number of neighbors, offers a good initial situation for an implementation on the GPU.

The remainder of this paper is structured as follows: In Sect. 2 we shortly introduce our reference particle method, peridynamics (PD), to describe the problem setting considered in this study. Note however that our approach is not restricted to PD simulations, but rather is applicable in general particle methods. In Sect. 3, we introduce the fundamental algorithm and discuss challenges in its extension to GPUs. Then, we present the measured run time of the GPU accelerated algorithm and compare these with the run time of the CPU implementation of the algorithm and our GPU accelerated implementation in Sect. 4. We conclude with the comparison of these implementations and a suggestion for use cases of the different implementations in Sect. 5.

2 Neighbors in Particle Methods

Figure 1 shows sketches of reference particle clouds considered in this paper. Initial particle configurations in simulations are often uniformly distributed. Thus, we consider a regular particle arrangement and a uniformly distributed, but irregularly spaced particle cloud. These initial particle clouds are redistributed at later time steps of the simulation by the respective particle method. Thus, we may encounter particle clouds with large variations in the particle density. Throughout this paper

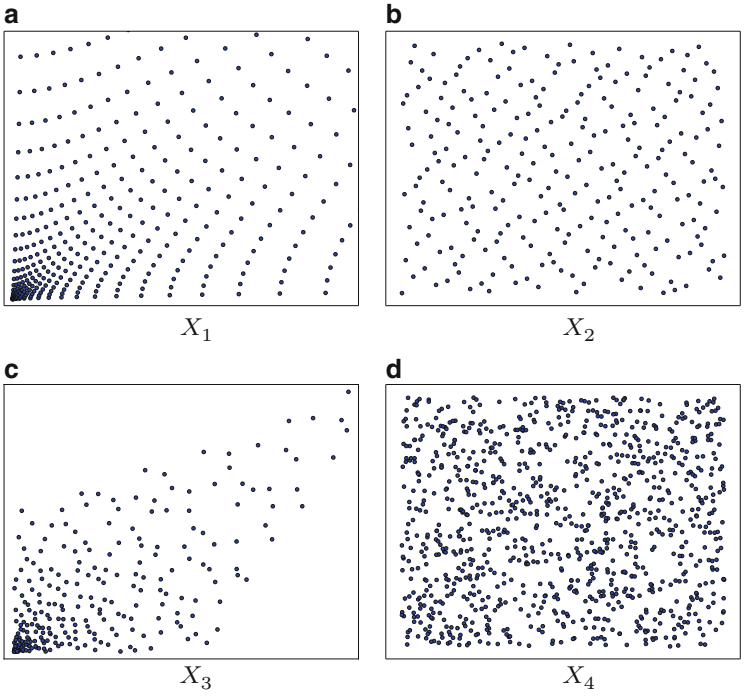


Fig. 1 Sketch of reference particle clouds: graded point cloud of a regular grid X_1 (**a**), points of the Halton sequence X_2 (**b**), graded Halton sequence X_3 (**c**) and uniformly distributed random points X_4 (**d**). Both point clouds were graded with $g(x) = ||x||^2 x$

we consider these reference cases, see also Fig. 1, for the evaluation of our search algorithm.

As a reference particle method we consider peridynamics (PD) [16,21,22], which is a non-local generalization of continuum mechanics, with a focus on discontinuous solutions as they arise in fracture mechanics. The principle of this theory is, that particles interact with other particles at a finite distance by exchanging forces – very similar to SPH and MD.

We present briefly the essential ingredients of the simple bond-based PD, which are important for the neighbor search algorithm. For further details see [16, 21, 22]. The PD equation of motion for the displacement field u is given by the integral equation

$$\varrho(x)\ddot{u}(x,t) = \int_{B_\delta(x)} f(u(x',t) - u(x,t), x' - x)dx' + b(x,t), \tag{1}$$

with mass density $\varrho(x)$, f as the kernel function modeling the interaction of particles x and x' in the initial reference configuration and $b(x,t)$ denotes the external force. Here $B_\delta(x)$ denotes the radial interaction zone, with the cut-off radius

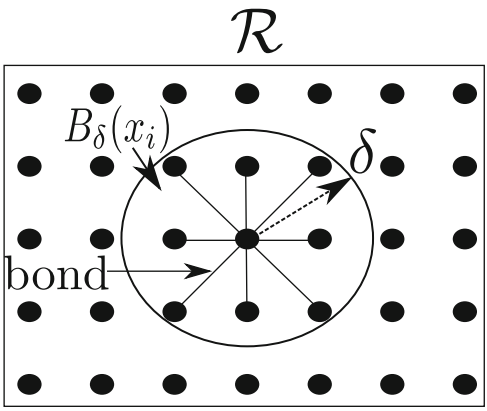


Fig. 2 Shows the reference configuration $\mathcal{R}$ of the particle cloud $X := \{x_i | i = 1, \ldots n\}$. All particles inside the interaction zone $B_\delta(x_i)$ of the particle x_i are connect with bonds to exchange forces

δ for the internal forces, see Fig. 2. As usual the deformed material configuration and instantaneous particle cloud $Y = \{y_i\}$ are obtained by $y_i(t) = x_i + u(x_i, t)$. Discretizing (1) in space by a collocation approach using the particle cloud $X = \{x_i\}$ yields

$$\varrho(x^i)\ddot{u}(x_i, t) = \sum_{j \in \mathcal{F}_i} f(u(x_j, t) - u(x_i, t), x_j - x_i)\tilde{V}_j + b(x_i, t), \tag{2}$$

with $\mathcal{F}_i = \left\{ j \mid ||x_j - x_i|| \leq \delta,\ i \neq j \right\}$, f as the kernel function and V_j as the volume associated with the particle x_j. We determine $\mathcal{F}_i$ in the reference configuration $\mathcal{R}$, see Fig. 2. All particles inside $B_\delta(x_i)$ are connected with an bond to x_i and they exchange forces through the kernel function f, which depends on the stretch of these bonds. $\tilde{V}_j$ denotes the scaled volume of particle x_j which is the intersection of $V_j \cap B_\delta(x_i)$. This set depends only on the initial positions of the point cloud and can be precomputed.

To prevent particle contact and overlap an additional force term f_s is introduced in the discrete model [16]. An example for the force term f_s is a "hard" potential, e.g. the repulsive part of the Lenard-Jones potential. The overall discrete PD model then reads as

$$\begin{aligned} \varrho(x_i)\ddot{u}(x_i, t) = &\sum_{j \in \mathcal{F}_i} f_b(u(x_j, t) - u(x_i, t), x_j - x_i)\tilde{V}_j \\ &+ \sum_{j \in \mathcal{F}^s_{i,t}} f_s(u(x_j, t) - u(x_i, t), x_j - x_i)V_j + b(x_i, t). \end{aligned} \tag{3}$$

Thus, a second set of interacting particles

$$\mathcal{F}^s_{i,t} = \left\{ j \mid ||y_j(t) - y_i(t)|| \leq \min(0.9||x_i - x_j||, 1.35r_i),\ i \neq j \right\}, \tag{4}$$

which employs that the instantaneous particle positions $y(t) = x + u(x, t)$ needs to be computed. This set denotes all particles which are around the particle x_i with the

radius r_i, due to deformation with respect to the instantaneous configuration and is at time $t = 0$ the empty set.

This set depends on the current time and must be updated in every time step during the simulation. Remember the computational complexity $\mathcal{O}(n^2)$ of the naïve approach, which would dominate the overall computational effort of most particle methods. Thus, this step must be realized in an extremely efficient way to allow for the simulation of large particle clouds.

3 Neighbor Search

In this section we shortly review the fundamental algorithm presented in [7] which will serve as the basis for our massive parallel implementation using CUDA. The basic idea is to sort the points with respect to a space–filling curve (SFC). This is a very successful and widely used approach for sorting multidimensional data with respect to topological information [1, 3, 4, 13, 19, 20, 25]. Such a sorting strategy is essentially realized via the following four steps (see also Algorithm 1).

Algorithm 1: Origin algorithm described in [7]. The steps highlighted in blue are discussed in this section, because these need to be adapted for the implementation on the GPU

Data: Morton order compare operator $\leq_M$ and Point cloud P
Result: m-nearest neighbors $\forall p_i \in P$
$P \leftarrow \text{ParallelSort}(P, \leq_M)$;
for $p_i \in P$ **do**
  $A_i \leftarrow nn^m(p_i, \{p_{i-m}, \ldots, p_{i+m}\})$;
  if $p_i^{\lceil rad(A_i) \rceil} < p_{i+m}$ **then**
    $u \leftarrow i$
  else
    $I \leftarrow 1$; **while** $p_i^{\lceil rad(A_i) \rceil} < p_{i+2^i}$ **do** $++I$; $u \leftarrow \min(i + 2^I, n)$;
  end
  if $p_i^{-\lceil rad(A_i) \rceil} > p_{i-m}$ **then**
    $l \leftarrow i$;
  else
    $I \leftarrow 1$; **while** $p_i^{-\lceil rad(A_i) \rceil} > p_{i-2^i}$ **do** $++I$; $l \leftarrow \min(i - 2^I, n)$;
  end
  if $l \neq u$ **then**
    CSEARCH(p_i, l, u);
  end
end

1. Generating keys: The multidimensional point cloud X is transformed with $T : \mathbb{R}^d \rightarrow \mathbb{R}$, which is the inverse of a SFC. Thus, a one dimensional key $k_i = T(x_i)$ is assigned to every particle $x_i \in X$.

2. Sorting keys: These one-dimensional keys k_i can easily be sorted and induce a respective ordering of the particles $x_i \in X$.
3. Range scan: Moreover it is easy to identify "neighbor keys" in one dimension to obtain a good initial guess $\tilde{N}(x_i)$ for the geometric neighbors in multidimensions of a particular particle x_i.
4. Geometrical validation: Finally, we need to check if all geometrical neighbors $x_j \in X$ of particle $x_i \in X$ are already found.

In the following we review this approach in more detail and discuss its implementation on a GPU. In a SFC-method, a multidimensional point $x \in \mathbb{R}^d$ is transformed to a (large) integer value $T(x) = k \in \mathbb{N}$, which can then be easily sorted. Thus, the transformation $T : \mathbb{R}^d \to \mathbb{N}$ is essentially influencing the quality of the resulting ordering of the data. There exists many different space filling curves, e.g. the Hilbert curve, the Peano curve, the Lebesgue curve, also referred to as Morton order, which can in principle be employed. One early application of the Morton order is efficient range searching of multidimensional data in dynamically balanced trees [24]. Using the Morton order, the transformation $T : \mathbb{R}^d \to \mathbb{N}$ is computationally cheap, but provides some sub–optimal locality. To clarify this let us consider the generation and comparison of keys for the Morton order in the following.

$$x = \begin{pmatrix} 0 \\ 3 \end{pmatrix}_{b=10} = \begin{pmatrix} 000 \\ 0111 \end{pmatrix}_{b=2} \xrightarrow{T(x):\mathbb{R}^2 \to \mathbb{N}} (000101)_{b=2} \tag{5}$$

Equation (5) shows the generation of the key $T(x)$ for the point $x = (0, 3)^T$. First the coordinates of the point in base 10 are converted to base 2 and then mapped to the scalar key $(x_0 y_0 x_1 y_1 x_2 y_2)_{b=2}$ in base 2 by bit-interleaving. Equation (6) shows the comparison of two points $x_1 = (0, 3)^T$ and $x_2 = (1, 2)^T$ using the Morton order compare operator $\leq_M$.

$$T(x_1) = T\begin{pmatrix} 0 \\ 3 \end{pmatrix} = (000101)_{b=2} \leq (000110)_{b=2} = T\begin{pmatrix} 1 \\ 2 \end{pmatrix} = T(x_2) \tag{6}$$

The computational effort of the comparison $T(x_1) \leq T(x_2)$ is small, since it involves only two operations: the exclusive-or operation and the most significant bit (MSB) operation delivers the result. Thus, the largest cost of the computational work associated with the Morton order compare operator

$$x_1 \leq_M x_2 \Leftrightarrow T(x_1) \leq T(x_2) \tag{7}$$

is associated with the respective computation of the respective keys $T(x_1)$ and $T(x_2)$.

Applying the Morton order compare operator to the nodes of a uniform 4×4 grid yields to the ordering depicted in Fig. 3. From this plot we find that we assign "adjacent" keys to the points $(3, 1)^T$ and $(0, 2)^T$. Thus, these points are "neighbors" with respect to the Morton order, but not with respect to their Euclidean

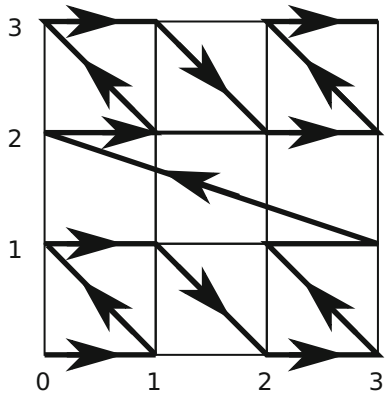


Fig. 3 The resulting space–filling curve with the Morton Order compare operator $\leq_M$ on a uniform grid with 4×4 points. Because of its shape, the *curve* is also called Z–Curve

distance. Other space-filling curves, e.g. the Hilbert curve, may provide a better data locality but the computational costs associated with the respective key generation or compare operator may be much larger.

We obtain an initial estimate of the m-nearest neighbors of a particle x_i by selecting $m/2$ particles x_j with the largest keys $k_j = T(x_j) \leq k_i = T(x_i)$ and $m/2$ particles with the smallest keys $k_j \geq k_i = T(x_i)$ and collect these keys in the set $\hat{N}(k_i)$. This initial guess $\tilde{N}(x_i) = \{x_j | T(x_j) = k_j, k_j \in \hat{N}(k_i)\}$ then needs to be validated, i.e. we need to check if $\hat{N}(k_i)$ in fact contains the particles x_j which are closest to x_i.

To this end, the Morton order divides a d-dimensional unit cell recursively in 2^d sub cells. In each sub cell there exist two keys $q = T(x)$ and $p = T(x')$, so that all particles x_l with $q \leq T(x_l) < p$ are included in this sub cell. To validate the initial guess $\tilde{N}(x_i)$, we have to check if the smallest key q and the largest key p are included in $\hat{N}(k_i)$. Otherwise the range of the set $\hat{N}(k_i)$ needs to be extended. Algorithm 2 describes the extension of the set using some geometrical information about the sub cells. Note that we use $\upsilon = 4$ as suggested in [7] and have not yet optimized this parameter for the GPU.

The described transformation T is so far suitable for point clouds with integer values as coordinates. With the extensions, described in [5, 7], the Morton Order $\leq_M$ compare operator handles floating point values and multidimensional input data with $d > 3$.

Thus, in summary we need to provide the operations to generate the keys and handle recursiveness in Algorithm 2 on the GPU. To provide the exclusive-or operation and the most significant bit (MSB) operation, because on a GPU, these two operation are not available in the standard library, we need to define a new data type for the IEEE 754 representation of float and double, to access the exponent and the mantissa of the respective double or floating values of the particles.

Another issue for the implementation on a GPU are the recursive calls of Algorithm 2, because the programming model on the GPU does not support "real" recursiveness in kernel functions. The kernel function is launched on the device

with a specified grid of blocks. Then, a device function is called, recursively, within a kernel function. Thus, some recursive calls need to be implemented with the help of device functions and the grid of blocks need to be adjusted, because of the Single Instruction Multiple Threads (SIMT) architecture, depending on the employed GPU.

Algorithm 2: Function *CSearch(...)* [7] extends the range of the set $\{x_{i-ck}, \ldots, x_{i+ck}\}$ with using some geometrical information about the generated sub cubes

```
CSearch(point p_i, int l, int h) ;
if h-l < υ then
    A_i = nn^m(p_i, {p_{i-m}, ..., p_{i-m}}) ;
    return
end
b = (h + l)/2 ;
A_i = nn^m(p_i, A_i ∪ p_b) ;
if dist(p_i, box(p_l, p_h)) ≥ rad(A_i) then
    return
end
if p_i < p_b then
    CSearch(p_i, l, b-1);
    if p_b < p_i^{⌈r(A_i)⌉} then
        CSearch(p_i, b+1, h);
    else
        CSearch(p_i, b+1, h);
    end
    if p_i^{-⌈r(A_i)⌉} < p_b then
        CSearch(p_i, l, b-1);
    end
end
```

3.1 Parallel Sorting

With the increasing popularity of GGPU many standard sorting algorithms are available for GPU [18, 23]. The library Thrust [26], a powerful library of parallel algorithms and data structures, was extended with CUDA support and is now integrated in the CUDA SDK. The Thrust library contains the data structure *vector*, to store the multidimensional input data, in our case the particle locations x_j, and an optimized parallel merge sort algorithm, which utilizes a user defined *StrictWeakOrdering comp* which we realize with the help of $\leq_M$, see Eq. (7).

4 Results and Discussion

To evaluate the implementation of the extended algorithm the measured computational time is compared with its theoretical complexity and with the run time of the C++ Library STANN [6], the reference implementation of our base algorithm [7].

In all presented experiments we choose $m = 168$ (if not stated otherwise) for 3D particle clouds which corresponds to the number of geometric neighbors x_j of a particle x_i, i.e. $||x_j - x_i|| \leq \delta = 3a$, on a regular lattice with the lattice constant a. For 2D points clouds we choose $m = 24$, respectively. The number of threads p per processor is determined by the available hardware. The Nvidia K20c we used throughout this paper contains a Kepler GK110 G chip set. The specification of the chip set describes 13 multiprocessors each with 2,048 threads. In a optimal case all 13 multiprocessor with all threads can be executed in parallel. Thus, we use $p = 13 \cdot 2{,}048 = 26{,}624$.

First we compare the measured run times with the theoretical complexity $\mathcal{O}(\lceil \frac{n}{p} \rceil m \log(m))$ [7], where n denotes the number of particles of the point cloud X, p the number of threads and m the number of neighbors. In Fig. 4 we consider particle clouds X from $n = 512$ up to $n = 2^{21}$, which corresponds to the memory limit of the GPU. From the depicted plots we can observe that our implementation shows the theoretical complexity with respect to n for most of the considered point clouds X: nodes of a regular grid X_0, nodes of a graded grid X_1 (Fig. 1a), Halton points X_2 (Fig. 1b), graded Halton points X_3 (Fig. 1c) and uniformly distributed random points X_4 (Fig. 1d). For X_0-X_3 we find the expected linear asymptotic behavior. Only or the uniformly distributed random points X_4 we see, the influence

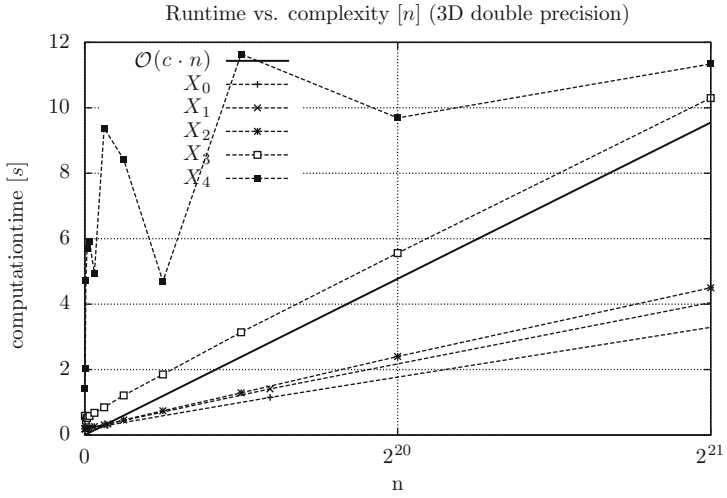


Fig. 4 Comparison of the theoretical complexity $\mathcal{O}(c \cdot n)$ with the measured computational time on the GPU

of the recursive calls in Algorithm 2 in the serrations of the computational time up to $n = 2^{20}$. For small particle clouds X_4 the particle density around some particles is low and to find all m neighbors the range of the initial guess $\tilde{N}(x_i)$ needs to be extended.

To confirm this assertion we consider $m = 24$ up to $m = 830$ for the point clouds X_1-X_4 with $n = 9{,}706$ and $n = 77{,}672$ in three dimensions. From these plots given in Fig. 5 we see for X_1-X_3 we attain the anticipated complexity with a very small constant. Only for X_4 we can again observe some oscillations. A more robust results is obtained with extending the initial estimate to the range $[-m, m]$ instead of $[-\frac{m}{2}, \frac{m}{2}]$. For the point cloud X_4 we see again that the range need to be extended.

Secondly the extended algorithm for the GPU is compared with the STANN library in the version 0.74 which does not support any parallel implementation for finding the m-nearest neighbors. The configurations shown in Table 1 were used to build the libraries. For the computation time on the GPU we measured the full work flow. This means that the measured computation time includes the copying process of the point cloud to the device and copying the resulting list of the m–nearest neighbors back to the host.

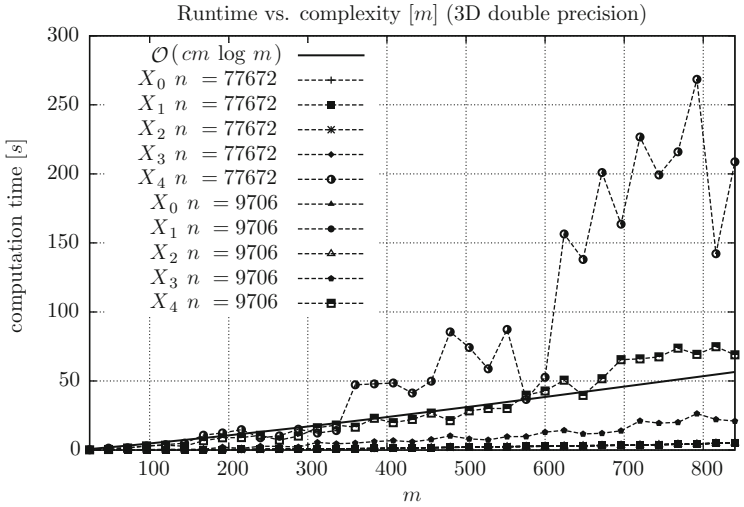


Fig. 5 Comparison of the theoretical complexity $\mathcal{O}(cm \log m)$ with the measured computational time on the GPU

Table 1 Build configuration of the libraries for the CPU and GPU

	CPU	GPU
Compiler	g++ (4.6.3)	nvcc (release 5.5, V5.5.0)
Compiler options	-o3	-arch=sm_35
Hardware	Intel Xeon 5500 (1 Core)	NVIDIA Tesla K20c

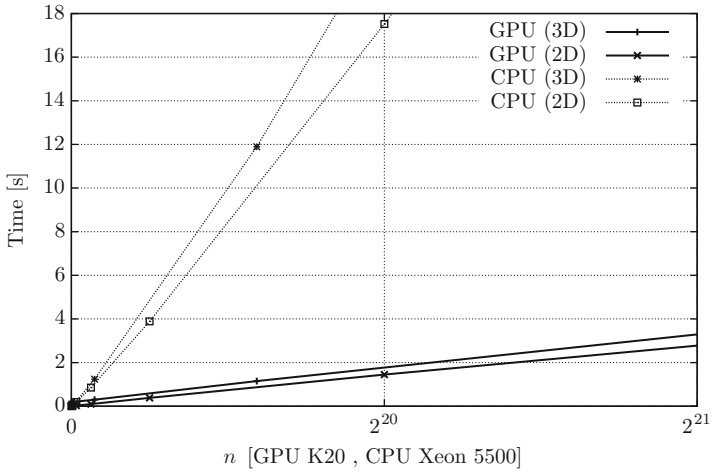


Fig. 6 Computational time for finding the nearest neighbor in the nodes of regular point grid X_0

The neighborhoods for all point clouds X_0, X_1, X_2, X_3 and X_4 were computed in two dimensions ($m = 24$) and three dimensions ($m = 168$) with double precision (*double*) floating point values. The computational time for the STANN library was measured on 1 core of an Intel Xeon 5500 CPU. The extended algorithm was executed on a Nvidia Tesla K20c GPU. The number of multidimensional input points is limited by the memory of the GPU, since our implementation of the presented algorithm does currently not support streaming techniques and thereby only single-GPU computing.

Figure 6 shows the measured computational time for the nodes of a regular grid X_0 with n from 512 up to $n = 2^{21}$. The depicted plots show that the search on the GPU is substantially faster than on a single core of the CPU. Moreover, we find that the computational time in two and three dimensions on the GPU are essentially identical which indicates that the additional operations in three dimensions come for free.

Note however, that for this perfectly homogeneous node arrangement the results are not very representative for the general case since here essentially no (expensive) geometric validation of the initial guess is necessary. Thus the initial guess $\tilde{N}(x_i)$ contains all geometrical neighbors with respect to the Euclidean distance.

The plots in Fig. 7 shows that for the graded nodes of a regular grid X the asymptotic behavior holds.[1] The difference between the computational time between the dimensions should be $\frac{24 \log 24}{168 \log 168} \approx 0.088$, excluding the computational

[1] Storing the graded nodes of the two dimensional regular grid in row-major order in a thrust vector leads to some issues in the costs of the Merge sort algorithm. To avoid this the nodes need to be stored randomized.

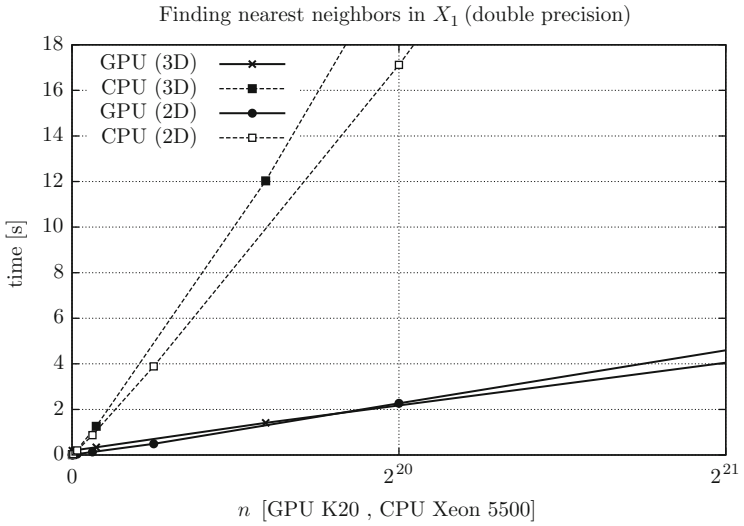


Fig. 7 Computational time for finding the nearest neighbor in a graded point cloud of a regular grid X_1 Fig. 1a

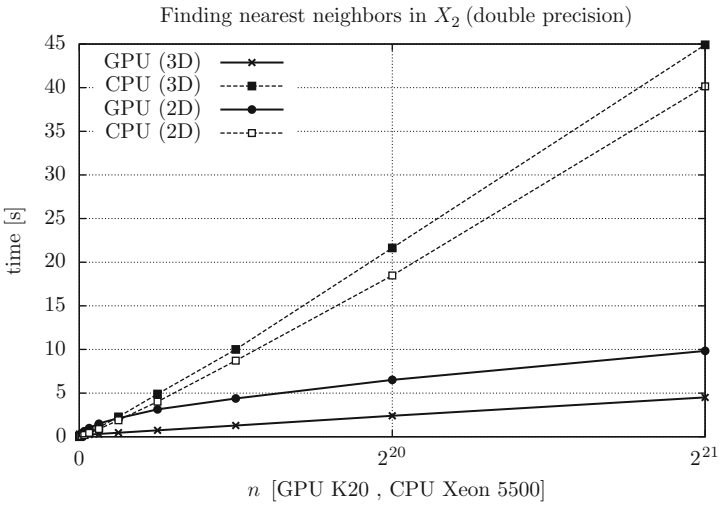


Fig. 8 Computational time for finding the nearest neighbor in the Halton sequence X_2 Fig. 1b

costs for the Euclidean distance in three dimensions. On the GPU we differ between the dimensions with the factor 0.01 in X_2.

Figure 8 shows the computational time for the Halton sequence X_2. Here we see a difference of 2.18 in the computational time. Here we see the sub-optimal locality of the curve for particles which are close in respect to the Morton order but not with respect to their Euclidean distances.

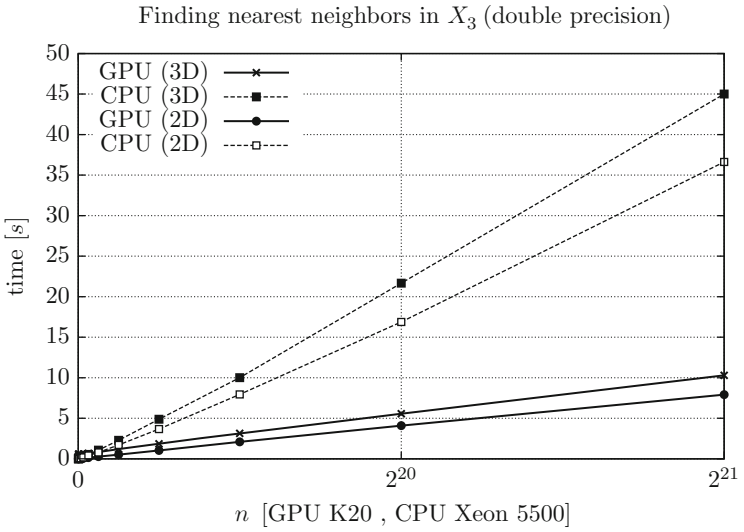


Fig. 9 Computational time for finding the nearest neighbor graded Halton sequence X_3 Fig. 1c

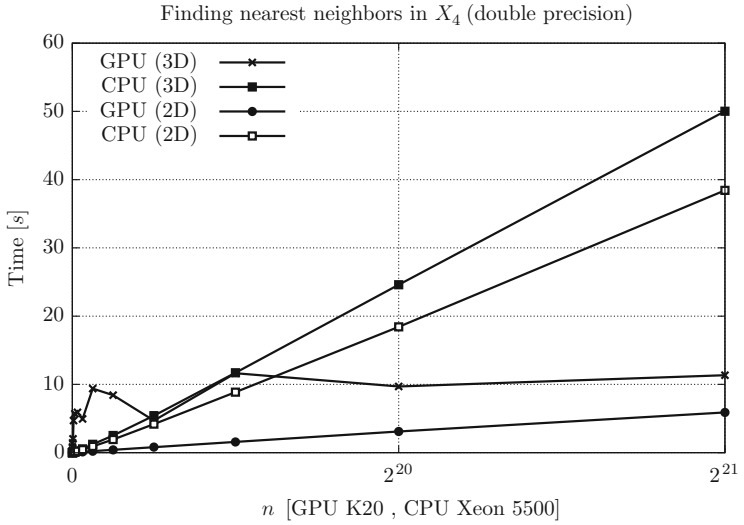


Fig. 10 Computational time for finding the nearest neighbor in a uniformly distributed random points X_4 Fig. 1d

Figure 9 shows the measured time for the graded Halton sequence X_3. Here we see a factor 0.77 between the computational time in two and three dimensions which is a factor of 10 away of the theoretical value.

Figure 10 shows the results for the point cloud X_4. Here we see artifacts of the differing particle density up to particle clouds with $n > 2^{21}$. Here the factor

between the dimensions is 0.52. In this point cloud we see that the computational time depends on the particle density. Particle clouds with differing density has a enormous effect on the computational time.

5 Conclusion

We presented an efficient neighbor search for particle clouds on GPUs. As a reference particle method we considered peridynamics. We used the basic idea of sorting points with respect to a space-filling curve as a massive parallel implementation on a GPU and described briefly the four steps essentially realizing such a sorting strategy. We validated our implementation against the theoretical complexity of the algorithm and compared the measured run times with the performance of the library STANN. We are in general faster by a factor of 4. Only for very small particle clouds X_4 we need more computational time. Using distributed points clouds, with a differing particle density, influence the computational time enormous, because of the extension of the range of the initial guess $\tilde{N}(x_i)$. The costs for copying the multidimensional point cloud X to the device and copying the resulting m neighbors back to the host does not influences the overall computational time. Thus, the library is suitable to update the neighborhood every time step in dynamic particle clouds.

To make the algorithm applicable in uncertainty quantification the dimension d of the point should be enlarged to $d > 3$. Advanced techniques, like data streaming or Multi-GPU computation, bypass the restriction of the memory of the GPU.

References

1. S. Aluru, F.E. Sevilgen, Parallel domain decomposition and load balancing using space-filling curves, in *Proceedings of the 4th IEEE Conference on High Performance Computing*, Bangalore, 1997, pp. 230–235
2. S. Arya, D.M. Mount, N.S. Netanyahu, R. Silverman, A.Y. Wu, An optimal algortihm for approximate nearest neighbor searching in fixed dimensions, in *Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, Arlington, 1994, vol. 5, pp. 573–582
3. M. Bader, *Space-Filling Curves – An Introduction with Applications in Scientific Computing* (Springer, Berlin/Heidelberg, 2013)
4. C. Böhm, S. Berchtold, A.D. Keim, Searching in high-dimensional spaces: index strucutres for improving the performance of multimedia databases. ACM Comput. Surv. **33**, 322–373 (2001)
5. T.M. Chan, A minimalist's implementation of an approximate nearest neighbor algorithm in fixed dimensions, https://cs.uwaterloo.ca/~tmchan/sss.ps, May 2006
6. M. Connor, P. Kumar, Stann, https://sites.google.com/a/compgeom.com/stann/
7. M. Connor, P. Kumar, Fast construction of k-nearest neighbor graphs for point clouds. IEEE Trans. Vis. Comput. Graph. **14**(4), 599–608 (2009)
8. A. Dashti, I. Komarov, R.M. D'Souza, Efficient computation of k-nearest neighbour graphs for large high-dimensional data sets on GPU clusters. PLoS ONE **8**, e74113 (2013), plosone.org
9. V. Garcia, E. Debreuve, M. Barlaud, kNN CUDA, http://vincentfpgarcia.github.io/kNN-CUDA/

10. R.A. Gingold, J.J. Monaghan, Smoothed particle hydrodynamics: theory and application to non-spherical stars. Mon. Not. R. Astron. Soc. **181**, 375–389 (1977)
11. M. Griebel, S. Knapek, G. Zumbusch, *Numerical Simulation in Molecular Dynamics* (Springer, Berlin/Heidelberg, 2007)
12. P. Leite, J.M. Teixeira, T. Farias, B. Reis, V. Teichrieb, J. Kelner, Nearest neighbor searches on the gpu. Int. J. Parallel Program. **40**(3), 313–330 (2012) (English)
13. J. Mellor-Crummey, D. Whalley, K. Kennedy, Improving memory hierarchy performance fir irregular applications using data and computation reorderings. Int. J. Parallel Program. **29**, 217–247 (2001)
14. D.M. Mount, S. Arya, ANN: a library for approximate nearest neighbor searching, http://www.cs.umd.edu/~mount/ANN/
15. S.A. Nene, S.K Nayar, A simple algorithm for nearest neighbor search in high dimensions. IEEE Trans. Pattern Anal. Mach. Intell. **19**, 989–1003 (1997)
16. M.L. Parks, R.B. Lehoucq, S.J. Plimpton, S.A. Silling, Implementing peridynamics within a molecular dynamics code. Comput. Phys. Commun. (EL, ed.) **179**, 777–783 (2008)
17. S. Plimpton, Fast parallel algorithms for short-range molecular dynamics. J. Comput. Phys. **117**, 1–19 (1995)
18. N. Satish, M. Harris, M. Garland, Designing efficient sorting algorithms for manycore GPUs, in *IEEE International Symposium in Parallel & Distributed Processing*, Rome, 2009, pp. 1–10
19. M.A. Schweitzer, *A Parallel Multilevel Partition of Unity Method for Elliptic Partial Differential Equations*. Lecture Notes in Computational Science and Engineering, vol. 29 (Springer, New York, 2003)
20. Y.D. Sergeyev, R.G. Strongin, D. Lera, *Introduction to Global Optimization Exploiting Space-Filling Curves* (Springer, New York/Heidelberg, 2013)
21. S.A. Silling, Reformulation of elasticity theory for discontinuties and long-range forces. Sandia report SAND98-2176, Sandia National Laboratories, 1998
22. S.A. Silling, E. Askari, A meshfree method based on the peridynamic model of solid mechanics. Comput. Struct. **83**, 1526–1535 (2005)
23. E. Sintorn, U. Assarsson, Fast parallel GPU-sorting using a hybrid algorithm. J. Parallel Distrib. Comput. **68**, 1381–1388 (2008)
24. H. Tropf, H. Herzog, Multidimensional range search in dynamically balanced trees. Angew. Inform. (Appl. Inform.) **2**, 71–77 (1981). Vieweg Verlag
25. M.S. Warren, J.K. Salmon, A parallel hashed oct-tree n-body algorithm, in *Proceedings of the 1993 ACM/IEEE Conference on Supercomputing (Supercomputing'93)*, Portland (ACM, New York, 1993), pp. 12–21
26. W. Wen-mei, *GPU Computing Gems Emerald Edition* Applications of GPU Computing Series, 1st edn. (Morgan Kaufmann, Burlington, Massachusetts 2011)

Robust Discretization of Nonlocal Models Related to Peridynamics

Qiang Du and Xiaochuan Tian

Abstract We study some nonlocal models related to peridynamics. These models are parameterized by a horizon that serves as a length scale measuring the range of nonlocal interactions. We focus on robust numerical schemes that can approximate both nonlocal models and their local limits as the horizon parameter vanishes. A representative linear model is used as an illustration. We show the lack of robustness of some standard numerical methods and describe a remedy to get asymptotically compatible schemes by utilizing elements of the recently developed nonlocal vector calculus and nonlocal calculus of variations. Such findings may be useful to ongoing research on modeling and simulations of nonlocal and multiscale problems.

Keywords Nonlocal models • Peridynamics • Meshfree methods • Local limit • Nonlocal calculus of variation • Finite difference • Finite element • Quadrature • Convergence analysis • Asymptotical compatibility

1 Introduction

Recent success of peridynamics (PD) in modeling materials singularities has renewed interests in the study of nonlocal continuum models. Peridynamics, proposed by Silling [21], is an integral-type nonlocal continuum model of materials. It has been applied to various materials systems [1, 13, 22, 25]. Linear scalar PD operators also share similarities with nonlocal diffusion operators, as pointed out in [5], thus making the study of PD relevant to the study of nonlocal diffusion (ND) models, see references in [26]. A distinct feature of PD and ND models considered here is the horizon parameter δ, introduced by Silling [21], that characterizes the range of nonlocal interactions. In PD/ND models, spatial differentiations are formally avoided with the integral formation of nonlocal forces and fluxes [7]. As $\delta \to 0$, nonlocal effect diminishes and the limiting process provides a bridge linking nonlocal models and classical (local) differential equation models, when the latter

Q. Du (✉) • X. Tian
Department of Mathematics, Pennsylvania State University, University Park, PA 16802, USA
e-mail: qdu@math.psu.edu; tian_x@math.psu.edu

M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations VII*, Lecture Notes in Computational Science and Engineering 100,
DOI 10.1007/978-3-319-06898-5_6

are well-defined, as shown formally in [12, 24] by Taylor expansions for smooth functions and more rigorously for functions with minimal regularity both by Fourier analysis [7, 11] and in bounded domains [17, 18, 28] with suitable topologies via the framework of nonlocal calculus of variations. The replacement of differential operators by integral nonlocal operators, and the strong connections between nonlocal models and their local limits, along with works on meshfree discretizations [22], have motivated many researchers to explore similarities between PD/ND models and other nonlocal continuum theory such as those reviewed in [4] and meshfree approximations of local PDE models such as those represented by Moving Least Square (MLS) approximations and Reproducing Kernel Particle Methods (RKPM) [14, 15] and Smoothed Particle Hydrodynamics (SPH) [19, 20]. In this broader context, one may see greater benefit in developing suitable mathematical framework and effective numerical algorithms that are applicable to both local and nonlocal models. Such new developments may further enable us to better model and simulate laws of nature.

Our objective here is to reiterate a special message on the discretization of nonlocal models such as peridynamics based on recent studies: we advocate the use of robust discretization schemes that work for both nonlocal models and their local limits. This view is supported by a series of systematic mathematical and numerical analysis of relevant nonlocal models. To elucidate the message, we discuss the connections between nonlocal and local models, continuum descriptions and discrete approximations. We then address the question on how to develop numerical discretizations of nonlocal models that yield consistent approximations in the local limit regardless how the nonlocal horizon and the mesh spacing are coupled together. Such schemes are called asymptotically compatible (AC) schemes [27]. The study of AC schemes, as some researchers in the field have agreed upon, is showing its potential impact on the numerical studies of peridynamic models and more broadly on the verification and validation of numerical simulations and nonlocal modeling. Theoretically, the recently developed framework of nonlocal vector calculus and nonlocal calculus of variations [5, 6, 16–18] provide a rigorous basis for demonstrating the robustness. Practically speaking, robust discretization is also very important as multiscale problems in nature often exhibit varying degrees of nonlocality.

The paper is organized as follows. In Sect. 2, we briefly recall the peridynamic models and some interesting features. In Sect. 3, we begin with some discussions about the notion of nonlocality in mathematical modeling and in discrete approximations. We then move on to discuss some simple discrete schemes in Sect. 4. In Sect. 5, we review some elements of the recently developed nonlocal vector calculus and provide a glimpse of the nonlocal calculus of variations in Sect. 6 using the linear bond-based PD model as an illustration. In Sect. 7, we discuss AC schemes as robust discretizations to both nonlocal models and their local limits. Though the discussions are presented for variational problems and Galerkin approximations, similar studies are also applicable to direct discretizations of the nonlocal models via quadrature formula and difference approximations as demonstrated in [26]. Some conclusions are given in Sect. 8.

2 Peridynamics and Nonlocal Balance Laws

In continuum mechanics, balance laws are often written as systems of partial differential equations. A typical example is given by the classical equation of motion:

$$\rho\,\ddot{\mathbf{y}}(\mathbf{x},t) = \nabla\cdot\boldsymbol{\sigma}(\mathbf{x},t) + \mathbf{b}(\mathbf{x},t)\,, \tag{1}$$

where $\mathbf{x}$ is the material point position, $\mathbf{y} = \mathbf{y}(\mathbf{x},t)$ the position of $\mathbf{x}$ at time t, $\ddot{\mathbf{y}}$ acceleration, ρ the mass density, and $\mathbf{b}$ external force. With $\mathbf{u} = \mathbf{y} - \mathbf{x}$ being the displacement, $\mathbf{F} = \nabla\mathbf{y} = \mathrm{I} + \nabla\mathbf{u}$ the deformation gradient, then $\boldsymbol{\sigma} = \hat{\sigma}(\mathbf{F})$ represents a constitutive model; for example, for elastic material, $\hat{\sigma}(\mathbf{F}) = W_{\mathbf{F}}(\mathbf{F})$ with the stored elastic energy density $W = W(F)$.

The Peridynamic (PD) theory takes on a form of differential-integral equations. It is proposed as an alternative to the classical continuum mechanics [21]. For example, the State-based PD equation of motion can be written as [23]

$$\rho\,\ddot{\mathbf{y}}(\mathbf{x},t) = \int \left\{\underline{\mathrm{T}}[\mathbf{x},\underline{\mathrm{Y}}](\mathbf{x}'-\mathbf{x}) - \underline{\mathrm{T}}[\mathbf{x}',\underline{\mathrm{Y}}'](\mathbf{x}-\mathbf{x}')\right\} dV_{\mathbf{x}'} + \mathbf{b}(\mathbf{x},t) \tag{2}$$

where, instead of introducing a deformation gradient, it adopts a notion of the deformation state $\underline{\mathrm{Y}}[\mathbf{x},t](\mathbf{x}'-\mathbf{x}) = \mathbf{y}(\mathbf{x}',t) - \mathbf{y}(\mathbf{x},t)$, and the PD force state $\underline{\mathrm{T}}[\mathbf{x},\underline{\mathrm{Y}}](\mathbf{x}'-\mathbf{x})$. The latter is an operator that maps from bonds between $\mathbf{x}$ and neighboring materials points $\mathbf{x}'$ to forces between $\mathbf{x}$ and $\mathbf{x}'$, as specified by a constitutive model. It is often assumed that $\underline{\mathrm{T}}[\mathbf{x},\underline{\mathrm{Y}}](\mathbf{x}'-\mathbf{x})$ vanishes when $\mathbf{x}'$ is outside a spherical neighborhood $B_\delta(\mathbf{x})$ of $\mathbf{x}$ with radius δ. The parameter δ is called the PD horizon. If we denote $\underline{\mathrm{T}} = \underline{\mathrm{T}}[\mathbf{x},\underline{\mathrm{Y}}](\mathbf{x}'-\mathbf{x})$ and $\underline{\mathrm{T}}' = \underline{\mathrm{T}}[\mathbf{x}',\underline{\mathrm{Y}}'](\mathbf{x}'-\mathbf{x})$, the balance law:

$$\frac{d}{dt}\int_\Omega \rho\dot{\mathbf{y}}\,dV_{\mathbf{x}} = \int_\Omega\int_{\mathbb{R}^n\setminus\Omega}\{\underline{\mathrm{T}}-\underline{\mathrm{T}}'\}\,dV_{\mathbf{x}'}\,dV_{\mathbf{x}} + \int_\Omega \mathbf{b}\,dV_{\mathbf{x}}$$

can be seen as a nonlocal analog/extension of the local counterpart:

$$\frac{d}{dt}\int_\Omega \rho\dot{\mathbf{y}}\,dV_{\mathbf{x}} = \int_{\partial\Omega}\boldsymbol{\sigma}\cdot\mathbf{n}\,d\Gamma + \int_\Omega \mathbf{b}\,dV_{\mathbf{x}}\,.$$

A nonlocal flux functional was introduced in [7] as

$$\mathcal{F}(\Omega_1,\Omega_2) = \int_{\Omega_1}\int_{\Omega_2}\{\underline{\mathrm{T}}-\underline{\mathrm{T}}'\}\,dV_{\mathbf{x}'}\,dV_{\mathbf{x}}\,,\quad \forall\,\{\Omega_i\subset\mathbb{R}^n\}_{i=1}^2$$

for any pair of domains Ω_1 and Ω_2, which shares properties of a local flux functional like additivity, action-reaction, and no-self-interaction. Moreover, it does not require the two domains to be in direct contact, thus broadens the notion of local flux to incorporate nonlocal interactions.

To briefly summarize, PD is a continuum model that represents a nonlocal balance law involving nonlocal fluxes and nonlocal operators; it avoids classical notions like the local flux and deformation gradient, thus possibly allowing more singular solutions. In the local limit, classical balance laws represented by PDE models can be recovered when such limits are physically meaningful, as demonstrated later. In addition, recent works have demonstrated that PD can be analyzed effectively via nonlocal vector calculus and nonlocal calculus of variations [6, 7, 16–18].

3 Nonlocality and Discretization

Having briefly described nonlocal peridynamic models, we now present some general discussions of local and nonlocal models and illustrate the interplay between nonlocality and discretization through a comparison of nonlocal operators and discrete operators.

So far we have largely attributed nonlocal models to those represented by differential-integral equations such as PD models. At the same time, the notion of local models has typically been reserved for partial differential equations. These names are used mostly for easy reference since arguably there is hardly any precise distinction between notions of locality and non-locality. It should be noted that, on one hand, local PDE models have historically been very popular in many scientific and engineering disciplines and have often appeared in elegant forms with the development of classical calculus; on the other hand, nonlocality is also ubiquitous in nature and nonlocal effects are generic due to model reduction and coarse graining. Incorporating nonlocality into the modeling process may turn out to be both convenient and necessary.

It is interesting to note that many nonlocal models may also be localized in an extended configuration space, or through approximations like moment closures. Moreover, numerical discretization and nonlocal interaction are often intertwined. Discretizations of partial differential operators in local models often induce some forms of nonlocality (on the discretized mesh scale), meanwhile, nonlocal operators may often be seen as weighted or averaged discrete operators [26]. It is beyond the scope of this work to address the origin and necessity of nonlocal models, but exploring connections and contrasts in these different models surely helps us further understand salient yet subtle features of nonlocality.

3.1 Nonlocality Due to Model Reduction

Generically, nonlocal features appear often in model reduction and coarse graining such as those following homogenization processes and the Mori-Zwanzig formalism. Here, we give two elementary examples that are widely known. First, we consider a local system for some coupled quantities of interest U_1 and U_2,

for instance, a system represented by a sparse matrix in block form. A standard elimination procedure leads to a reduced system associated with the so-called Schur complement

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} U_1 \\ U_2 \end{pmatrix} = \begin{pmatrix} F_1 \\ F_2 \end{pmatrix}, \quad \Rightarrow \quad (A_{11} - A_{12}A_{22}^{-1}A_{21})U_1 = F_1 - A_{22}^{-1}F_2 \,.$$

Even though (A_{ij}) can be local and sparse, $A_{11} - A_{12}A_{22}^{-1}$ is not in general. In a similar spirit, consider the boundary integral method for solving local PDEs, for example, $-\Delta u = 0$ in a two dimensional domain Ω with a compatible Neumann boundary condition $\partial_n u = g$ on $\Gamma = \partial\Omega$. We have the following integral equation of the second kind

$$-\pi u(x) + \int u(y)\partial_n G(x-y)dy = \int g(y)G(x-y)dy$$

where $G = G(x-y)$ is the Green's function. Thus, a local PDE problem is reduced to a nonlocal integral equation for the Dirichlet data on the boundary.

3.2 Discretizing Local Models Yielding Nonlocal Models

For illustration, consider a simple model PDE

$$u_t - u_x = 0 \tag{3}$$

with the spatial differentiation replaced by a difference:

$$u_t(x,t) - \frac{1}{h}(u(x+h,t) - u(x,t)) = 0, \tag{4}$$

or equivalently,

$$u_t(x,t) - \frac{1}{h}\int u(s,t)[\delta_{s-x-h} - \delta_{s-x}]ds = 0$$

with two Dirac-δ functions making up the kernel in the integral. This leads to a system of coupled ODEs defined on the whole real line (for all $x \in \mathbb{R}$). The singularly supported kernel may be further approximated or regularized by other kernels to get

$$u_t(x,t) - \int_{\mathbb{R}} u(x-s,t)\underline{\omega}(x,s,h)ds = 0\,, \tag{5}$$

which is a nonlocal continuum model characterized by a parameter $h > 0$.

The model equations represented by (3)–(5) are all continuum models. Nevertheless, Eq. (4) may be viewed as a discretization of the PDE (3) in the sense that an infinite limiting process of performing spatial differentiation is discretized by a finite process involving an algebraic combination of two function evaluations even though the ranges of independent variables are kept as continua. The latter distinguishes the discussion here from the traditional method of lines for discretizing PDEs, though (4) can surely be solved only at discretely sampled grid points $\{x_j = jh\}_{j=-\infty}^{\infty}$, which would be in line with more conventional practices. Similarly, (5) can be further discretized by adopting a numerical quadrature for the integral at some grid points or quadrature nodes.

The above discussion, to some degree, shares resemblance with similar ideas explored in SPH and RKPM and various nonlocal theories in mechanics [4] that often introduce a generic nonlocal state variable $\tilde{q} = \tilde{q}(x)$

$$\tilde{q}(x) = \int_{\Omega} W(x-s)q(s)ds$$

as a representation of a state variable $q = q(x)$ with a suitably chosen nonnegative weight function W. Although in most of these approaches, much focus has been given to discrete quadrature forms of the above representation. Consequently, the resulting approaches are more closely linked to numerical approximations of certain local models or PDEs, and a great deal of efforts have been devoted to study the convergence behavior of the numerical solution as the mesh gets refined. We note however that, it is also interesting to examine properties of nonlocal models with h being finite. For instance, while the local model (3) is a first order hyperbolic equation with a unit speed of propagation, the nonlocal model (4), for any finite value of the parameter $h > 0$, has an infinite speed of propagation, yet, it is distinctively clear that as $h \to 0$, the two models are consistent. Thus, the contrast of propagation speeds in two different regimes naturally motivates us to adopt a more general characterization of propagation speed that may capture common features of wave propagation in both local and nonlocal models.

3.3 Nonlocal Models Represented by Discrete Operators

Furthermore, nonlocal operators can be viewed as continuum forms of discrete operators. Taking for example the operator

$$\begin{aligned}\mathcal{L}_\delta u(x) &= 2\int_{-\delta}^{\delta} \big(u(x+s)-u(x)\big)\gamma_\delta(s)ds \\ &= \int_{-\delta}^{\delta} \big(u(x+s)-2u(x)+u(x-s)\big)\gamma_\delta(s)ds \\ &= 2\int_{0}^{\delta} \frac{u(x-s)-2u(x)+u(x+s)}{s^\alpha} s^\alpha \gamma_\delta(s)ds \end{aligned} \tag{6}$$

for an even kernel function γ_δ, and a parameter $\alpha \in [0, 2]$. The operator can be viewed as an averaged (continuum) difference operator. The fact that the average is taken over the range of $s \in (0, \delta)$ means that rather than a finite difference, we in fact have a *continuum of differences* if γ_δ is indeed supported over such a continuum region (or rather, it is more regular than singular measures representing finite combinations of Dirac-delta measures and their distributional derivatives). Or, one may say that the linear nonlocal model is a continuum combination of discrete schemes. Different properties of the nonlocal operators are naturally built into, or determined by, the kernels (weights) γ_δ. Quadrature approximations of integrals at discrete grid points lead naturally to a finite difference approximation schemes. For time-dependent equations, such an approximation can lead to a meshfree implementation as presented in [22] for PD models. Meanwhile, direct correspondence can also be made between $\mathcal{L}_\delta$ and classical differential operators as in [7] when γ_δ is formally taken to be derivatives of Dirac-delta measures. The local limit can also be recovered by taken a sequence of γ_δ with

$$2\int_0^\delta s^2\gamma_\delta(s)ds = C_\delta \to C_0 \in (0, \infty) \quad \text{as} \quad \delta \to 0 \tag{7}$$

to get the convergence of $\mathcal{L}_\delta$ to the local limit $C_0 \frac{d^2}{dx^2}$.

4 Approximating Nonlocal Models

With (6), it is intuitively clear that we may obtain a fully-discrete finite dimensional algebraic system by approximating the *continuum difference* represented by $\mathcal{L}_\delta$ by discrete finite differences through various quadrature approximations (thus the *quadrature based finite difference* discretization as named in [26]).

To first present a simple class of quadrature based difference approximation for $\mathcal{L}_\delta$, we consider discrete operators $\mathcal{L}^{\alpha,h}_{\delta,0}$ defined in [26] given by

$$\mathcal{L}^{\alpha,h}_{\delta,0}u_i = 2\sum_{m=1}^{r} \frac{u_{i-m} - 2u_i + u_{i+m}}{(mh)^\alpha} \int_{I_m} s^\alpha \gamma_\delta(s)ds\,. \quad i = 1, \ldots N\,, \tag{8}$$

where $\{u_i\}$ are approximations of $\{u(x_i)\}$, $\{I_m = ((m-1)h, mh)\}_{m=1}^r$ with $\delta = rh$. The freedom in picking the parameter $\alpha \in [0, 2]$ is provided to offer a wide choice of different approximations.

As illustrated in [26], the discrete operator in (8) is obtained from a simple Riemann sum like quadrature of the integral (6). We may consider other quadratures like the trapezoidal and Simpson's rules for the integral in (6). For example, we have a trapezoidal like rule given by

$$\mathcal{L}_{\delta,1}^{\beta,h} u_i = 2\sum_{m=1}^{r-1} \frac{u_{i-m} - 2u_i + u_{i+m}}{(mh)^\beta} \int_{I_m \cup I_{m+1}} \phi_m^1(s) s^\beta \gamma_\delta(s) ds + 2\frac{u_{i-r} - 2u_i + u_{i+r}}{r^\beta h^\beta} \int_{I_r} \phi_r^1(s) s^\beta \gamma_\delta(s) ds \tag{9}$$

for $i = 1, \ldots, N$, where $\{\phi_m^1\}$ are the standard piecewise linear hat basis function and the parameter $\beta = 0$ or 1 [26].

Both schemes (8) and (9) are shown to be convergent approximations to the nonlocal model associated with the operator (6) for a given horizon δ and as $h \to 0$ [26]. Schemes corresponding to (9) have higher order of convergence than those given by (8). All these schemes lead to linear systems with coefficient matrices being M-matrices. Moreover, these schemes are derived from direct discretizations of the nonlocal equations, or say, their strong forms via collocation and quadrature approximations. It is thus interesting to note that the discrete operator given by (9) with $\beta = 0$ has been shown to be completely equivalent to the discrete operator corresponding to Galerkin finite element approximations with piecewise constant elements [26]. The latter, in its most natural setting, is viewed as a discretization of the weak form of the nonlocal equation. This further illustrates the connections between different approaches in discretizing the nonlocal models, being based on either strong or weak forms. Such connections have been explored in the literature in the context of numerical PDEs, see [26] for more discussions.

While (8) and (9) are good discretizations of (6), their convergence properties in the local limit are much more subtle. For example, it was shown that a standard three point central difference of the local limit $C_0 \frac{d^2}{dx^2}$ is obtained from (8) with $\alpha = 2$ and (9) with $\beta = 1$ by using the kernel γ_δ chosen to satisfy (7). However, such nice consistency cannot be expected in general with other choices of the exponents α and β. Moreover, there are also potential problems when the horizon is coupled with the discretization parameter. The latter scenario is often encountered in practice when modeling and simulating multiscale problems and thus deserves more careful studies.

For illustration, we let $\delta = rh$ for a fixed integer r and take a special kernel $\gamma_\delta(r) = \delta^{-2} r^{-1}$ which leads to $C_\delta = C_0 = 1$, then following the calculation given in [26] for generic kernels, we get that the limit of (8) with $\alpha = 1$ as $h \to 0$ is in fact given by

$$C_{0,1}^r \frac{d^2}{dx^2} = \frac{r+1}{r} \frac{d^2}{dx^2},$$

which means that for any finite r, the scheme (8) with $\alpha = 1$ consistently overestimate the elastic constant of the correct limiting local elasticity model by a constant factor. This scaling factor depends on the choices of kernels and quadratures. Such a discrepancy was shown to be generic for the types of discretizations given in (8) and (9) (and thus for piecewise constant Galerkin finite element

approximations as well). This phenomenon has been observed by other research groups as issues mostly attributed to the quadrature approximations. To provide a mathematical understanding of such phenomenon, we will rely on the nonlocal calculus of variations framework developed for nonlocal models.

5 Nonlocal Vector Calculus Formulation of Linear Nonlocal Operators

The nonlocal vector calculus developed in [7] represents a systematic study of nonlocal models and nonlocal operators. The theory helps to elucidate the often contentious issue of boundary conditions or more properly nonlocal constraints. Various basic nonlocal operators have been defined as the basic building blocks. For example, for given unit vector $\boldsymbol{\alpha}(\mathbf{x}', \mathbf{x}) = (\mathbf{x}' - \mathbf{x})/|\mathbf{x}' - \mathbf{x}|$, we have the nonlocal divergence operator:

$$(\mathcal{D}\boldsymbol{\Psi})(\mathbf{x}) = \int_{\mathbb{R}^n} \big(\boldsymbol{\Psi}(\mathbf{x}, \mathbf{x}') + \boldsymbol{\Psi}(\mathbf{x}', \mathbf{x})\big) \cdot \boldsymbol{\alpha}(\mathbf{x}, \mathbf{x}')\, d\mathbf{x}' ,$$

for any 2-point tensor valued function $\boldsymbol{\Psi}: \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n\times m}$ and its adjoint:

$$(\mathcal{D}^*\mathbf{v})(\mathbf{x}, \mathbf{x}') = \big(\mathbf{v}(\mathbf{x}') - \mathbf{v}(\mathbf{x})\big) \otimes \boldsymbol{\alpha}(\mathbf{x}', \mathbf{x}) ,$$

for any vector valued function $\mathbf{v}: \mathbb{R}^n \to \mathbb{R}^m$ We let $\mathcal{G}^* = Tr(\mathcal{D}^*)$ for $m = n$, that is, $(\mathcal{G}^*\mathbf{v})(\mathbf{x}, \mathbf{x}') = \big(\mathbf{v}(\mathbf{x}') - \mathbf{v}(\mathbf{x})\big) \cdot \boldsymbol{\alpha}(\mathbf{x}', \mathbf{x})$.

Among various applications of the nonlocal vector calculus, it allows us to reformulate PD type linear nonlocal balance laws in ways similar to local linear PDE models. We give a couple of examples.

1. Nonlocal Laplacian. For a given symmetric kernel $\underline{\omega}(\mathbf{x}', \mathbf{x})$, a compositions of above operators gives,

$$-\mathcal{D}(\underline{\omega}\, \mathcal{D}^*\mathbf{u}) = \int \underline{\omega}(\mathbf{y}, \mathbf{x})(\mathbf{u}(\mathbf{y}) - \mathbf{u}(\mathbf{x}))\, d\mathbf{y}$$

   which is a scalar nonlocal Laplacian (nonlocal diffusion operator) $\mathcal{L} = -\mathcal{D}(\underline{\omega}\, \mathcal{D}^*)$, a nonlocal analog of the classical operator $\nabla \cdot (\omega\, \nabla)$. Often, the kernel function can be taken to be $\underline{\omega} = \omega_\delta(|\mathbf{y} - \mathbf{x}|)$ with ω_δ having a compact support over $|\mathbf{y} - \mathbf{x}| \le \delta$ (horizon parameter) and it is suitably scaled so that

$$-\mathcal{D}(\omega_\delta\, \mathcal{D}^*) \to \nabla \cdot (K\nabla) , \quad \text{as} \quad \delta \to 0$$

   for a diffusion coefficient K.

2. Linear bond-based PD operator $\mathcal{L}_b = -\mathcal{D}(\underline{\omega}_\delta \mathcal{G}^*\mathbf{I})$ with $\mathbf{I}$ being the identity tensor, whose local limit is given by $\mu\nabla\cdot\nabla\mu\nabla\nabla\cdot$, a special form of the Navier operator with Poisson ratio $1/4$.
3. PD Navier operator: $\mathcal{L}_s\mathbf{u} = -\eta\mathcal{D}(\underline{\omega}_\delta \mathcal{G}^*\mathbf{I}) - \sigma\mathcal{D}_{\omega_\delta}\big(\mathrm{Tr}\big(\mathcal{D}^*_{\omega_\delta}(\mathbf{u})\big)\mathbf{I}\big)$ where

$$\omega_\delta(\mathbf{x},\mathbf{y}) = \frac{n}{m(\mathbf{x})}\underline{\omega}_\delta(|\mathbf{x}-\mathbf{y}|)|\mathbf{x}-\mathbf{y}| \text{ and } m(\mathbf{x}) = \int_\Omega \underline{\omega}_\delta(|\mathbf{x}-\mathbf{y}|)|\mathbf{x}-\mathbf{y}|^2 d\mathbf{y}\,,$$

the weighted operators are defined by

$$\mathcal{D}_{\omega_\delta}\big(\mathbf{U}\big)(\mathbf{x}) = \mathcal{D}\big(\omega_\delta(\mathbf{x},\mathbf{y})\mathbf{U}(\mathbf{x})\big)(\mathbf{x}), \tag{10}$$

$$\mathcal{D}^*_{\omega_\delta}\big(\mathbf{u}\big)(\mathbf{x}) = \int_{\mathbb{R}^n} \mathcal{D}^*(\mathbf{u})(\mathbf{x},\mathbf{y})\,\omega_\delta(\mathbf{x},\mathbf{y})\,d\mathbf{y} \tag{11}$$

which map (1-point) functions to themselves and they correspond to the classical divergence and gradient operator in the local limit [7] and as $\delta\to 0$,

$$-\eta\mathcal{D}\big(\underline{\omega}_\delta\mathcal{G}^*\mathbf{I}\big) - \sigma\mathcal{D}_{\omega_\delta}\big(\mathrm{Tr}\big(\mathcal{D}^*_{\omega_\delta}\big)\mathrm{I}\big) \to \mu\nabla\cdot\nabla(\mu+\lambda)\nabla\nabla\cdot\,.$$

It is worthwhile to note that $\mathcal{D}$ and $\mathcal{D}^*$ are maps between functions with different independent variables as they mimic the local transforms between differential forms of different orders. On the other hand, the weighted operators $\mathcal{D}_\omega$, $\mathcal{D}^*_\omega$ are actually more aligned with classical derivative operators defined on vector fields, which are the local limits of $\mathcal{D}_\omega$ and $\mathcal{D}^*_\omega$ as demonstrated in [7]. More studies along these lines are currently underway. Symbolically, one may recover $\mathcal{D}^*$ from $\mathcal{D}^*_\omega$ by picking a special weight $\underline{\omega}_\delta$ as a Dirac-delta function at a point away from $\mathbf{x}$.

There are more extensive discussions of $\mathcal{D}$, $\mathcal{D}^*$, $\mathcal{D}_\omega$, $\mathcal{D}^*_\omega$, and $\mathcal{G}$, $\mathcal{G}^*$, $\mathcal{C}$, $\mathcal{C}^*$ (nonlocal curl's) in [7], along with nonlocal Green's identities such as

$$\int_{\mathbb{R}^n} \mathbf{u}\,\big(\mathcal{D}\big(\underline{\omega}\big(\mathcal{D}^*\mathbf{v}\big)\big)\big)dx - \int_{\mathbb{R}^n}\big(\mathcal{D}\big(\underline{\omega}\big(\mathcal{D}^*\mathbf{u}\big)\big)\big)\mathbf{v}dx = 0\,,$$

$$\int_{\mathbb{R}^n} \mathbf{u}\,\mathcal{D}_{\omega_\delta}\big(\mathcal{D}^*_{\omega_\delta}\mathbf{v}\big)dx = -\int_{\mathbb{R}^n}\big(\mathcal{D}^*_{\omega_\delta}\mathbf{u}\mathcal{D}^*_{\omega_\delta}\mathbf{v}\big)dx = \int_{\mathbb{R}^n}\mathcal{D}_{\omega_\delta}\big(\mathcal{D}^*_{\omega_\delta}\mathbf{u}\big)\mathbf{v}dx\,.$$

6 Nonlocal Calculus of Variations

Building upon the foundation of the nonlocal vector calculus, we may further develop the theory of nonlocal calculus of variations. For example, consider a linear variational problem related to the nonlocal (bond-based PD) model $-\mathcal{L}_\delta\mathbf{u} = \mathbf{b}$ defined on a domain Ω with

$$\mathcal{L}_\delta \mathbf{u} = -\mathcal{D}(\underline{\omega}_\delta \mathcal{G}^* \mathrm{I})(\mathbf{u}) = \int_{\mathbb{R}^n} \underline{\omega}_\delta(|\mathbf{y}-\mathbf{x}|) \frac{\mathbf{y}-\mathbf{x}}{|\mathbf{y}-\mathbf{x}|} \otimes \frac{\mathbf{y}-\mathbf{x}}{|\mathbf{y}-\mathbf{x}|} \big(\mathbf{u}(\mathbf{y}) - \mathbf{u}(\mathbf{x})\big)\, d\mathbf{y}\,.$$

We may take Ω_I as the constraint set of nonzero measure (i.e., it is not a lower-dimensional boundary) to impose the $\mathbf{u} = 0$ condition on Ω_I. An example is $\Omega_I = \{\mathbf{x} \in \Omega^c : \mathrm{dist}(\mathbf{x}, \partial\Omega) < \delta\}$, a δ-layer surrounding Ω.

For a kernel $\underline{\omega}_\delta$ that satisfies

$$\underline{\omega}_\delta(r) \geq 0, \forall r \in (0,\delta), \quad \mathrm{supp}(\underline{\omega}_\delta) \subset B_\delta(0), \quad \int_{\mathbb{R}^n} |\mathbf{x}|^2 \underline{\omega}_\delta(|\mathbf{x}|) d\mathbf{x} < \infty \tag{12}$$

where the last condition implies that $\underline{\omega}_\delta$ has finite second order moments (which is necessary for well-defined elastic moduli), we have the following

$$\text{Energy space:} \quad V = \{\mathbf{u} \in L^2(\Omega) \mid |\mathbf{u}|_v^2 = \int_\Omega \int_\Omega \underline{\omega}_\delta\, (\mathcal{G}^*\mathbf{u})^2 d\mathbf{y} d\mathbf{x} < \infty\};$$

$$\text{Bilinear form:} \quad B_\delta(\mathbf{u},\mathbf{v}) = \int_\Omega \int_\Omega \underline{\omega}_\delta (\mathcal{G}^*\mathbf{u})(\mathcal{G}^*\mathbf{v})\, d\mathbf{y} d\mathbf{x}\,;$$

$$\text{Solution space:} \quad V_{c,\delta} = \{\mathbf{u} \in V \mid \mathbf{u} = 0 \text{ in } \Omega_I\} \subset V\,;$$

$$\text{Weak form:} \quad \text{find } \mathbf{u} \in V_{c,\delta} \text{ such that } B_\delta(\mathbf{u},\mathbf{v}) = (\mathbf{b},\mathbf{v})\,, \quad \forall\, \mathbf{v} \in V_{c,\delta}\,.$$

Various properties of the function spaces V, $V_{c,\delta}$ and the bilinear form B_δ can be studied in both specific spaces that are identified with either L^2 spaces or fractional Sobolev spaces, and general abstract spaces that are defined via energy functionals [18]. These studies lead to results like

- V, equipped with the norm $\{\|\mathbf{u}\|_{L^2}^2 + |\mathbf{u}|_v^2\}^{1/2}$, is a Hilbert space. Moreover, the embeddings: $H^1 \hookrightarrow V \hookrightarrow L^2$ are continuous.
- The solution space $V_{c,\delta} = \{\mathbf{u} \in V \mid \mathbf{u} = 0 \text{ in } \Omega_I\}$ is a subspace of V that is closed and dense in L^2 (thus separable), containing all functions in $H_0^1(\Omega)$ with zero extensions on Ω_I as a dense subspace.
- $Z = \{\mathbf{u} \in V \mid B_\delta(\mathbf{u},\mathbf{u}) = 0\}$ contains only rigid body displacement.
- Nonlocal Poincare's inequality: for any $\mathbf{u} \in V_{c,\delta}$, $|\mathbf{u}|_v^2 \geq c\|\mathbf{u}\|_{L^2}^2$, for some generic constant $c > 0$, which implies the coercivity of $B_\delta(\cdot,\cdot)$.
- Well-posedness of the variational problem: given $\mathbf{b} \in V_{c,\delta}^*$ (the dual space of $V_{c,\delta}$), find $\mathbf{u} \in V_{c,\delta}$, such that $B_\delta(\mathbf{u},\mathbf{v}) = (\mathbf{b},\mathbf{v})$ for any $\mathbf{v} \in V_{c,\delta}$.
- The embeddings and Poincare constants are uniform in δ (for δ less than some δ_0) if

$$\underline{\omega}_\delta(r) = \frac{1}{r^2\delta^d}\hat{\omega}\big(\frac{r}{\delta}\big)\,, \quad \underline{\omega}_\delta'(r) \leq 0\,, \quad \int \hat{\omega}(|\mathbf{x}|) d\mathbf{x} = 1\,. \tag{13}$$

- *Asymptotically compact embedding* property: given the above $\underline{\omega}_\delta$, any sequence $\{u_\delta\}$ uniformly bounded in $V_{c,\delta}$ with respect to $\delta \to 0$ is also relatively compact in L^2 with each limit point in H^1.

The theory of nonlocal calculus of variations can be used in various settings. For example, let us consider a family of closed conforming subspaces $\{W_{\delta,h} \subset V_{c,\delta}\}$ parametrized by an additional parameter $h \in (0, h_0]$ representing the typical mesh size. Although the subspaces $\{W_{\delta,h}\}$ do not always need to assume a finite dimensional form [27], we only consider here the situation that they are primarily discrete function spaces of finite dimension. Then, the nonlocal calculus of variations immediately allows a systematic analysis of conforming Galerkin approximations: find $\mathbf{u}^h \in W_{\delta,h} \subset V_{c,\delta}$, such that

$$B(\mathbf{u}^h, \mathbf{v}^h) = (\mathbf{b}, \mathbf{v}^h), \quad \forall \mathbf{v}^h \in W_{\delta,h}.$$

Some typical results include: for $\underline{\omega}_\delta$ satisfying (12) and a given horizon $\delta > 0$, we have the best approximation property

$$\|\mathbf{u} - \mathbf{u}^h\|_v \leq \inf_{\mathbf{v}^h \in W_{\delta,h}} \|\mathbf{u} - \mathbf{v}^h\|_v, \quad \text{as } h \to 0.$$

Further error analysis can be made for standard FEM as in the case presented in [5, 28]. Similar conclusions remain valid when $W_{\delta,h} \subset V_{c,\delta}$ is made of other closed subspaces associated with XFEM, GFEM, PUM or other Meshfree methods. One may generalize to the Galerkin-Petrov setting so that either the strong form or a generalized weak form of the nonlocal equations can be used for the discretization. For a posterior error analysis and convergent conforming adaptive finite element algorithms, we refer to [9, 10]. More works on nonconforming methods and mixed methods are currently underway.

Extensions to more general state-based peridynamic models can also be made based on the analysis in [17]. Similar analysis can also be applied to recently studied models of nonlocal diffusion and convection [8].

7 Asymptotically Compatible Schemes for Nonlocal Models and Their Local Limit

In [27], asymptotically compatible schemes are studied for general parametrized problems. Below, we present a definition for the special case of nonlocal peridynamic models and their local limits with the nonlocal horizon being the parameter.

Definition 1. A family of approximations $\{u_{\delta,h}\}$ for nonlocal problems parametrized by the horizon δ and mesh size h is said to be asymptotically compatible (AC) to the solution $u_{0,0}$ of the local limit, if for any sequence $\delta_n \to 0$ and $h_n \to 0$, we have $\|u_{\delta_n,h_n} - u_{0,0}\|_{L^2} \to 0$, as $n \to \infty$.

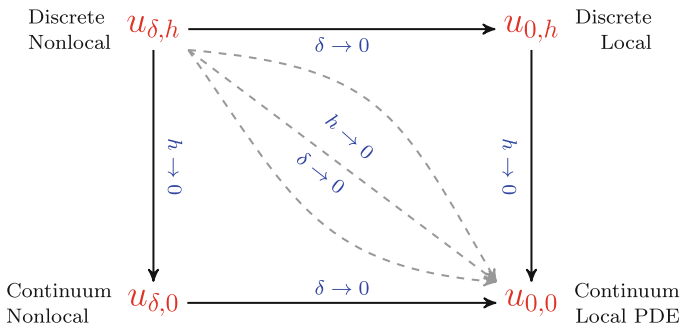


Fig. 1 Various convergence issues and limits as $\delta, h \to 0$

With both δ and h as parameters, there are various limiting processes as outlined in Fig. 1. Among these processes of interests, the limit of discrete approximation $u_{0,h}$ to the solution $u_{0,0}$ of local PDE problems, as the mesh is refined ($h \to 0$), has perhaps been most widely studied, as part of the subject on numerical PDEs. The limit as $\delta \to 0$ of solutions $u_{\delta,0}$ of nonlocal model to the local limit $u_{0,0}$ is a consequence of the nonlocal calculus of variations as described in the previous section. The other limiting cases all involve discretizations of nonlocal models.

Analytically, the keys to establish AC schemes for linear nonlocal variational problems are the uniform embeddings and the asymptotically compact embedding properties associated with the energy spaces.

We denote the local limit of nonlocal operators $\mathcal{L}_\delta$ by $\mathcal{L}_0$. For a finite $\delta_0 > 0$, we need the following assumptions that include, first of all, conditions ensuring the convergence of approximations to elements of $V_{c,\delta}$ as $h \to 0$ for each fixed $\delta \in (0, \delta_0]$, while the second part is concerned with the limiting behavior as both $h \to 0$ and $\delta \to 0$ at the same time.

Assumption 2. *Assume that the family of subspaces $\{W_{\delta,h} \subset V_{c,\delta}\}$ satisfies the following properties.*

(i) For each $\delta \in (0, \delta_0]$, the family $\{W_{\delta,h}, h \in (0, h_0]\}$ is dense in the energy space in the sense that, $\forall v \in V_{c,\delta}$, there exists a sequence $\{v_n \in W_{\delta,h_n}\}$ with $h_n \to 0$ as $n \to \infty$, such that

$$\|v - v_n\|_v \to 0 \quad as \quad n \to \infty\,. \tag{14}$$

(ii) $\{W_{\delta,h}, \delta \in (0, \delta_0], h \in (0, h_0]\}$ is asymptotically dense *in H_0^1, i.e., $\forall v \in H_0^1$, there exists a sequence $\{v_n \in W_{\delta_n,h_n}\}_{h_n \to 0,\, \delta_n \to 0}$ as $n \to \infty$, such that*

$$\|v - v_n\|_{H^1} \to 0 \quad as \quad n \to \infty\,. \tag{15}$$

As seen from the previous discussion and more detailed studies given in [26], there are finite element approximations to the nonlocal diffusion models that are

not AC. In particular, if δ is taken to be proportional to h, then as $h \to 0$, the piecewise constant conforming finite element solutions actually converges to the wrong limit. It is of practical interests to provide some constructive remedies to avoid such undesirable effects. Indeed, in [27], it has been shown that as long as the condition $h = o(\delta)$ is met as $\delta \to 0$, then we are able to obtain the correct local limit even for discontinuous piecewise constant finite element approximations when they are of conforming type, which corresponds to the schemes given in (8) with $\alpha = 1$ and (9) with $\beta = 0$.

Meanwhile, it has been shown in [27] that all conforming Galerkin approximations of the nonlocal models containing continuous piecewise linear functions are automatically AC. Naturally, these schemes turn to provide higher order accuracy to the nonlocal problems, yet we emphasize that the advantage is not limited to the gain in accuracy (which is generally applicable only with smooth functions) but also to the robustness in offering consistent solutions to the desirable physical models regardless how the modeling and discretization parameters are coupled. The latter is one of the main findings of our recent studies.

It should be noted that the above results are established for linear variational problems with interaction kernels satisfying (12) and (13) with homogeneous nonlocal constrained value conditions. Nevertheless, the discussions are quite general, for example, they also hold for linearized state-based peridynamic models as illustrated in [27]. The same notion is independent of the weak formulations and is applicable to collocation and quadrature based methods. There is also the flexibility to adapt the theory for other forms of approximations such as those based on the meshfree spaces and reproducing kernel spaces [14, 15] and Smoothed Particle Hydrodynamics (SPH) [19, 20].

Numerical experiments have been conducted for a 1d nonlocal problem $-\mathcal{L}_\delta u = f$ on $(0, 1)$ with the nonlocal constraint $u = 0$ outside $(0, 1)$ and a special kernel is chosen to be $\gamma_\delta(s) = \delta^{-2}s^{-1}$. Detailed reports can be found in [27]. We present some plots of pointwise errors between the limiting local solution and solutions of nonlocal problems using discontinuous piecewise linear finite element methods. The diminishing errors indicate the numerical convergence regardless how δ vanishes with respect to the mesh size h, demonstrating the robustness of numerical approximations (Figs. 2–4).

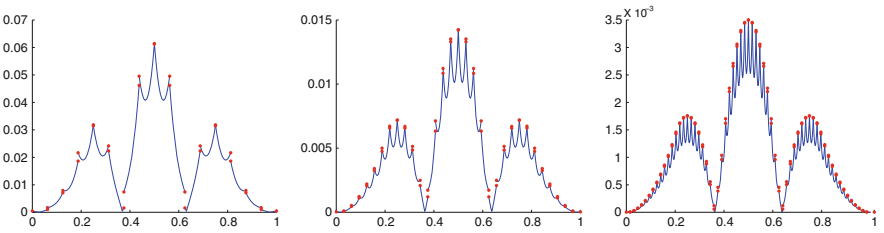


Fig. 2 Pointwise error $|u_{\delta,h}(x) - u_{0,0}(x)|$ with $r = \frac{\delta}{h} = 3$ and $h = 2^{-k}$, $k = 4, 5, 6$

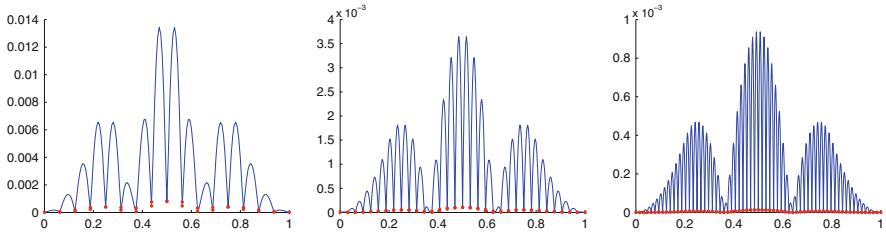


Fig. 3 Pointwise error $|u_{\delta,h}(x) - u_{0,0}(x)|$ with $\delta = h^2$ and $h = 2^{-k}$, $k = 4, 5, 6$

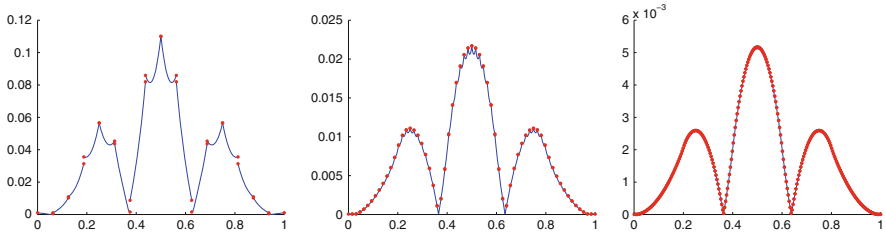


Fig. 4 Pointwise error $|u_{\delta,h}(x) - u_{0,0}(x)|$ with $\delta = \sqrt{h}$ and $h = 2^{-k}$, $k = 4, 6, 8$

8 Conclusions

Much of the works presented here are based on several of our recent works and they serve two purposes: one is to argue for necessity of nonlocal models and the intimate connections between nonlocality and discrete models, while the other is to raise the awareness that coupling the scale of nonlocality and the mesh spacing can sometimes be risky and thus calling for a more general mathematical frameworks for its analysis. These observations have more universal values beyond the study of peridynamics. Indeed, discussions of the nonlocality introduced in the physical models and in discretizations have been made in many prior works. For example, in [4], a historical perspective has been given including discussions of works by Bazant and Belytschko to incorporate nonlocality into traditional PDE models [2,3]. It raises the point that in a finite element solution, the element size then serves as a length scale so that mesh refinement tends to localizes the nonlocal regularization effects and making the computation mesh-sensitive. Such issues also served as strong motivation to the meshfree methods that shared an essential feature: *the approximation functions are not locally constructed*, and for MLS or RKPM, *nonlocality is embedded in the weight function*. Chen et al. [4] also made the same argument given earlier that *the non-local model is identical to the continuous form of the meshless kernel approximation* and thus leaving the issue to reconcile errors in the discrete kernel approximations.

While mathematical analysis of meshfree methods have been around for many years now, the analysis for nonlocal schemes to the nonlocal PD models have only got started very recently. Yet, we note that recent studies in [26] and [27] have

offered more clear elucidations of the roles of the nonlocal scaling parameter and the discretization parameter in numerical discretizations of nonlocal models and their local counterparts, so that we may gain more insight on an issue that is universal to other meshfree methods as well. To offer a mathematical understanding of the intertwining issues related to changing modeling parameter and mesh parameter, the nonlocal calculus of variations framework developed for the nonlocal models [5, 6, 16–18] becomes a useful tool. Such a framework was built upon the basic elements of the nonlocal vector calculus [5, 7] and it provides a rigorous basis to study AC schemes as discussed in [26] and [27].

Our studies so far have focused on the linear variational problems and Galerkin type approximations. We are currently exploring the extensions to time-dependent problems, nonlinear models, approximations in terms of generalized or extended finite element methods and other meshfree formulations. Further development of AC schemes for realistic nonlocal materials models will have more practical impact on the verification and validation of numerical simulations and nonlocal modeling.

Acknowledgements The authors would like to thank Max Gunzburger, Rich Lehoucq, Tadele Mengesha, Michael Parks, Stewart Silling, Florin Bobaru, John Foster, Erdogan Madenci and John Mitchell for their discussions on the subject. This work is supported in part by the U.S. NSF grant DMS-1318586, and AFOSR MURI center for material failure prediction through peridynamics.

References

1. E. Askari, F. Bobaru, R.B. Lehoucq, M.L. Parks, S.A. Silling, O. Weckner, Peridynamics for multiscale materials modeling. J. Phys.: Conf. Ser. **125**, 012078 (2008)
2. Z.P. Bazant, T. Belytschko, T.-P. Chang, Continuum theory for strain-softening. J. Eng. Mech. **110**(12), 1666–1692 (1984)
3. Z.P. Bazant, M. Jirásek, Nonlocal integral formulations of plasticity and damage: survey of progress. J. Eng. Mech. **128**(11), 1119–1149 (2002)
4. J.S. Chen, C.-T.g. Wu, T. Belytschko, Regularization of material instabilities by meshfree approximations with intrinsic length scales. Int. J. Numer. Methods Eng. **47**(7), 1303–1322 (2000)
5. Q. Du, M. Gunzburger, R.B. Lehoucq, K. Zhou, Analysis and approximation of nonlocal diffusion problems with volume constraints. SIAM Rev. **56**, 676–696 (2012)
6. Q. Du, M. Gunzburger, R.B. Lehoucq, K. Zhou, Analysis of the volume-constrained peridynamic Navier equation of linear elasticity. J. Elast. **113**, 193–217 (2013)
7. Q. Du, M. Gunzburger, R.B. Lehoucq, K. Zhou, A nonlocal vector calculus, nonlocal volume-constrained problems, and nonlocal balance laws. Math. Models Methods Appl. Sci. **23**, 493–540 (2013)
8. Q. Du, Z. Huang, R. Lehoucq, Nonlocal convection-diffusion volume-constrained problems and jump processes. Discret. Contin. Dyn. Syst. B **19**, 961–977 (2014)
9. Q. Du, L. Ju, L. Tian, K. Zhou, A posteriori error analysis of finite element method for linear nonlocal diffusion and peridynamic models. Math. Comput. **82**, 1889–1922 (2013)
10. Q. Du, L. Tian, X. Zhao, A convergent adaptive finite element algorithm for nonlocal diffusion and peridynamic models. SIAM J. Numer. Anal. **51**, 1211–1234 (2013)
11. Q. Du, K. Zhou, Mathematical analysis for the peridynamic nonlocal continuum theory. Math. Model. Numer. Anal. **45**, 217–234 (2011)

12. E. Emmrich, O. Weckner, On the well-posedness of the linear peridynamic model and its convergence towards the Navier equation of linear elasticity. Commun. Math. Sci. **5**(4), 851–864 (2007)
13. B. Kilic, E. Madenci, Coupling of peridynamic theory and the finite element method. J. Mech. Mater. Struct. **5**(5), 707–733 (2010)
14. W.K. Liu, Y. Chen, S. Jun, J.S. Chen, T. Belytschko, C. Pan, R.A. Uras, C.T. Chang, Overview and applications of the reproducing kernel particle methods. Arch. Comput. Methods Eng. **3**(1), 3–80 (1996)
15. W.K. Liu, S. Li, T. Belytschko, Moving least-square reproducing kernel methods (i) methodology and convergence. Comput. Methods Appl. Mech. Eng. **143**(1), 113–154 (1997)
16. T. Mengesha, Q. Du, Analysis of a scalar nonlocal peridynamic model with a sign changing kernel. Discret. Contin. Dyn. Syst. B **18**(5), 1415–1437 (2013)
17. T. Mengesha, Q. Du, Nonlocal constrained value problems for a linear peridynamic Navier equation. J. Elast. **116**(1), 27–51 (2014)
18. T. Mengesha, Q. Du, The bond-based peridynamic system with dirichlet type volume constraint. Proc. R. Soc. Edinb. A **144**, 161–186 (2014)
19. J. Monaghan, Smoothed particle hydrodynamics. Annu. Rev. Astron. Astrophys. **30**, 543–574 (1992)
20. J. Monaghan, Smoothed particle hydrodynamics. Rep. Prog. Phys. **68**(8), 1703 (2005)
21. S.A. Silling, Reformulation of elasticity theory for discontinuities and long-range forces. J. Mech. Phys. Solids **48**(1), 175–209 (2000)
22. S.A. Silling, E. Askari, A meshfree method based on the peridynamic model of solid mechanics. Comput. Struct. **83**(17), 1526–1535 (2005)
23. S.A. Silling, M. Epton, O. Weckner, J. Xu, E. Askari, Peridynamic states and constitutive modeling. J. Elast. **88**(2), 151–184 (2007)
24. S.A. Silling, R.B. Lehoucq, Convergence of peridynamics to classical elasticity theory. J. Elast. **93**(1), 13–37 (2008)
25. S.A. Silling, O. Weckner, E. Askari, F. Bobaru, Crack nucleation in a peridynamic solid. Int. J. Fract. **162**(1), 219–227 (2010)
26. X. Tian, Q. Du, Analysis and comparison of different approximations to nonlocal diffusion and linear peridynamic equations. SIAM J. Numer. Anal. **51**, 3458–3482 (2013)
27. X. Tian, Q. Du, Asymptotically compatible schemes and applications to robust discretization of nonlocal models. SIAM J. Numer. Anal. **52**(4), 1641–1665 (2014)
28. K. Zhou, Q. Du, Mathematical and numerical analysis of linear peridynamic models with nonlocal boundary conditions. SIAM J. Numer. Anal. **48**(5), 1759–1780 (2010)

Non-intrusive Uncertainty Quantification with Sparse Grids for Multivariate Peridynamic Simulations

Fabian Franzelin, Patrick Diehl, and Dirk Pflüger

Abstract Peridynamics is an accepted method in engineering for modeling crack propagation on a macroscopic scale. However, the sensitivity of the method to two important model parameters – elasticity and the particle density – has not yet been studied. Motivated by Silling and Askari (Comput Struct 83(17–18):1526–1535, 2005) and Kidane et al. (J Mech Phys Solids 60(5):983–1001, 2012) we use Peridynamics to simulate a high-speed projectile impacting a plate and study the overall damage on the plate. We have extended the setting by the magnitude of the force of the indenter and selected the parameter range such that a sharp transition in the response function occurs.

We describe the simulation setting as an uncertainty quantification problem and use a non-intrusive stochastic collocation method based on spatially adaptive sparse grids to propagate the uncertainty. We show first convincing results of its successful application to Peridynamics and compare to Monte Carlo sampling. If the magnitude of the force is deterministic, a strong sensitivity of the damage in the plate with respect to the elasticity factor can be shown for the 2-dimensional setting. If it is non-deterministic, it dominates the simulation and explains most of the variance of the solution. The error of the expectation value estimation reaches an early saturation point for the studied collocation methods: We found parameter ranges where the quantity of interest oscillates. Moreover, faster convergence and higher robustness than for the Monte Carlo method can be observed.

Keywords Uncertainty quantification • Sparse grids • Peridynamics • Sensitivity analysis • ANOVA

F. Franzelin (✉) • D. Pflüger
Universität Stuttgart, Universitätsstr. 38, 70569 Stuttgart, Germany
e-mail: fabian.franzelin@ipvs.uni-stuttgart.de; dirk.pflueger@ipvs.uni-stuttgart.de

P. Diehl
Universität Bonn, Wegelerstr. 6, 53115 Bonn, Germany
e-mail: diehl@ins.uni-bonn.de; schweitzer@ins.uni-bonn.de

M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations VII*, Lecture Notes in Computational Science and Engineering 100,
DOI 10.1007/978-3-319-06898-5_7

1 Introduction

Most simulations depend on certain parameters, such as material parameters or model parameters. Frequently, the exact values of parameters are not given, and the effect of uncertainty in the choice of parameters on certain output values are of interest. In plenty of applications, the influence of parameters to the simulation results are of high relevance. In this paper, we present a non-intrusive method for a d-dimensional parameter study for simulations, both to estimate the influence of parameters and to quantify uncertainties. To this end, we consider the peridynamic theory [20, 21], a non-local particle-based theory in continuum mechanics with a focus on discontinuous functions as they arise in fracture mechanics.

Silling and Askari used in their publication about Peridynamics [21] the impact of a spherical indenter on a cylindrical plate to show the method's capability to model complex fractures. Kidane et al. [9] employed a similar setting. They simulated the impact of steel projectiles with ballistics speed on aluminum plates, with regard to a data-on-demand uncertainty quantification (UQ) protocol.

Motivated by their work, this paper brings together these two aspects, Peridynamics and UQ, using stochastic collocation based on spatially adaptive sparse grids, a numerical scheme for higher-dimensional discretizations [3]. Our simulation setting is the impact of a high-speed projectile of steel on a ceramic plate. The aim of this paper is to quantify the sensitivity of the damage in the plate with regard to three relevant model parameters. First, we consider a scaling factor for the elasticity in the Prototype Brittle Material (PMB) model, which is used to model the ceramic material of the plate. The second one is the density of particles in the plate, which we model by the initial distance between two particles, and the third one is a scaling factor for the magnitude of the force of the indenter. We formulate the parameter study as a UQ problem and interpret these parameters as uncertain, i.e., as random variables.

Due to the uncertainty of the inputs, the model's outcome becomes uncertain as well, but neither statistical moments nor a probability distribution are known a priory. They depend on the uncertainty of the parameters, which have to be propagated through the simulation. We consider non-intrusive UQ, sample the stochastic space on so-called collocation nodes, compute the outcome of interest and analyze the effects. The most common method based on this principle is Monte Carlo sampling. It is independent of the dimensionality of the stochastic space, easy to implement, and it is guaranteed to converge to the true solution in the limit of infinitely many samples. Its main drawback is a rather slow convergence rate.

In a simulation context each collocation node requires a full-scale simulation and is often very costly. This motivated us to use the knowledge about the solution, namely the simulation results at the collocation nodes, and interpolate between them. A very common approach to construct such an interpolant is the non-intrusive generalized polynomial chaos expansion (gPC), presented in [25, 26]. The expansion is based on the Askey Scheme, which assigns parameter distributions to optimized orthogonal bases. The corresponding coefficients are computed by

spectral projection, which leads of course to a possibly high-dimensional quadrature problem. However, the basis functions have global support and suffer therefore Gibb's phenomenon when sharp transitions occur [25]. Therefore, [24] introduced the multi-element gPC, which decomposes the stochastic space adaptively using an Analysis-of-Variance (ANOVA) based decomposition criterion. Oladyshkin extended the polynomial chaos expansion to arbitrary parameter distributions, which is known as the arbitrary polynomial chaos method [12].

However, we propose a different approach, which is called adaptive sparse grid collocation (ASGC) method in [5]. It is based on standard spatially adaptive sparse grids [3, 11, 15, 16] and expands the stochastic space using a hierarchical basis. It can cope with the curse of dimensionality to a large extent per construction, which gPC methods can not. Moreover, the sparse grid method provides, due to its hierarchical approach, a direct estimate for the interpolation error for free, which allows one to adaptively refine to peculiarities of the underlying function. This grid-based approach scales linearly in the number of collocation nodes and then allows the efficient estimation of the statistical moments. The basis functions have local support and can thus resolve sharp transitions without suffering a lot from Gibb's phenomenon. Moreover, we consider several strategies for adaptive refinement and are able to further reduce the number of samples. Sparse grid collocation is a highly active field of research in the context of UQ: Jakeman et al. worked on the detection of discontinuities in high-dimensional stochastic spaces [8], Archibald et al. extended it to hybrid parallel architectures [1], and Zhang et al. used it to improve Bayesian inference in UQ settings [27], to name but a few.

In this work, we employ the ASGC method to simulations with Peridynamics as a first, still low-dimensional proof of concept for its efficiency and robustness in a real-world example. To keep the number of simulation runs small we propose a new refinement criterion. Having the explicit function representation of the solution in hand, we estimate global sensitivity values by variance decomposition based on [22].

In the next section we introduce the UQ problem formally, and describe how we compute global sensitivity values. In Sect. 3 we present the adaptive sparse grid collocation method in detail. In Sect. 4, we introduce the peridynamic model. The simulation setting is described in Sect. 5. We present numerical results in Sect. 6 for a two- and three-dimensional setting that can explain properties of the underlying simulations, and conclude this work with some remarks in Sect. 7.

2 Problem Setup

The task we address in this paper is to estimate sensitivity values of peridynamic parameters with respect to some quantity of interest we compute out of the results of peridynamic simulations. We describe the variation of the input by probability density functions, which we propagate through the simulation. In this section we

discuss first the forward propagation of uncertainty and second the sensitivity analysis based on ANOVA.

2.1 Forward Propagation of Uncertainty

Let the system be driven by some model $\mathcal{M} = (\boldsymbol{\theta}, p(\boldsymbol{\theta}), F(\boldsymbol{\theta}), u)$ which depends on a finite number of random parameters $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_d) \in \Gamma$ with a given joint probability density function $p(\boldsymbol{\theta})$ and cumulative distribution function $F(\boldsymbol{\theta})$, and a response function $u: \Gamma \rightarrow \mathbb{R}$ for some quantity of interest. So we seek a functional representation of $u(\boldsymbol{\theta})$ under the model $\mathcal{M}$ in order to extract its probabilistic characterization introduced by the probabilistic inputs $\boldsymbol{\theta}$. Knowledge of the probability laws of $\boldsymbol{\theta}$ leads to the probability laws of u.

However, the model can be arbitrarily complex and an analytical solution might not even exist. To overcome this problem we replace the real solution by some suitable approximation f such that

$$f(\boldsymbol{\theta}) \approx u(\boldsymbol{\theta}) \; . \tag{1}$$

Once we have an approximation f we can give estimates of the probability density function (risk analysis), statistical moments (expectation value, variance), confidence intervals of u, and even do a sensitivity analysis.

However, computing global sensitivity values based on ANOVA includes computing multi-dimensional variances of f. Motivated by Lemma 1, we can neglect the probability density function $p(\boldsymbol{\theta})$ for the computation of these variances due to a suitable transformation of the stochastic space to the unit hypercube, as long as the random variables are independent.

Lemma 1. *Let $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d) \in \Gamma$ be a multivariate random variable from which its components θ_i are arbitrarily distributed but independent. They have a continuous and differentiable cumulative function $F(\boldsymbol{\theta})$ and a continuous probability density function $p(\boldsymbol{\theta})$. Then it holds that $\boldsymbol{\xi} := F(\boldsymbol{\theta}) \in [0,1]^d$ is uniformly distributed under F and the expectation value for some function $u: \Gamma \rightarrow \mathbb{R}$ is*

$$\mathbb{E}(u) = \int_{\Omega} u(\boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta} = \int_{[0,1]^d} u(F^{-1}(\boldsymbol{\xi})) d\boldsymbol{\xi} \; . \tag{2}$$

Proof. Due to the independence of θ_i and θ_j for $i \neq j$ we can define ξ_i componentwise as

$$\xi_i = F(\theta_i) \quad \text{for } 1 \leq i \leq d \; . \tag{3}$$

Standard statistics secures that each ξ_i is uniformly distributed in $[0, 1]$. In order to proof Eq. (2) we just substitute $\boldsymbol{\theta}$ by $F^{-1}(\boldsymbol{\xi})$. According to the substitution theorem for integrals we need to transform the ranges of the integral and insert a volume correction factor to the integral, i.e.,

$$\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}\boldsymbol{\xi}} = |\det J_{F^{-1}}(\boldsymbol{\xi})| \,, \tag{4}$$

where $J_{F^{-1}}(\boldsymbol{\xi})$ is the Jacobian matrix of F^{-1}. Moreover, we know that $F^{-1}(\boldsymbol{\xi})$ is bijective, Lebesgue-measurable and differentiable, and therefore the transformation theorem for probability densities [23] holds, and thus

$$p(\boldsymbol{\xi}) = |\det J_{F^{-1}}(\boldsymbol{\xi})| p(F^{-1}(\boldsymbol{\xi})) = |\det J_{F^{-1}}(\boldsymbol{\xi})| p(\boldsymbol{\theta}) = 1 \,. \tag{5}$$

Inserting Eqs. (4) and (5) in the definition of the expectation value leads to the general calculation rule for the expectation value under F

$$\begin{aligned} \mathbb{E}(u) &= \int_{\Gamma} u(\boldsymbol{\theta}) p(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta} \\ &= \int_{[0,1]^d} u(F^{-1}(\boldsymbol{\xi})) \underbrace{\frac{|\det J_{F^{-1}}(\boldsymbol{\xi})|}{|\det J_{F^{-1}}(\boldsymbol{\xi})|}}_{=1} p(\boldsymbol{\xi}) \, \mathrm{d}\boldsymbol{\xi} \\ &= \int_{[0,1]^d} u(F^{-1}(\boldsymbol{\xi})) \mathrm{d}\boldsymbol{\xi} \,. \end{aligned} \tag{6}$$

□

For the more general case where no independence assumption of the marginal distributions can be made, we want to refer the reader to other, more elaborate statistical transformations, for example the Rosenblatt-transformation [18] or the Nataf-transformation.

However, we approximate the variance of u by the variance of f using Steiner's translation theorem $\mathbb{V}(f) = \mathbb{E}(f^2) - \mathbb{E}(f)^2$ and estimate each term with Eq. (2).

With respect to Peridynamics, we consider a model $\mathcal{M}(\boldsymbol{\theta}, p(\boldsymbol{\theta}), F(\boldsymbol{\theta}), u)$ with the uncertain parameters $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$. With θ_1 as a material parameter, θ_2 as a discretization parameter and θ_3 as the external force. The response function u maps the uncertain parameters via simulation to the damage of the material, our quantity of interest. The model is formally introduced in Sects. 4 and 5.

2.2 Global Sensitivity Analysis

For the global sensitivity analysis we use the unanchored ANOVA decomposition [22] of the surrogate f. In general, this would require $2^d - 1$ integrations.

For large d this is often unfeasible. To overcome this, one can employ the anchored ANOVA approach where the integrals are replaced by function evaluations at some anchor point [6]. As we deal with rather low dimensionalities so far, the unanchored approach is sufficiently efficient for the purposes of this paper. To be specific, we compute the ANOVA representation of f as

$$f(\theta_1,\ldots,\theta_d) = f_0 + \sum_{i=1}^{d} f_i(\theta_i) + \sum_{i=1}^{d}\sum_{i<j}^{d} f_{i,j}(\theta_i,\theta_j) + \cdots + f_{1,2,\ldots,d}(\theta_1,\ldots,\theta_d)\ . \tag{7}$$

This decomposition is unique if each member $f_{i_1,\ldots,i_k} \neq f_0$ has zero mean, i.e.,

$$\int_{\Gamma_{i_1}\times\cdots\times\Gamma_{i_k}} f_{i_1,\ldots,i_k}(\theta_{i_1,\ldots,i_k})\, p(\theta_{i_1,\ldots,i_k})\mathrm{d}\theta_{i_1,\ldots,i_k} = 0 \tag{8}$$

for some uncorrelated and independent probability measure $p(\theta_{i_1,\ldots,i_k}) = \prod_{j=1}^{k} p(\theta_{i_j})$ [6, 22]. In total we have 2^d members, which we identify in the following by $\mathcal{P}(\mathcal{D})$ being the power set over $\mathcal{D} := \{1,\ldots,d\}$. On this set of sets we introduce a total order $\mathcal{J} \prec \mathcal{K} \Leftrightarrow |\mathcal{J}| < |\mathcal{K}|$, $\mathcal{J},\mathcal{K} \in \mathcal{P}(\mathcal{D})$ and can now write Eq. (7) in a compact form (notation adapted from [7]) as

$$f(\theta_1,\ldots,\theta_d) = f_\emptyset + \sum_{k=1}^{d}\sum_{\mathcal{J}\in\mathcal{P}(\mathcal{D})} f_{\mathcal{J}}(\theta_{\mathcal{J}})\ . \tag{9}$$

where $f_\emptyset = f_0$. To guarantee orthogonality, we obtain the constant term $f_\emptyset$ from

$$f_\emptyset = \mathbb{E}(f) = \int_\Gamma f(\boldsymbol{\theta})\, p(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}\ , \tag{10}$$

and the higher-order terms from

$$\mathbb{E}(f_{\mathcal{J}}) = \int_{\Gamma_{\mathcal{D}\setminus\mathcal{J}}} f(\boldsymbol{\theta})\, p(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}_{\mathcal{D}\setminus\mathcal{J}} = f_\emptyset + \sum_{\mathcal{K}\prec\mathcal{J}} f_{\mathcal{K}} + f_{\mathcal{J}}\ , \tag{11}$$

where we integrate with respect to all $\theta_k, k \in \mathcal{D}\setminus\mathcal{J}$. Note, that as f is a function, $\mathbb{E}(f_{\mathcal{J}})$ is as well a function that depends on $\theta_{\mathcal{J}}$.

If we insert the recursive definition of the lower order terms $f_{\mathcal{K}}$ we obtain $f_{\mathcal{J}}$ as a linear combination of the $f_{\mathcal{K}}$ with $\mathcal{K} \prec \mathcal{J}$,

$$f_{\mathcal{J}} = \mathbb{E}(f_{\mathcal{J}}) + \sum_{\mathcal{K}\prec\mathcal{J}} (-1)^{|\mathcal{J}|-|\mathcal{K}|}\mathbb{E}(f_{\mathcal{K}})\ . \tag{12}$$

Having the ANOVA decomposition at hand we can compute global sensitivity values for the input parameters. We want to know the contribution of the variance of each combination of parameters to the variance of the solution. We decompose the variance of the solution by writing it as a sum of individual variances using the ANOVA representation and obtain

$$\mathbb{V}(f) = \sum_{\mathcal{J} \in \mathcal{P}(\mathcal{D})} \mathbb{V}(f_{\mathcal{J}}) \tag{13}$$

if the random variables are independent. Due to the linearity of the variance operator and Eq. (12) we can write the variance of each member as

$$\mathbb{V}(f_{\mathcal{J}}) = \mathbb{E}(f_{\mathcal{J}}^2) + \sum_{\mathcal{K} \prec \mathcal{J}} (-1)^{|\mathcal{J}|-|\mathcal{K}|} \mathbb{E}(f_{\mathcal{K}}^2) \tag{14}$$

taking into account that each member has zero mean.

The ratio

$$S_{\mathcal{J}} = \frac{\mathbb{V}(f_{\mathcal{J}})}{\mathbb{V}(f)} \tag{15}$$

is now a global sensitivity measure called the *main effect*. Certainly, all $S_{\mathcal{J}}$ are non-negative, and they sum up to 1. The main effect describes how much of the total variance can be explained by one parameter combination. For the untruncated ANOVA decomposition one has 2^d combinations covering interactions up to dth order. This might get confusing for higher-dimensional problems. Hence we want to introduce another sensitivity measure, which describes the overall contribution of one single parameter to the variance of the solution. It is called the *total effect* and is computed by summing up all the main effects where the parameter is involved, i.e.,

$$S_{\theta_k} = \sum_{\mathcal{J} \in \mathcal{D}: k \in \mathcal{J}} S_{\mathcal{J}} \,. \tag{16}$$

A total effect $S_{\theta_k} = 1$ means that the whole variance in the solution is caused by θ_k, a total effect of $S_{\theta_k} = 0$ that θ_k is irrelevant.

3 Adaptive Sparse Grid Collocation Method

Spatially adaptive sparse grids were first introduced to the setting of uncertainty quantification in [5] and have been employed to both interpolation and quadrature in various settings even before, see [3, 16] and the references therein. The main idea of ASGC is to use an adaptively refined sparse grid function as an approximation of u for the unknown functional dependencies of the random input parameters $\boldsymbol{\theta}$, as

described in Sect. 2, and extract statistical quantities using the approximation. The approximation is constructed by interpolation, which means that every sparse grid point is a collocation node in terms of UQ, and a parameter combination in terms of numerical simulations.

3.1 Sparse Grids

To briefly recall the most important properties and to clarify our notation, we describe the basic principles of sparse grids in the following; see, e.g., [3, 15–17] for further details. Sparse grids are based on a hierarchical (and thus inherently incremental and adaptive) formulation of a one-dimensional basis, which is then extended to the d-dimensional setting via a tensor product approach.

As discussed in Sect. 2, we scale the stochastic parameter space Γ to the unit-hypercube and restrict ourselves to piecewise d-linear functions $f_{\mathcal{I}} : [0,1]^d \to \mathbb{R}$. They are defined on an equidistant mesh with N grid points and mesh-width $h_{l_k} := 2^{-l_k}$ in dimension k.

We denote $\boldsymbol{l}$ and $\boldsymbol{i}$ as multi-indices with level and index for each dimension, $|\boldsymbol{l}|_1$ as the classical l^1 norm for vectors, and comparison operators on multi-indices component-wise. We can define grid points $\boldsymbol{\theta}_{\boldsymbol{l},\boldsymbol{i}} = (i_1 2^{l_1}, \ldots, i_d 2^{l_d}) \in [0,1]^d$ on a hierarchy of grids uniquely by the set $\mathcal{I}_{\boldsymbol{l}}$ of level-index tuples $(\boldsymbol{l}, \boldsymbol{i})$ with

$$\mathcal{I}_{\boldsymbol{l}} := \{(\boldsymbol{l}, \boldsymbol{i}) : 1 \leq i_k < 2^{l_k}, i_k \text{ odd}, k = 1, \ldots, d\} ,$$

omitting even-indexed ones.

To obtain the space of piecewise d-linear functions, we first define one-dimensional basis functions $\phi_{l,i}$, depending on a level l and an index i, out of the reference hat function $\phi(\theta) := \max(1 - |\theta|, 0)$ via translation and scaling as $\phi_{l,i}(\theta) := \phi(2^l \theta - i)$, see Fig. 1 (left) for the basis functions up to level 3. We then obtain d-dimensional basis functions $\phi_{\boldsymbol{l},\boldsymbol{i}}$ as a product of the respective one-dimensional ones,

$$\phi_{\boldsymbol{l},\boldsymbol{i}}(\boldsymbol{\theta}) := \prod_{k=1}^{d} \phi_{l_k,i_k}(\theta_k) ,$$

which are centered at the grid points $\boldsymbol{\theta}_{\boldsymbol{l},\boldsymbol{i}}$, see Fig. 1.

We then define hierarchical increment spaces $W_{\boldsymbol{l}}$ for which the grid points are the Cartesian product of the one-dimensional ones on the respective one-dimensional levels as $W_{\boldsymbol{l}} := \text{span}(\{\phi_{\boldsymbol{l},\boldsymbol{i}} : (\boldsymbol{l}, \boldsymbol{i}) \in \mathcal{I}_{\boldsymbol{l}}\})$ and functions $f_{\mathcal{I}_{\boldsymbol{l}}}(\boldsymbol{\theta}) \in W_{\boldsymbol{l}}$ as a sum of basis functions weighted by hierarchical coefficients $v_{\boldsymbol{l},\boldsymbol{i}}$ (so-called surpluses),

$$f_{\mathcal{I}_{\boldsymbol{l}}}(\boldsymbol{\theta}) = \sum_{(\boldsymbol{l},\boldsymbol{i}) \in \mathcal{I}_{\boldsymbol{l}}} v_{\boldsymbol{l},\boldsymbol{i}} \phi_{\boldsymbol{l},\boldsymbol{i}}(\boldsymbol{\theta}) . \tag{17}$$

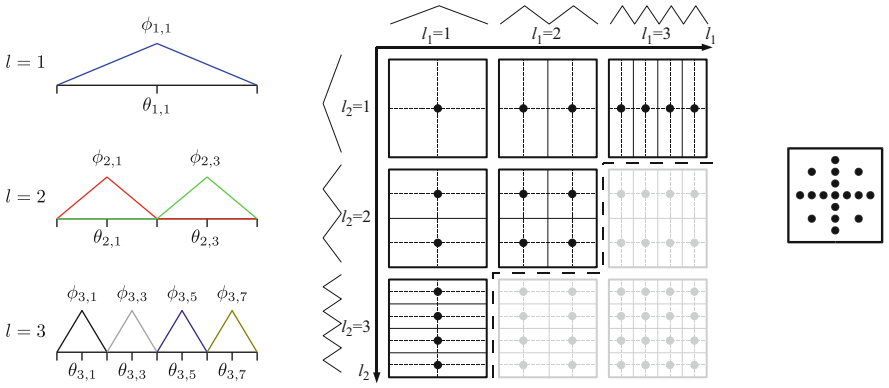


Fig. 1 One-dimensional basis functions up to level 3 (*left*), and tableau of hierarchical increments $W_{\boldsymbol{l}}$ up to level 3 in both dimensions (*center*). Leaving out the grayed-out $W_{\boldsymbol{l}}$, we obtain the sparse grid of level 3 (*right*)

Figure 1 (center) shows the grids of the two-dimensional hierarchical increments $W_{\boldsymbol{l}}$ up to level 3 in each dimension. Note that in each $W_{\boldsymbol{l}}$, all basis functions have supports with piecewise disjoint interiors.

The hierarchical representation now allows one to select only those subspaces that contribute most to the overall solution. This can be done by an a priori selection (see [3] for details). We then obtain a sparse grid space such as

$$V_n^{(1)} := \bigoplus_{|\boldsymbol{l}|_1 \leq n+d-1} W_{\boldsymbol{l}} \,,$$

which in this case is optimal with respect to both the L^2-norm and the maximum-norm for suitably smooth functions. In the example in Fig. 1, we can neglect the gray $W_{\boldsymbol{l}}$ for $n = 3$, which leads to the regular (non adaptive) sparse grid in Fig. 1 (right). This reduces the number of collocation or grid points significantly from $O((2^n)^d)$ for full grids to $O(2^n n^{d-1})$ with only slightly deteriorated accuracy if the function f under consideration is sufficiently smooth, i.e., if the mixed second derivatives $\left|D^{\mathbf{2}} f\right| := \left|\frac{\partial^{2d}}{\partial\theta_1^2 \cdots \partial\theta_d^2} f\right|$ are bounded.

In general, sparse grids are more beneficial for higher-dimensional approximation problems. But the hierarchical approach is also beneficial when every single collocation node is costly. Moreover, in their adaptive formulation, sparse grids are often able to reduce the number of grid points even more, especially when the function to be interpolated does not meet the smoothness requirements. We present spatially adaptive sparse grids in the next section.

3.2 Adaptive Refinement

We suggest to use adaptively refined grids to keep the number of collocation nodes and thus costly simulation runs as small as possible, and to be able to deal with sharp transitions in the quantity of interest. The main idea is to add new collocation nodes where the local error with respect to some metric is large. The optimal way would be to run the simulation for all new candidates and select the one which contributes most with respect to the quantity one wants to optimize. This is, of course, computationally not feasible in the context of UQ. Therefore we exploit the hierarchical structure of the method and use the hierarchical surpluses as an indicator for the local interpolation error. We thus select the most promising nodes of the current grid and create all their hierarchical descendants.

We propose three approaches to measure the contribution associated with the collocation points, from which the first is designed to minimize the interpolation error, the second refines the grid due to a local variance measure, and the third minimizes the error with respect to the expectation value. For this purpose, let $\mathcal{A} \subseteq \mathcal{I}$ be the admissible set of level index vectors $(\boldsymbol{l}, \boldsymbol{i})$ out of all collocation nodes $\mathcal{I}$, which are refinable, i.e. are leaf nodes in at least one direction. With $\mathcal{I}$ we refer to an adaptively refined set of sparse grid collocation nodes and omit therefore the subscript used for regular (non-adaptive) sparse grids in the previous section. Furthermore, note that we applied the refinement criteria presented below individually and not sequentially.

1. *Absolute surplus refinement:* Let $f_{\mathcal{I}}$ be the sparse grid function which interpolates our solution. We use the coefficients $v_{\boldsymbol{l},\boldsymbol{i}}$ as a local interpolation error measure [16] and refine the grid according to the largest one,

$$\max_{(\boldsymbol{l},\boldsymbol{i})\in\mathcal{A}} |v_{\boldsymbol{l},\boldsymbol{i}}| \,. \tag{18}$$

2. *Variance surplus refinement:* As proposed in [10], we interpret the hierarchical coefficients $w_{\boldsymbol{l},\boldsymbol{i}}$ of the interpolant of the squared solution $f_{\mathcal{I}}^2 = \sum_{(\boldsymbol{l},\boldsymbol{i})\in\mathcal{I}} w_{\boldsymbol{l},\boldsymbol{i}}\, \phi_{\boldsymbol{l},\boldsymbol{i}}$ as a measure for the local variance. By refining the area where the local variance is large, we minimize the error in the global variance of the solution. Thus, we refine the collocation node $(\boldsymbol{l}, \boldsymbol{i})$ with the largest absolute coefficient

$$\max_{(\boldsymbol{l},\boldsymbol{i})\in\mathcal{A}} |w_{\boldsymbol{l},\boldsymbol{i}}| \,. \tag{19}$$

3. *Expectation value refinement:* The previous two are greedy criteria and may get stuck in local extrema as they just consider hierarchical coefficients for refinement. Moreover, since we want to assess a quadrature problem one should consider the contribution of each refinable node to it's solution. Therefore, we propose a new refinement criterion, which refines the grid according to the largest contribution of a collocation node to the expectation value, i.e.

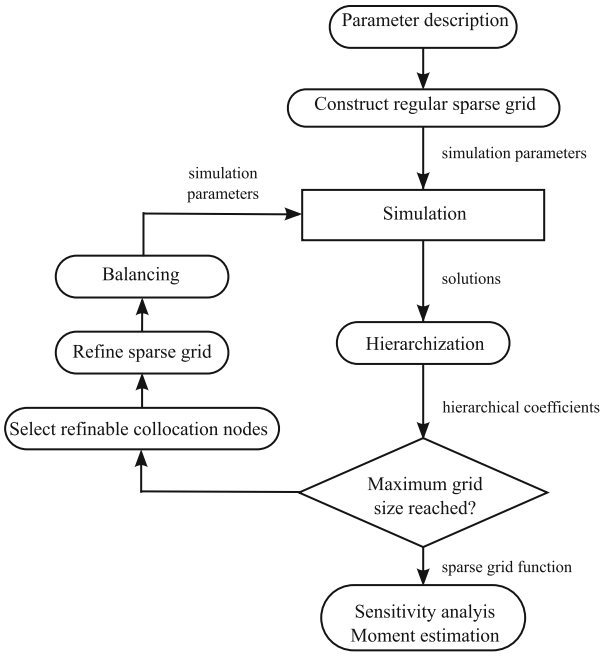


Fig. 2 The ASGC pipeline for a forward propagation problem in UQ

$$\max_{(\boldsymbol{l},\boldsymbol{i})\in\mathcal{A}} |\mathbb{E}(f_{\mathcal{I}\setminus\{(\boldsymbol{l},\boldsymbol{i})\}}) - \mathbb{E}(f_{\mathcal{I}})| = \max_{(\boldsymbol{l},\boldsymbol{i})\in\mathcal{A}} \int_{\Gamma} |f_{\mathcal{I}\setminus\{(\boldsymbol{l},\boldsymbol{i})\}}(\boldsymbol{\theta}) - f_{\mathcal{I}}(\boldsymbol{\theta})| p(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta}$$

$$= \max_{(\boldsymbol{l},\boldsymbol{i})\in\mathcal{A}} \int_{\Gamma} \left| \sum_{(\tilde{\boldsymbol{l}},\tilde{\boldsymbol{i}})\in\mathcal{I}\setminus\{(\boldsymbol{l},\boldsymbol{i})\}} v_{\tilde{\boldsymbol{l}},\tilde{\boldsymbol{i}}} \phi_{\tilde{\boldsymbol{l}},\tilde{\boldsymbol{i}}}(\boldsymbol{\theta}) - \sum_{(\tilde{\boldsymbol{l}},\tilde{\boldsymbol{i}})\in\mathcal{I}} v_{\tilde{\boldsymbol{l}},\tilde{\boldsymbol{i}}} \phi_{\tilde{\boldsymbol{l}},\tilde{\boldsymbol{i}}}(\boldsymbol{\theta}) \right| p(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta}$$

$$= \max_{(\boldsymbol{l},\boldsymbol{i})\in\mathcal{A}} |v_{\boldsymbol{l},\boldsymbol{i}}| \int_{\Gamma} \phi_{\boldsymbol{l},\boldsymbol{i}}(\boldsymbol{\theta}) p(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta} \ . \tag{20}$$

Observe that the integral $\int_{\Gamma} \phi_{\boldsymbol{l},\boldsymbol{i}}(\boldsymbol{\theta}) p(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta}$ reduces to $2^{-|\boldsymbol{l}|_1}$ if the random variables are independent and uniformly distributed on the unit hypercube. Thus, we refine the grid point $(\boldsymbol{l}, \boldsymbol{i})$ with maximum contribution

$$\max_{(\boldsymbol{l},\boldsymbol{i})\in\mathcal{A}} |\mathbb{E}(f_{\mathcal{I}\setminus\{(\boldsymbol{l},\boldsymbol{i})\}}) - \mathbb{E}(f_{\mathcal{I}})| = \max_{(\boldsymbol{l},\boldsymbol{i})\in\mathcal{A}} |v_{\boldsymbol{l},\boldsymbol{i}}| 2^{-|\boldsymbol{l}|_1} \ . \tag{21}$$

This refinement criterion includes for each collocation node the volume of the basis function's support and describes therefore a local estimate for the quadrature error.

The sensitivity analysis is a quadrature problem on a sparse grid. Therefore we want to make two remarks here.

First, the sensitivity analysis described in Sect. 2.2 can be applied directly to sparse grids. Every ANOVA term but the constant one is a sparse grid function, which is obtained by dimension-wise integration. For all one-dimensional sub-grids, one just needs to sum over all hierarchical surpluses multiplied by the volume of their respective one-dimensional basis functions. The result is the coefficient of the lower-dimensional ANOVA term.

Second, Bungartz proposes in [2] to use balanced grids for quadrature. This makes sure that every collocation node has none or both children in every dimension, which balances the error cancellation. We therefore enforce balanced grids for all the criteria by applying a balancing step after refinement.

Finally, the complete ASGC pipeline for this task is shown in Fig. 2.

4 Peridynamics

The principle of this theory is that particles in a continuum interact with surrounding particles in a finite distance by exchanging forces. Some of these concepts are similar to concepts in molecular dynamics. We refer to the notation, used in [21], to describe the Peridynamics.

Figure 3 shows the body $\mathcal{R}$ of the continuum in a reference configuration with particles $P := \{p_i | i = 1, \ldots, m\}$. Each particle p_i has an initial position x_i in the so-called reference configuration Ω and a neighborball $B_\delta(x_i)$ with the radius δ, in which the particles interact by forces. Literally speaking, the particle does not "see" particles outside its own neighborball and is not influenced by them. Within the neighborball we consider the particles being connected by virtual bonds which can break during the simulation due to the displacement of the particles, i.e., when particle p_j leaves the fixed neighborball $B_\delta(x_i)$. Computing the acceleration at position x in the reference configuration Ω at time t yields the integral equation

$$\varrho(x)\ddot{\boldsymbol{u}}(x,t) = \int_{B_\delta(x)} g(x', x, \boldsymbol{u}(\cdot, t))dx' + b(x,t) \,. \tag{22}$$

Equation (22) contains the mass density $\varrho(x)$ of the material at position x in the reference configuration Ω, $\boldsymbol{u}$ is the displacement vector field, $b(x,t)$ denotes the external force at position x at time t, and g is the so-called pairwise force function.

Figure 3 sketches the discretization in space by a collocation approach of the continuum by $P := \{p_i | i = 1, \ldots, m\}$ particles on an equidistant lattice with mesh-widths Δx, Δy, and Δz. Each particle p_i has a surrounding volume $V_i = \Delta x \cdot \Delta y \cdot \Delta z$. To discretize the equation of motion (22), the set

$$\mathcal{F}_i = \{j \mid \|x_j - x_i\| \leq \delta, \; j \neq i\} \,, \tag{23}$$

is defined for each particle p_i. The set $\mathcal{F}_i$ contains the indices of all particles, which are in the neighborball $B_\delta(x_i)$ of particle p_i. This means that the distance between

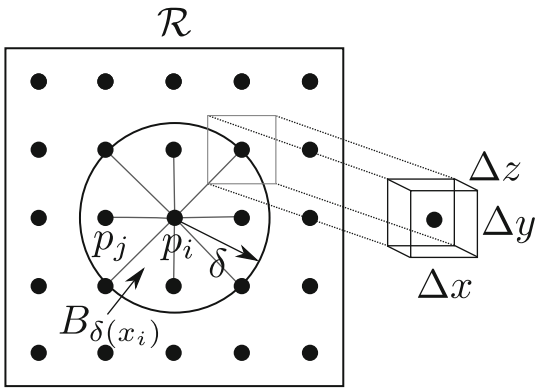


Fig. 3 The body $\mathcal{R}$ of the continuum with particles $P := \{p_i | i = 1, \ldots, m\}$ at time $t = 0$ (the reference configuration), particle p_i at position x_i with the neighborball $B_\delta(x_i)$, particle p_j at position x_j, the relative position $x_j - x_i$, and an exemplary surrounding volume $V = \Delta x \cdot \Delta y \cdot \Delta z$ on an equidistant lattice with mesh-widths Δx, Δy, and Δz

particle p_j and p_i is less or equal than δ with respect to the reference configuration Ω of the body.

We then obtain the semi-discrete Peridynamic equation,

$$\varrho(x_i)\ddot{\boldsymbol{u}}(x_i, t) = \sum_{j \in \mathcal{F}_i} g(x_j, x_i, \boldsymbol{u}(\cdot, t))\tilde{V}_j + b(x_i, t) , \tag{24}$$

with $\varrho(x_i)$ as the mass density function, g as the pairwise force function and b the external force. The volume $\tilde{V}_j$ is the scaled volume V_j of particle p_j with the linear dimensionless scaling function

$$v(x_j, x_i) = \begin{cases} -\frac{1}{\Delta x}\|x_j - x_i\| + \left(\frac{\delta}{\Delta x} + \frac{1}{2}\right), & \delta - \frac{\Delta x}{2} \leq \|x_j - x_i\| \leq \delta \\ 1, & \|x_j - x_i\| \leq \delta - \frac{\Delta x}{2} \\ 0, & \text{otherwise} . \end{cases} \tag{25}$$

The scaling function v is needed for particles p_j which are bound to particle p_i but are close to the neighborball $B_\delta(x_i)$. A part of their volume is outside the neighborball of particle p_i. The influence of such a particle differs from that of a particle, which lies completely within $B_\delta(x_i)$. If the distance $\|x_j - x_i\| = \delta$, then the volume V_j is simply scaled by 0.5, because nearly one half of the volume is inside and one half of the volume is outside $B_\delta(x_i)$.

4.1 Material Model

Equation (22) describes the interaction between the particles, but gives no explicit information about the behavior of the simulated material. The modeling of the material is hidden in the kernel function g. To simulate brittle material, like ceramic or fused silica, the prototype microelastic brittle (PMB) material model is used. In

the PMB material model the assumption is made that the pairwise force function for inner forces g depends only on the bond stretch s, which is defined by

$$s(x_j, x_i, \boldsymbol{u}(\cdot, t)) := \frac{\|\boldsymbol{u}(x_j, t) - \boldsymbol{u}(x_i, t) + x_j - x_i\| - \|x_j - x_i\|}{\|x_j - x_i\|} . \tag{26}$$

The easiest way to model failure is to let bonds break when they are stretched beyond a predefined constant value. The function s obviously attains positive values if the bond is under tension. An isotropic material has the property that it behaves identically in all directions.

Therefore the bond stretch is independent of the direction of $x_j - x_i$, and the pairwise force function g in a PMB material model is defined as

$$g(x_j, x_i, \boldsymbol{u}(\cdot, t)) = h(x_j, x_i, \boldsymbol{u}(\cdot, t)) \frac{\boldsymbol{u}(x_j, t) - \boldsymbol{u}(x_i, t) + x_j - x_i}{\|\boldsymbol{u}(x_j, t) - \boldsymbol{u}(x_i, t) + x_j - x_i\|} , \tag{27}$$

where h is a linear scalar-valued function which implements the behavior of the material and the decision if the bond is broken or "alive". It is defined as

$$h(x_j, x_i, \boldsymbol{u}(\cdot, t)) = \begin{cases} c \cdot s(x_j, x_i, \boldsymbol{u}(\cdot, t)) \cdot \mu(x_j, x_i, \boldsymbol{u}(\cdot, t)), & \|x_j - x_i\| \leq \delta \\ 0, & \|x_j - x_i\| > \delta , \end{cases} \tag{28}$$

with c being the material dependent stiffness constant of the PMB material model and s the bond stretch (26). The function μ is an history dependent, scalar-valued function that models the inability of the material to "heal" broken bonds. It is defined as

$$\mu(x_j, x_i, \boldsymbol{u}(\cdot, t)) = \begin{cases} 1 & s(x_j, x_i, \boldsymbol{u}(\cdot, t)) < s_{00} - \alpha s_{\min}(t'), \ \forall\, 0 \leq t' \leq t, \\ 0 & \text{otherwise} , \end{cases} \tag{29}$$

where s_{00} is the critical stretch for bond failure in the material and $s_{\min}(t')$ is the minimal bond stretch up to t', i.e. $s_{\min}(t') = \min_{x_j - x_i} s(x_j, x_i, \boldsymbol{u}(\cdot, t'))$. Hence the stiffness constant c and the critical stretch for bond failure s_{00} are material parameters in the PMB material model. However, the factor α is usually not considered as an material parameter, but fixed at 0.25 [21].

Nevertheless, α influences the elasticity of the material: increasing α leads to a higher critical stretch for the bonds and keeps them alive longer. This, consequently, affects the local damage, which is defined at position x at time t as

$$\varphi(x, \boldsymbol{u}(\cdot, t)) = 1 - \frac{1}{V_{B_\delta(x)}} \int_{B_\delta(x)} \mu(x', x, u(\cdot, t)) \mathrm{d}x' . \tag{30}$$

This yields to the total damage, our quantity of interest

$$c_t = \frac{1}{m} \sum_{i=1}^{m} \varphi(x_i, \boldsymbol{u}(\cdot, t)) \,, \tag{31}$$

where $\varphi(x_i, \boldsymbol{u}(\cdot, t))$ is the relative number of bond failures for particle i at time t.

To study the influence of the uncertain parameters $\alpha =: \theta_1$, $\Delta x =: \theta_2$ and $K =: \theta_3$, the force the indenter exerts to the plate, to the total damage $c_t =: u$, our response function, we use a specific model $\mathcal{M}_{\text{Peridynamics}} = (\boldsymbol{\theta}, p(\boldsymbol{\theta}), F(\boldsymbol{\theta}), u)$.

5 Simulation

For the simulation we use a common peridynamic example which is described in [13, 14]. Figure 4 shows the CAD model of this experiment. The target is a homogeneous plate of fused silica with a diameter of 0.074 m and a height of 0.0025 m. The Peridynamics material parameters for fused silica are $s_{00} = 6.99 \cdot 10^{-6}$ and $c = 2.19 \cdot 10^{22}$. The indenter is a sphere with diameter 0.01 m and impacts the cylinder with a directed velocity of $v = 100$ m/s perpendicular to the surface of the target, see Fig. 5. The properties of the uncertain parameters we used for the simulations are shown in Table 1.

The particles are arranged in an equidistant grid with the same spacing Δx in all three dimension, a first parameter. Increasing Δx leads to a reduction of the particle density in the plate. A standard value for Δx in practice is 0.5 mm, which leads to 876,435 particles in the simulation and a total of 102,865,590 bonds. Decreasing Δx a lot gets very fast computationally unfeasible. Here, we consider $\Delta x \in [0.4, 0.6]$. Note, Δx defines as well the radius $\delta := 3\Delta x$ of the neighborball $B_{\delta(x_i)}$ of each

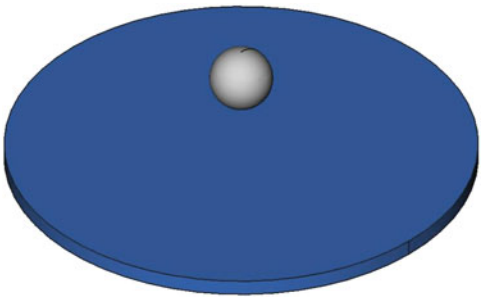


Fig. 4 CAD model of the simulation configuration

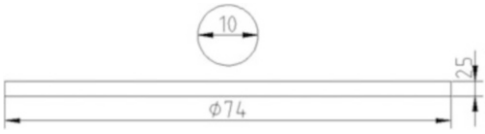


Fig. 5 Setting of the experiment

Table 1 Range of the parameters that define the stochastic input space for the quantification of uncertainties

Parameter	Min	Max	Unit	Distribution	Description
Δx	0.4	0.6	mm	$\mathcal{U}(0.4, 0.6)$	Grid spacing
α	0	1	–	$\mathcal{U}(0, 1)$	Model parameter for bond stretch failure
K	10^{12}	10^{20}	N/m^2	$\mathcal{U}(12, 20)$	Magnitude of force on indenter

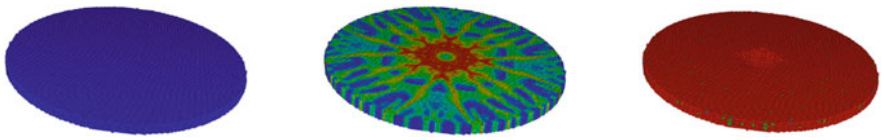


Fig. 6 Simulation runs for $K = 10^{12}$ (*left*), $K = 10^{16}$ (*center*), and $K = 10^{20}$ (*right*) at the same time step $t = 300$, $\alpha = 0.25$, and $\Delta x = 0.0005$

particle. This means that by varying Δx the properties of the material of the plate vary as well. However, this interplay has yet not been quantified in peridynamic theory and is part of ongoing research.

The influence of the second parameter α has not yet been studied in detail. It is by definition restricted to the unit interval, but literature suggests to use $\alpha = 0.25$ as this leads to somehow realistic fracturing behavior. We want to especially study its influence on the peridynamic method in detail and set therefore the parameter range to $[0, 1]$.

For the external force K the *fix indent* command is used with a sphere that has a constant diameter of 0.01 m. The magnitude of the force K, which is exerted by the indenter, variates from 10^{12} up to 10^{20} on a logarithmic scale. The parameter range of it is chosen such that there is no damage in the plate for low K, and that the plate gets destroyed almost completely for large K, see Fig. 6.

Altogether, these three parameters define the stochastic space $\Gamma := [0.4, 0.6] \times [0, 1] \times [12, 20]$ for the UQ-setting. They are independent and uniformly distributed by definition, which reduces the transformation F according to Sect. 2 to be a linear scaling. As the peridynamic simulation framework we use LAMMPS [13], and SG^{++} [15] for sparse grids and UQ.

6 Numerical Results

We have studied our approach for both an analytical example and the peridynamic simulation setting. The analytical example is close to the peridynamic simulation with just the first two parameters, but does not exhibit the "real-world" issues as we will observe for the peridynamic simulations. It validates the ASGC method and shows its faster convergence with respect to the absolute error of the expectation

value compared to vanilla Monte Carlo and quasi-Monte Carlo. For comparisons to polynomial chaos methods we refer to [10].

In the following numerical examples we use the mean squared error with respect to M uniformly drawn Monte Carlo samples as an error indicator for the interpolation quality of an interpolant $f_{\mathcal{I}}$ with $|\mathcal{I}| = N$ collocation nodes, i.e.

$$\epsilon_I := \frac{1}{M} \sum_{i=1}^{M} (f_{\mathcal{I}}(\boldsymbol{\theta}_i) - u(\boldsymbol{\theta}_i))^2 \ . \tag{32}$$

The error in the expectation value is computed as

$$\epsilon_E := |\mathbb{E}(f_{\mathcal{I}}) - \mathbb{E}_r(u)| \ , \tag{33}$$

where the reference value $\mathbb{E}_r(u)$ is either the analytical solution of the quadrature problem or obtained by the sample mean over all Monte Carlo observations, i.e.

$$\mathbb{E}_r(u) = \frac{1}{M} \sum_{i=1}^{M} u(\boldsymbol{\theta}_i) \ . \tag{34}$$

The respective reference value for the variance $\mathbb{V}_r(u)$ is computed accordingly.

We compared the convergence of the ASGC method with three other non-intrusive forward propagation methods: Monte Carlo with uniformly drawn samples, quasi-Monte Carlo with Sobol-sequences, and a full-grid interpolant with hierarchical basis where the collocation nodes are equidistantly distributed over the complete stochastic space. With respect to Monte Carlo, we generated a training set, from which we generated 20 Monte Carlo paths by permutation. We estimated different expectation values for each path by adding samples iteratively to it. By averaging over all paths we reduce the variance of Monte Carlo and get one sufficiently smooth Monte Carlo path, which we used for comparison.

6.1 Analytical Example

We consider the function

$$u(\theta_1, \theta_2) = \arctan(50(\theta_1 - 0.35)) + \frac{\pi}{2} + 4\theta_2^3 + e^{\theta_1 \theta_2 - 1} \tag{35}$$

with two stochastic parameters $\theta_1, \theta_2 \ \sim \ \mathcal{U}(0, 1)$. A graphical representation is given in Fig. 7. The function u is designed to represent the characteristics of the peridynamic simulation just after the indenter has hit the plate. We can think of the parameters θ_1 and θ_2 as representatives for $1 - \alpha$ and Δx. Their functional dependencies are motivated as follows:

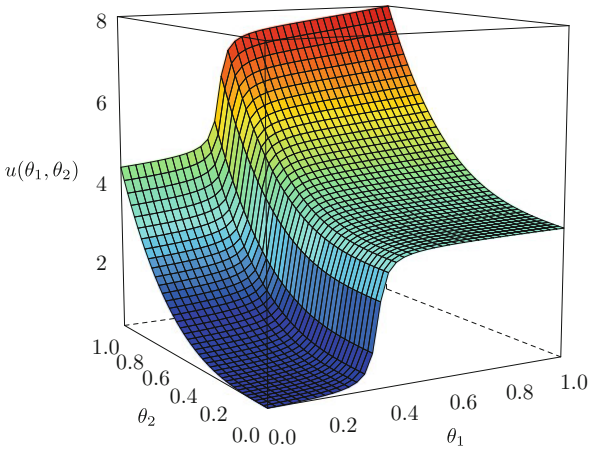


Fig. 7 3-dimensional plot of $u(\theta_1, \theta_2)$

1. For high α the damage will be low, as the modeled material is rather hard compared to the softer material for lower α. In-between we expect a sharp transition at $\theta_1 = 0.35$, which we model with a shifted and steep arctan function depending on θ_1.
2. The term depending just on θ_2 is motivated by the fact that a particle-based numerical simulation should converge to the true solution when one increases the number of particles in the plate. Increasing the number of particles means decreasing the initial distance Δx between them. We want to reach convergence with $\theta_2 \to 0$. We expect to overestimate the damage for too few particles and use therefore a cubic dependence on θ_2.
3. The last term in u models an unknown dependency between the two parameters.

We first construct 4 interpolants of u: the regular sparse grid interpolant, and the adaptively refined sparse grid interpolants using the three refinement strategies of Sect. 3.2. Considering the refinement strategies, we just refined collocation nodes with a hierarchical surplus larger than $\epsilon := 10^{-10}$, per iteration we refined $\min\{10, N\frac{5}{100}\}$ of the collocation nodes with the largest contribution and stopped if either there were no more points to be refined or the upper limit of 3,000 collocation nodes was reached. To estimate their interpolation quality we used 10^6 Monte Carlo test samples. The error in the expectation value is computed as the absolute difference to the up to 14 digits exact reference value $\mathbb{E}_r(u) = 3.51449126644638$. For the Monte Carlo method we computed 2^{14} training samples, and the same number of quasi-Monte Carlo samples.

With respect to the mean squared interpolation error ϵ_I, as shown in Fig. 8, we can observe that the full grid method converges as expected with $\mathcal{O}(N^{-2})$. The regular sparse grid method shows almost the same error as the full grid method on the same discretization level. And as it requires less grid points per level, it converges therefore faster with an upper bound of $\mathcal{O}(N^{-5}\log(N)^6)$ [3]. The adaptive methods converge with the same speed as the regular sparse grid

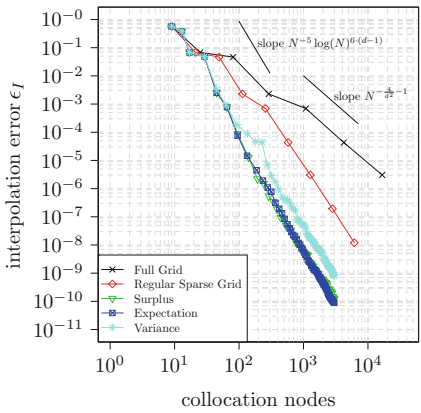


Fig. 8 Interpolation error measured by the mean squared error at 10^6 Monte Carlo test samples. The *black lines* in the right upper corner show the slopes for the regular sparse grid and the full grid [3]

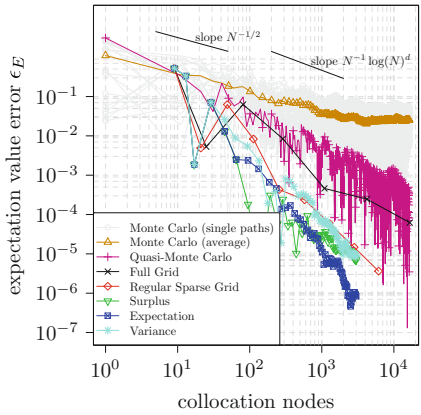


Fig. 9 Absolute error of the estimated expectation value for different methods. The *black lines* in the upper part of the figure show the slopes for Monte Carlo and quasi-Monte Carlo [4]

method. However, the absolute interpolation error is three orders of magnitude lower compared to the regular sparse grid after the fourth refinement iteration with 20 collocation nodes. The surplus and the expectation value refinement approaches show similar behavior while the variance-based approach is one order of magnitude worse.

For the absolute error in the expectation value estimation, Fig. 9, we observe two important things: First, even though the interpolation error is monotonically decreasing, the error in the expectation value is not. Between the regular sparse grid of level 2 (25 collocation nodes) and level 3 (81) the error grows and cannot be fully recovered by the grid of level 4. The reason is cancellation of the integral around the steep transition in direction of θ_1. A level 2 grid is too coarse to represent the transition well, but due to cancellation, the error in the expectation value is low. This cancellation effect is destroyed on the next level because of a better approximation of the steep transition on just one side, which is once more compensated on the next level. This effect is known [2, 16] and a major problem for adaptively refined grids. To overcome it to some extent we introduced a balancing step after refinement. But

as we can see from the convergence plots it does not solve the problem completely. However, the expectation value refinement strategy reaches an accuracy of at least one order of magnitude higher than the other grid-based methods. The Monte Carlo methods show a significantly slower convergence. For 20 collocation nodes the accuracy of the grid-based methods is already one and for 2,000 collocation nodes four order of magnitudes higher. The quasi-Monte Carlo method converges as expected with $\mathcal{O}(N^{-1}\log(N)^d)$ [4], which is still significantly slower than the sparse grid methods.

With respect to the global sensitivity values we computed main effects of $S_{\theta_1} = 0.577934314364$, $S_{\theta_2} = 0.421292846812$, and $S_{\theta_1,\theta_2} = 0.0007728388246$, which sum up to 1. Hence, most of the variance of u is explained by the arctan and the cubic term. The interactive term has almost no influence on the variance.

Starting with these promising results we assess the peridynamic example in the following section.

6.2 Peridynamic Example

For this example we use the simulation described in Sect. 5. We split the analysis in two parts. First we assess the 2-dimensional example where we vary α and Δx. After that we add the magnitude of force K to the setting and discuss the resulting 3-dimensional setting.

Both of the simulations are, of course, time dependent. Hence, we have for each time step a different sparse grid interpolant. For the 2-dimensional setting we restrict ourselves to the study of the uncertainty at time step $t = 300$ of the simulation, at which the variance is largest, see Fig. 10. The Monte Carlo and the quasi-Monte Carlo training sets for comparison contain 2,500 samples each.

We determine the reference expectation value $\mathbb{E}_r(u)$ measured with $M = 6{,}000$ Monte Carlo test samples as the exact value is unknown. The Monte Carlo samples

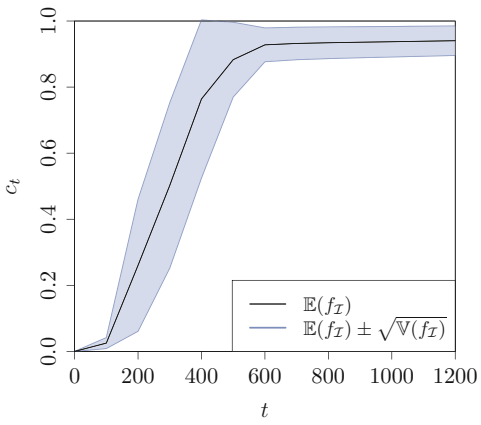


Fig. 10 The expectation value and the standard deviation over time, estimated using the ASGC method

have been drawn uniformly according to the distribution of the uncertainty in the parameters. However, the quality of this estimation clearly affects the convergence with respect to the expectation value of the compared methods. The evolution of the estimated reference expectation value $\mathbb{E}_r(u)$ of the Monte Carlo test set is shown in Fig. 11. We assume that the variance of u exists and apply the central limit theorem to estimate an upper bound for the error of $\mathbb{E}_r(u)$ [4] for the most common confidence levels $\beta \in \{0.1, 0.05, 0.01\}$ given the empirical variance $\mathbb{V}_r(u) = 0.06304$, i.e.,

$$|\mathbb{E}_r(u) - \mathbb{E}(u)| \leq \Phi_{1-\frac{\beta}{2}} \sqrt{\frac{\mathbb{V}_r(u)}{M}} \tag{36}$$

The resulting upper bounds are shown in Table 2. In these circumstances we can not expect that the accuracy of expectation value estimates of the competing methods to be reliable beyond $5.33 \cdot 10^{-3}$.

With this in mind, we focus now on the convergence of the expectation value at time step $t = 300$, shown in Fig. 12, and investigated the absolute error in the expectation value, see Eq. (33).

The error convergence of the expectation value estimation for most of the sparse grid methods is, in contrast to the analytical example, similar to that of the Monte Carlo results, while showing a lower error. The regular sparse grid approach performs even a bit better than the adaptively refined ones for level 2, which is the same behavior as we have seen for the analytical example. The results for the

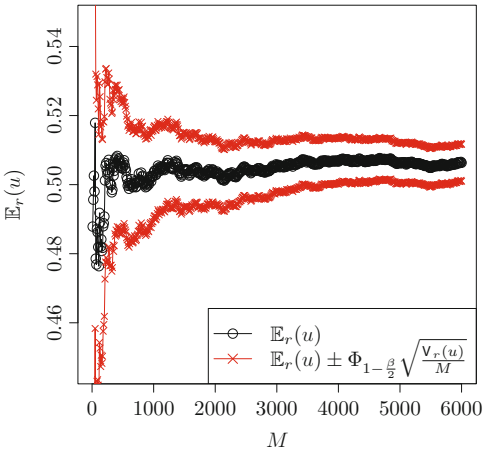


Fig. 11 The Monte Carlo simulation confidence interval obtained with the central limit theorem with a confidence level of $\beta = 0.1$

Table 2 Upper bound for the error of $\mathbb{E}_r(u)$ for different confidence levels, $M = 6{,}000$ test samples, which have an empirical variance of $\mathbb{V}_r(u) = 0.06304$

Confidence level	$\beta = 0.1$	$\beta = 0.05$	$\beta = 0.01$
Upper error bound	$5.33 \cdot 10^{-3}$	$6.35 \cdot 10^{-3}$	$8.35 \cdot 10^{-3}$

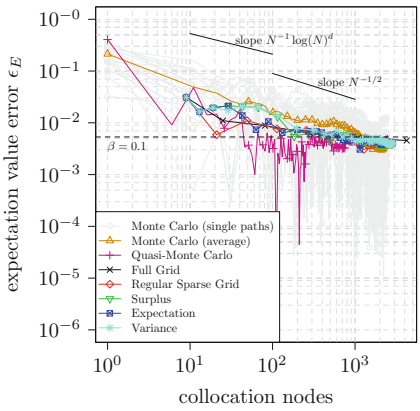


Fig. 12 Absolute error of the estimated expectation value for different methods. The *black dashed line marks* the upper bound for the error of the reference value at a confidence level of $\beta = 0.1$

adaptive sparse grids show that the criterion for adaptive refinement is very critical up to 200 collocation nodes. While the strategy based on the optimization of the expectation value performs best, its greedy counterparts, the absolute surplus and the variance surplus, cannot keep up and get stuck in local features.

However, the sparse grid methods, independent of the refinement criteria, reach a saturation point at about 110 collocation nodes. Runs with quasi-Monte Carlo samples using Sobol sequences show the same behavior, though reaching the trusted accuracy earlier at about 70 collocation nodes but oscillating more around the saturation value until 200 nodes. The Monte Carlo method reaches it with more than 1,000 nodes, too. In contrast to the regular sparse grid and the ASGC methods, the Monte Carlo methods keep oscillating in a range of two order of magnitudes around the saturation point even for large numbers of collocation nodes without reducing the error. Furthermore, the hierarchical representation of sparse grids provides the means to easily detect the early saturation here, which is much more costly for the Monte Carlo approaches due to the highly oscillating behavior.

The early saturation point is also visible in the convergence plot of the interpolation error, see Fig. 13. We can observe a saturation for the error convergence, now for around 300 collocation points, which is much too early to be credited to numerical issues. Admittedly, we did not expect this behavior beforehand. The quantity of interest we study is the mean over the damage of each particle, which we expected to be smooth. Furthermore, the interpolation error should not increase spending more samples as it is the case for all grid-based methods. These observations suggest noise or at least high frequency oscillations in the quantity of interest. This would also explain why the Monte Carlo approaches stagnate with large oscillations, and why the error is smallest for grids, which cover the parameter range more regularly.

To support this suggestion we studied the hierarchical surpluses of the regular sparse grid and the full grid, see Fig. 14. For sufficiently smooth functions the hierarchical surpluses converge to zero with increasing level. This behavior can be exploited to find non-smooth dependencies in individual dimensions [2]. In the peridynamic simulation the interquartile range of the hierarchical surpluses

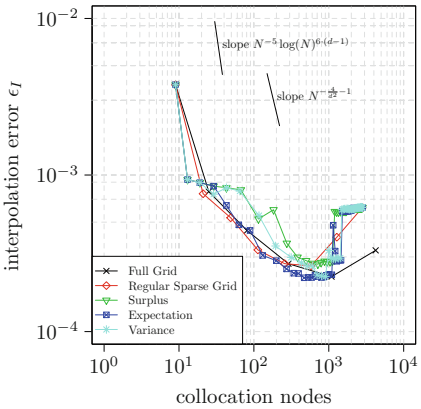


Fig. 13 Interpolation error measured at $t = 300$ by the mean squared difference at 6,000 Monte Carlo test samples (uniformly distributed). The error has a local minimum between 300 and 1,000 collocation nodes

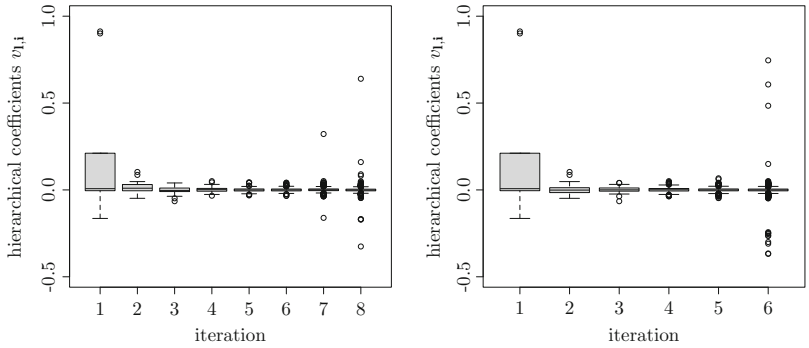


Fig. 14 Box plot of the hierarchical surpluses which are added when incrementing the level for the regular sparse grid (*left*) and the full-grid (*right*). The *gray boxes mark* the interquartile range, the *black line* within the boxes is the median, the so called "whiskers" mark the range where surpluses are located, which do not differ more than 1.5 times the interquartile range from the median. The surpluses, which lie outside of these ranges, are marked as *circles*. Note the exploding values of the hierarchical surpluses for increment iteration 7–8 for the regular sparse grid method, and on level 1 and 6 for the full grid method

decreases as expected up to level 6 for regular sparse grids and up to level 5 for the full grid. However, the hierarchical surpluses explode for higher levels and the interpolation error increases. These collocation nodes with the largest hierarchical surpluses lie mostly on the border of the domain and have a high level in at least one dimension. We investigated the collocation node with the highest hierarchical surplus 0.6397 added in the 8th iteration of the regular sparse grid. It is located at $\alpha = 1$ and $\Delta x = 0.76953125$. Its left and right neighbor in direction of Δx are spatially very close, i.e. 2^{-8}, due to the full grid resolution of the sparse grid on

the border. The quantity of interest, the average damage on the plate, for the left neighbor is approximately 0.2, for the right one is 0.21, while for the node itself is roughly 0.85. This explains the large hierarchical surplus and supports the statement of, at least, local high-frequency oscillations or, in other words, instabilities for extreme parameter combinations. The expectation value estimation is not influenced by these oscillations as the support of these high-level basis functions is rather small.

To investigate this behavior further, we show the adaptively refined sparse grid with 207 collocation nodes using the expectation value refinement strategy and the corresponding sparse grid function in Figs. 15 and 16, respectively. In Fig. 15 we can observe that most of the collocation nodes are spent in the regime $\alpha \in [0.25, 0.75]$. In between there is a transitional phase where the total damage decreases from 0.9 to 0.2. This is also the range in which most oscillations or noise occur, see the test samples with the highest errors in Fig. 17. The two steep slopes of the interpolant at about $\Delta x = 0.1$ and 0.5 have been successfully identified and resolved by the refinement strategy.

We furthermore investigated the interpolant and the error for three non-adaptive, regular sparse grids of level 1, 3 and 5, which are shown in Fig. 18. In the top left, the piecewise linear interpolant for a sparse grid of level 1 which consists of 9 collocation nodes is shown. As the real function varies much less in Δx compared to α, it can be much better approximated by a linear function in the Δx direction.

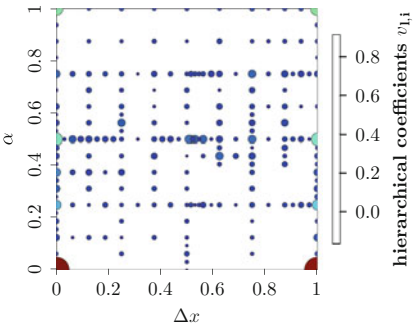


Fig. 15 An adaptively refined sparse grid with expectation value optimization refinement strategy. Color and radius of the collocation nodes represent the hierarchical coefficients. The corner nodes have the highest contribution to the function while the coefficients of the inner nodes are oscillating around zero

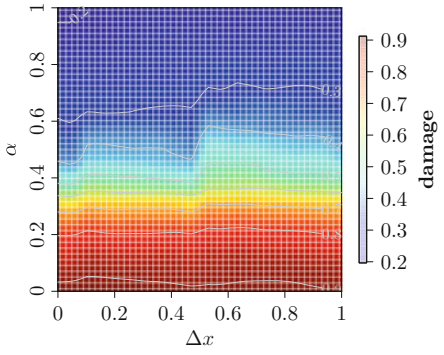


Fig. 16 Sparse grid function for the collocation nodes in Fig. 15. Note the steep slopes at $\Delta x \approx 0.1$ and 0.5. These two regions are well resolved by the adaptivity criterion and are therefore not artifacts of the sparse grid function but the behavior of the true solution

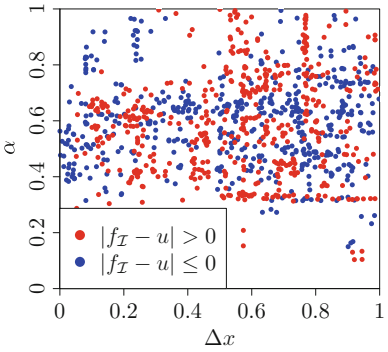


Fig. 17 The plot represents the largest local errors of the sparse grid interpolant represented in Fig. 16, using the 993 test samples, which explain 53 % of the interpolation error ϵ_I. Plenty of samples which are close to each other have errors (colors) oscillating around zero, which indicates highly frequent oscillations or noise in that regime

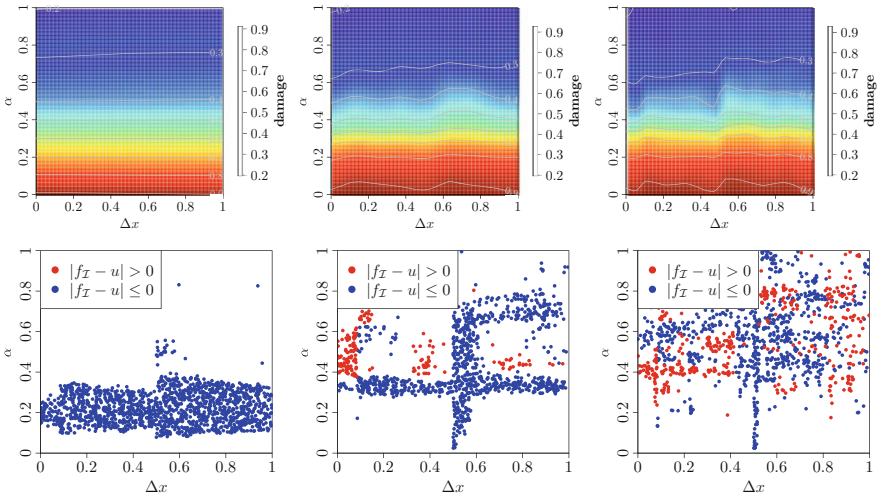


Fig. 18 Sparse grid functions for sparse grids with boundary points for levels 1, 3 and 5 (*top*), and the Monte Carlo test samples which cause at least 50 % of the interpolation error (978, 743 and 784, *bottom*)

This is directly reflected by the region in which most of the error occurs (bottom left).

The transition in the α-direction is far from linear in the lower third of the parameter space. With a level 3 grid, however, the transition can now be better represented (center column). There remain mainly two vertical clusters, in the middle and on the left, and a horizontal region of large error. These are exactly the bends that can be seen for a higher sampling resolution on level 5 in the upper

right plot. The remaining region of large error is spread out throughout vast parts of the domain.

With respect to the peridynamic theory and the sensitivities we want to study, several areas with different influence of Δx for fixed α can be roughly classified in Fig. 16. First, there are two regimes, $\Delta x \approx 0.5$ and $\Delta x \approx 0.1$, where the total damage changes rapidly with varying Δx. Increasing the particle density (reducing Δx) starting from $\Delta x = 1$, we observe similar behavior of the damage until about $\Delta x = 0.5$. One would expect that increasing the density leads to a convergence of the damage behavior (propagation of the displacement waves and cracks, amount of damage). This can be observed in simulations for $\Delta x < 0.5$, but fails for very high particle densities ($\Delta x \approx 0.1$), which is also reflected in simulations where the damage propagation changes significantly for very high particle densities. The latter effect is yet to be investigated further.

Considering α, there are two regions with less sensitivity on Δx. For $0.1 \leq \alpha \leq 0.35$ and $\alpha \geq 0.85$, the sensitivity on the particle density is very small, and the overall damage in the simulation during the penetration of the indenter varies very little. Silling and Askari proposed in [21] to choose $\alpha = 0.25$, for which the sensitivity on Δx is relatively small. In other regions of α, the influence of the particle density is significant, leading to variations of up to more than 15 % of the total damage. The typical value $\alpha = 0.25$ delivers a relatively stable behavior of the damage with respect to Δx. Considering peridynamic simulations, this analysis enables us to derive some heuristics about appropriate ranges of the particle density for chosen values of α.

The standard deviation of the solution was already shown in Fig. 10. It reaches it's maximum in the penetration phase of the projectile. In this phase ($0 < t < 400$) 96–99 % of the total variance is explained by the variance of α, see the left part of Fig. 19. For $t \geq 400$ the second-order interactions between Δx and α become more important and explain up to 10 % of the variance. The main effect of Δx has its maximum of 5 % at time step 500. However, the sensitivity values do not show the oscillating behavior of the solution. This is not surprising, as they are comparatively small in the global context.

In the three-dimensional setting we observe similar behavior with respect to the convergence. Concerning the sensitivity analysis, the dominant parameter is the force parameter K as its total effect is $S_K = 0.995580573617$. The total effects $S_\alpha = 0.0135270393259$ and $S_{\Delta x} = 0.0021731614627$ are rather small in comparison. This is not surprising considering the choice of the parameters and their ranges. The force on the indenter must have the highest influence. However, the main effect of α has an impact of 12.9 % at time step 100 when the indenter hits the plate. After the impact the sensitivity with respect to α reduces almost linearly until it vanishes after time step 800. Therewith the main effect of α directly depicts the behavior of the indenter itself. We want to recall that this values have been computed without additional samples and without introducing additional numerical errors. This is a great advantage over Monte Carlo since to obtain sensitivity values one would need $M(d + 2)$ number of samples [19]. Every sample is costly in

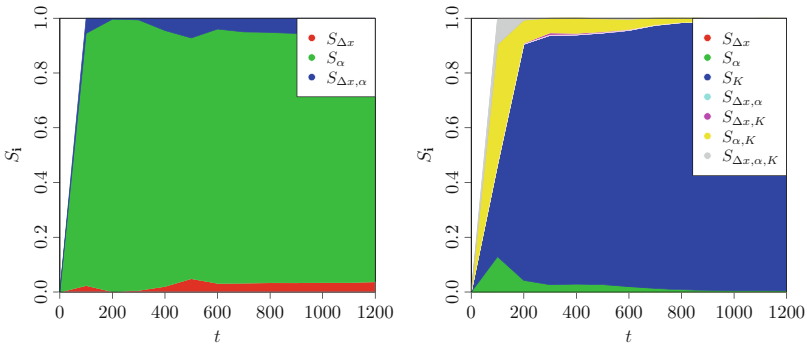


Fig. 19 Main effects and higher-order interactions of α, Δ and K for the 2-dimensional (*left*) and the three-dimensional (*right*) peridynamic UQ-setting

Peridynamics and therefore it is unfeasible to do Monte Carlo based sensitivity analysis.

7 Conclusions

In this paper we have shown the first application of adaptive sparse grids for uncertainty quantification and sensitivity analysis in a peridynamic simulation setting. We have simulated the impact of a high-speed projectile on a ceramic plate using the PMB model of Peridynamics with two and three uncertain model parameters. The first two parameters were α, which describes the elasticity of the material, and Δx, which models the particle density in the plate. As a third parameter, we have considered the force K exerted by the projectile.

With regard to the adaptive sparse grid collocation method, we have introduced a new refinement criterion that is motivated by the UQ setting, and we have shown that it can be very effective compared to traditional and more greedy strategies. For the peridynamic setting, adaptivity did not pay off as much in terms of the number of collocation nodes as we expected, as it is known from other scenarios, and as the analytical example with similar behavior suggested. This was due to unexpected noise in the quantity of interest.

For both the 2- and the 3-dimensional simulation setting, the sparse grid methods outperformed vanilla Monte Carlo in terms of moment estimation by up to one order of magnitude for small numbers of collocation nodes. For the analytical example, this holds for quasi-Monte Carlo based on Sobol-sequences, too. For the peridynamic setting the accuracy was similar than that of quasi-Monte Carlo while oscillating much less. This is promising for UQ settings where the number of costly simulation runs has to be kept low.

In the simulation setting, we have observed an early saturation of the error convergence for the expectation value, which is due to noise or highly frequent oscillations in the transition phase between high and low damage on the plate with respect to changes in the particle density. Our approach allows us to identify parameter regions that show high sensitivity to the choice of parameters, which means unstable behavior from a simulation point of view. Furthermore, we have successfully computed reliable global sensitivity values for the uncertain parameters using the unanchored ANOVA decomposition method on sparse grids. We did this without increasing the number of samples and without introducing additional numerical errors through binning.

In contrast to stochastic approaches, sparse grids have several advantages. The hierarchical surpluses can be directly used to detect outliers or instabilities in the simulation, to detect an early saturation, and for adaptive refinement. Having constructed the sparse grid interpolant, it can be used to compute moments and sensitivity values without introducing further errors.

Considering adaptivity, the effects of the different refinement strategies have to be studied in more detail, especially for higher-dimensional settings. To overcome the problem of non-smooth dependencies in the UQ context as observed for peridynamic simulations, we propose to use a regression-based reconstruction of the response function instead of interpolation which has shown promising results in first tests. Another important point of future work will be to extend the current scenario to a higher-dimensional setting. We aim to add the bulk modulus and the critical stress intensity factor to study material properties of solids. Especially from an engineering point of view such a setting would be of major interest because the studied materials do not need to exist. Moreover, connecting the material properties with the initial mesh width would allow to study their relationship in detail.

References

1. R. Archibald, R. Deiterding, J. Jakeman, Extending adaptive sparse grids for stochastic collocation to hybrid parallel architectures, Oak Ridge National Lab, No. 4, 2–6 Jan (2012)
2. H.-J. Bungartz, S. Dirnstorfer, Multivariate quadrature on adaptive sparse grids. Computing **71**(1), 89–114 (2003). (English)
3. H.-J. Bungartz, M. Griebel, Sparse grids. Acta Numer. **13**, 1–123 (2004)
4. R.E. Caflisch, Monte Carlo and quasi-Monte Carlo methods. Acta Numer. **7**, 1–49 (1998)
5. B. Ganapathysubramanian, N. Zabaras, Sparse grid collocation schemes for stochastic natural convection problems. Comput. Phys. **225**(1), 652–685 (2007)
6. M. Griebel, Sparse grids and related approximation schemes for higher dimensional problems, in Proceedings of the conference on Foundations of Computational Mathematics (FoCM05), Santander, Spain
7. M. Griebel, M. Holtz, Dimension-wise integration of high-dimensional functions with applications to finance. J. Complex. **26**(5), 455–489 (2010)
8. J. Jakeman, R. Archibald, D. Xiu, Characterization of discontinuities in high-dimensional stochastic problems on adaptive sparse grids. Comput. Phys. **230**(10), 3977–3997 (2011)

9. A. Kidane, A. Lashgari, B. Li, M. McKerns, M. Ortiz, H. Owhadi, G. Ravichandran, M. Stalzer, T.J. Sullivan, Rigorous model-based uncertainty quantification with application to terminal ballistics, part I: systems with controllable inputs and small scatter. J. Mech. Phys. Solids **60**(5), 983–1001 (2012)
10. X. Ma, N. Zabaras, An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations. Comput. Phys. **228**(8), 3084–3113 (2009)
11. F. Nobile, R. Tempone, C.G. Webster, A sparse grid stochastic collocation method for partial differential equations with random input data. SIAM J. Numer. Anal. **46**(5), 2309–2345 (2008)
12. S. Oladyshkin, H. Class, R. Helmig, W. Nowak, A concept for data-driven uncertainty quantification and its application to carbon dioxide storage in geological formations. Adv. Water Resour. **34**(11), 1508–1518 (2011)
13. M.L. Parks, R.B. Lehoucq, S.J. Plimpton, S. Silling, Implementing peridyanmics within a molecular dynamics code. Comput. Phys. Commun. **179**, 777–783 (2008)
14. M.L. Parks, P. Seleson, S.J. Plimpton, S. Silling, R.B. Lehoucq, Peridynamics with LAMMPS: a user guide v0.3 beta. Sandia report, Sandia National Laboratories, Nov 2011
15. D. Pflüger, *Spatially Adaptive Sparse Grids for High-Dimensional Problems* (Verlag Dr. Hut, München, 2010)
16. D. Pflüger, Spatially adaptive refinement, in *Sparse Grids and Applications*. LNCSE (Springer, 2012), pp. 243–262
17. D. Pflüger, B. Peherstorfer, H.-J. Bungartz, Spatially adaptive sparse grids for high-dimensional data-driven problems. J. Complex. **26**(5), 508–522 (2010)
18. M. Rosenblatt, Remarks on a multivariate transformation. Ann. Math. Stat. **23**(1952), 470–472 (1952)
19. A. Saltelli, P. Annoni, I. Azzini, F. Campolongo, M. Ratto, S. Tarantola, Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. Comput. Phys. Commun. **181**(2), 259–270 (2010)
20. S. Silling, Reformulation of elasticity theory for discontinuties and long-range forces. Sandia report SAND98-2176, Sandia National Laboratories, 1998
21. S. Silling, E. Askari, A meshfree method based on the peridynamic model of solid mechanics. Comput. Struct. **83**(17–18), 1526–1535 (2005)
22. I.M. Sobol, Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. Math. Comput. Simul. **55**(1–3), 271–280 (2001)
23. D. Stirzaker, *Elementary Probability* (Cambridge University Press, Cambridge/New York, 2003)
24. X. Wan, E. Karniadakis, Multi-element generalized polynomial chaos for arbitrary probability measures. SIAM J. Sci. Comput. **3**, 901–928 (2006)
25. D. Xiu, E. Karniadakis, The Wiener–Askey polynomial chaos for stochastic differential equations. SIAM J. Sci. Comput. **24**(2), 619–644 (2002)
26. D. Xiu, E. Karniadakis, Modeling uncertainty in flow simulations via generalized polynomial chaos. Comput. Phys. **187**(1), 137–167 (2003)
27. G. Zhang, D. Lu, M. Ye, M. Gunzburger, C.G. Webster, An adaptive sparse-grid high-order stochastic collocation method for Bayesian inference in groundwater reactive transport modeling. Adv. Water Resour. **49**(10), 6871–6892 (2013)

Regularization and Multi-level Tools in the Method of Fundamental Solutions

Csaba Gáspár

Abstract The Method of Fundamental Solution is applied to potential problems. The source and collocation points are supposed to coincide and are located along the boundary. The singularities due to the singularity of the fundamental solution are avoided by several techniques (regularization and desingularization). Both the monopole and the dipole formulations are investigated. The resulting algebraic systems have advantageous properties provided that pure Dirichlet or pure Neumann boundary condition is prescribed. Otherwise, the original problem is converted to a sequence of pure Dirichlet and pure Neumann subproblems, the solutions of which converge rapidly to the solution of the original mixed problem. The iteration is embedded to a multi-level context in a natural way. Thus, the computational cost can be significantly reduced, and the problem of large and ill-conditioned matrices is also avoided.

Keywords Method of fundamental solutions • Meshfree method • Regularization • Desingularization • Multi-level method

1 Introduction

The Method of Fundamental Solutions (MFS) has become quite popular due to its simplicity and the fact that it is a truly meshfree method which requires neither domain nor boundary mesh (or grid) structure. To solve inhomogeneous problems, the MFS is often coupled with the well-known Method of Particular Solutions. If L is a second-order partial differential operator has the form

$$Lu = f \quad \text{in } \Omega, \qquad u|_{\Gamma_D} = u_D, \quad \frac{\partial u}{\partial n}|_{\Gamma_N} = v_N, \tag{1}$$

C. Gáspár (✉)
Széchenyi István University, Egyetem tér 1, H-9026 Győr, Hungary
e-mail: gasparcs@sze.hu

M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations VII*, Lecture Notes in Computational Science and Engineering 100,
DOI 10.1007/978-3-319-06898-5_8

(where Ω is the domain of the partial differential equation, $\Gamma_D \cup \Gamma_N = \partial\Omega$ is a disjoint decomposition of its boundary to a Dirichlet part Γ_D and a Neumann part Γ_N), then the solution u is expressed as a sum of a particular solution u_P and a homogeneous solution u_H:

$$u = u_P + u_H$$

Here u_P solves the inhomogeneous equation $Lu_P = f$ (no boundary condition is prescribed), while u_H is the solution of the following homogeneous problem supplied with modified boundary conditions:

$$Lu_H = 0 \quad \text{in } \Omega, \qquad u_H|_{\Gamma_D} = u_D - u_P|_{\Gamma_D}, \quad \frac{\partial u_H}{\partial n}|_{\Gamma_N} = v_N - \frac{\partial u_P}{\partial n}|_{\Gamma_N}. \tag{2}$$

To create a particular and a homogeneous solution, completely different tools can be used. It is usual to apply a radial basis function (RBF) technique to get a particular solution (see e.g. [3]), while the homogeneous solution can be obtained by using the boundary knot method (BKM, see [4,5]) or the Method of Fundamental Solutions (MFS, see e.g. [1,3]). The first approach is based on nonsingular general solutions, while the second one utilizes the fundamental solution of the applied partial differential operator L. In this paper we are interested in the MFS, so that we assume that the original problem (1) is a homogeneous equation supplied with mixed boundary condition.

In its original form, the (approximate) homogeneous solution is expressed in an RBF-like form:

$$u(x) = \sum_{j=1}^{N} \alpha_j \Phi(x - \tilde{x}_j), \tag{3}$$

where Φ is a fundamental solution of the applied partial differential operator L, and $\tilde{x}_1, \ldots, \tilde{x}_N$ are so-called *source points* located outside of the domain Ω. Therefore u has no singularities inside the domain. To determine the a priori unknown coefficients $\alpha_1, \ldots, \alpha_N$, the boundary conditions are enforced in some *boundary collocation points* $x_1, \ldots, x_N \in \partial\Omega$:

$$\begin{aligned} \sum_{j=1}^{N} \alpha_j \Phi(x_k - \tilde{x}_j) &= u_D(x_k) \quad \text{(for all } k : x_k \in \Gamma_D\text{)} \\ \sum_{j=1}^{N} \alpha_j \frac{\partial \Phi}{\partial n_k}(x_k - \tilde{x}_j) &= v_N(x_k) \quad \text{(for all } k : x_k \in \Gamma_N\text{)} \end{aligned} \tag{4}$$

which is a linear system of equations for the coefficients $\alpha_1, \ldots, \alpha_N$. Here n_k denotes the outward normal unit vector at the boundary collocation point x_k.

The method can be directly applied if the fundamental solution Φ is expressed in an explicit (and more or less simple) form, e.g. in the case of the 2D and 3D Laplace, Helmholtz or modified Helmholtz operators etc. For 3D axisymmetric potential problems, see [12].

The form (3) can be regarded as a discrete form of an indirect boundary element method, where the solution u is sought in the form of a single layer potential:

$$u(x) = \int_{\tilde{\Gamma}} \alpha(y)\Phi(x-y)\, d\Gamma_y,$$

where $\tilde{\Gamma} = \partial\tilde{\Omega}$, $\tilde{\Omega}$ is a larger domain containing the original domain Ω. An alternative approach is the indirect boundary element method based on the double layer potential:

$$u(x) = \int_{\tilde{\Gamma}} \alpha(y)\frac{\partial\Phi}{\partial n_y}(x-y)\, d\Gamma_y,$$

where n_y denotes the outward normal unit vector at the boundary point y. A discrete approximation of this integral is:

$$u(x) = \sum_{j=1}^{N} \alpha_j \frac{\partial\Phi}{\partial n_j}(x-\tilde{x}_j), \tag{5}$$

where n_j is the outward normal unit vector at the source point $\tilde{x}_j$. The coefficients α_j can be computed by enforcing the boundary conditions at the boundary collocation points:

$$\begin{aligned} &\sum_{j=1}^{N} \alpha_j \frac{\partial\Phi}{\partial n_j}(x_k-\tilde{x}_j) = u_D(x_k) \quad \text{(for all } k : x_k \in \Gamma_D\text{)} \\ &\sum_{j=1}^{N} \alpha_j \frac{\partial^2\Phi}{\partial n_k \partial n_j}(x_k-\tilde{x}_j) = v_N(x_k) \quad \text{(for all } k : x_k \in \Gamma_N\text{)} \end{aligned} \tag{6}$$

Both (3) and (5) (referred to as monopole and dipole formulation, respectively) are suitable to generate approximate solutions. It turns out that (3) is more advantageous to handle pure Neumann problems, while (5) is the proper choice when pure Dirichlet problems are considered.

From computational point of view, it is well known that the matrices of the systems (4) and (6) are fully populated, nonsymmetric and highly ill-conditioned. The farther the source points are located from the boundary, the higher the condition numbers, and the systems (4) and (6) quickly become severely ill-conditioned. Note that the number of source and boundary collocation points need not be identical;

if they differ, some least squares technique or a generalized solution based on the Singular Value Decomposition (SVD) can be applied. However, the problem of the ill-conditioned matrices still remains the case.

If the boundary collocation points are located in the vicinity of the boundary, the condition numbers becomes much more moderate, but numerical singularities appear in the approximate solution. As an extreme case, when the source points and the boundary collocation points coincide, the diagonal entries of the matrices in (4) and (6) make no sense due to the singularity of the fundamental solution (as well as its normal derivative) at the origin. To avoid the problem of singularities, special techniques have to be applied to define the above diagonal terms in a proper way. These techniques (called regularization and desingularization) are outlined in the next section.

2 Regularization and Desingularization in the MFS

Here we briefly recall the fundamental ideas of the regularization and desingularization. For details, see e.g. [9].

From now on we suppose that the source and collocation points coincide. For simplicity, we restrict ourselves to the 2D Laplace equation supplied with mixed boundary condition:

$$\Delta u = 0 \quad \text{in } \Omega, \qquad u|_{\Gamma_D} = u_D, \quad \frac{\partial u}{\partial n}|_{\Gamma_N} = v_N \tag{7}$$

The fundamental solution is $\Phi(r) = \frac{1}{2\pi} \log r$ (in polar coordinates).

A general idea of regularization is to replace Φ with another function which is close to it in some sense, but has no singularity at the origin. The simplest way is to use a truncated fundamental solution:

$$\Phi(r) := \begin{cases} \frac{1}{2\pi} \log cr & \text{if } cr > 1 \\ 0 & \text{if } cr \le 1 \end{cases} \tag{8}$$

Here $c > 0$ is a carefully defined scaling constant; it should be inversely proportional to the characteristic distance of the boundary collocation points. Thus, Φ is continuous everywhere and satisfies the Laplace equation outside of the circle centered at the origin with radius $1/c$.

Another technique is to replace Φ with the fundamental solution of the fourth-order partial differential operator $\Delta\left(I - \frac{1}{c^2}\Delta\right)$:

$$\Phi(r) := \frac{1}{2\pi}(K_0(cr) + \log cr),$$

where K_0 is the modified Bessel function of the third kind (see [8]). Again, c should be inversely proportional to the characteristic distance of the boundary collocation points. Since the function K_0 rapidly decreases far from the origin, the function Φ approximates the harmonic fundamental solution (apart from a small neighbourhood of the origin), but has no singularity at the origin (the singularities of the two terms cancel out). For 3D and/or more general partial differential operators, the same ideas can be applied provided that the corresponding fundamental solution can be explicitly calculated.

Denoting by Φ one of the above regularized fundamental solutions, the approximate solution of (7) can be expressed in a 'monopole formulation':

$$u(x) = \sum_{j=1}^{N} \alpha_j \Phi(x - x_j), \tag{9}$$

and the coefficients $\alpha_1, \ldots, \alpha_N$ can be computed by solving the algebraic system:

$$\begin{aligned} &\sum_{j=1}^{N} \alpha_j A_{kj} = u_D(x_k) \quad \text{(for all } k : x_k \in \Gamma_D\text{)} \\ &\sum_{j=1}^{N} \alpha_j B_{kj} = v_N(x_k) \quad \text{(for all } k : x_k \in \Gamma_N\text{)} \end{aligned} \tag{10}$$

where $A_{kj} := \Phi(x_k - x_j)$, $B_{kj} := \frac{\partial \Phi}{\partial n_k}(x_k - x_j)$. The diagonal entries B_{kk} must be defined in another way due to the (stronger) singularity of $\frac{\partial \Phi}{\partial n}$ at the origin (see e.g. [18]). To this end, consider the auxiliary Dirichlet problem

$$\Delta w = 0, \qquad w|_\Gamma = 1 \tag{11}$$

(the exact solution of which is obviously $w \equiv 1$). Expressing w in the same form:

$$w(x) = \sum_{j=1}^{N} \beta_j \Phi(x - x_j),$$

and exploiting the fact that $\frac{\partial w}{\partial n}|_\Gamma \equiv 0$, the diagonal entries B_{kk} can be defined as

$$B_{kk} := -\frac{1}{\beta_k} \sum_{j \neq k} \beta_j B_{kj},$$

(provided that $\beta_k \neq 0$). This is the *desingularization* idea. See also [14, 18]. Note that, instead of the function $w \equiv 1$, other harmonic functions can also be used (see

e.g. [6]). By a proper choice of the particular solution w (using possibly several ones), it can be ensured that the definition of the diagonal entries B_{kk} makes sense, i.e. $\beta_k \neq 0$.

The dipole formulation can be constructed similarly. Let us still denote by Φ a regularized fundamental solution, then the approximate solution of (7) can be expressed also in a 'dipole form':

$$u(x) = \sum_{j=1}^{N} \alpha_j \frac{\partial \Phi}{\partial n_j}(x - x_j), \tag{12}$$

and the coefficients α_j can be computed by solving the algebraic system:

$$\begin{aligned} &\sum_{j=1}^{N} \alpha_j C_{kj} = u_D(x_k) \quad \text{(for all } k : x_k \in \Gamma_D\text{)} \\ &\sum_{j=1}^{N} \alpha_j Q_{kj} = v_N(x_k) \quad \text{(for all } k : x_k \in \Gamma_N\text{)} \end{aligned} \tag{13}$$

where $C_{kj} := \frac{\partial \Phi}{\partial n_j}(x_k - x_j)$ and $Q_{kj} := \frac{\partial^2 \Phi}{\partial n_k \partial n_j}(x_k - x_j)$, if $j \neq k$. Again, the diagonal entries must be calculated in another way. Since the dipole formulation and the normal derivative of the monopole formulation can be considered a discretized forms of the double layer potential, and the normal derivative of a single layer potential, respectively, it turns out (see [10]) that the proper definition is $C_{kk} := B_{kk}$ (while $C_{kj} = -B_{jk}$ for $j \neq k$). After defining the matrix C, the diagonal entries Q_{kk} can be defined by the same desingularization procedure as earlier, applying a dipole formulation to the auxiliary problem (11).

Numerical features. Both the monopole and the dipole formulations (9) and (12) are suitable to handle mixed boundary value problems. The condition numbers remain moderate (compared with the traditional version of the MFS). As an illustrative example, consider the Laplace equation in the square $[-1,\ 1] \times [-1,\ 1]$ with the very simple test solution

$$u(x, y) = x^2 - y^2 \tag{14}$$

(where we have applied the more familiar notations x, y for the space coordinates). Along the half of the boundary, Dirichlet boundary condition, while along the remaining part of the boundary, Neumann boundary condition is prescribed. The boundary is discretized by N boundary points located along the boundary in an equidistant way. Table 1 shows the relative L_2-errors of the approximate solutions computed on the boundary and also the condition numbers with different numbers of boundary collocation points. The scaling constant in (8) was set to $c := N$ (i.e.

Table 1 Relative L_2-errors in the domain and condition numbers. Test solution: (14). Mixed boundary conditions

N	8	16	32	64	128	256	512
Monopoles:							
Relative L_2-error (%)	17.85	7.876	3.772	1.898	0.9262	0.4859	0.2442
Condition number	8.0	21.5	53.3	127.1	294.8	670.9	1.5E+3
Dipoles:							
Relative L_2-error (%)	14.59	4.874	1.457	0.5639	0.2593	0.1272	0.0634
Condition number	3.0	6.4	13.7	29.4	63.1	135.8	287.3

it is inversely proportional to the characteristic distance of the boundary collocation points). It can be seen that the condition numbers remain moderate while the accuracy is still acceptable. No essential difference can be observed between the monopole and dipole formulation.

To solve the discrete equations (10) and (13), a simple direct technique e.g. Gaussian elimination seems to be suitable. Iterative solution techniques such as the conjugate gradient method or other Krylov subspace methods are not necessarily efficient. We note, however, that this does not remain the case if pure Dirichlet or pure Neumann problem is to be solved. To illustrate the situation, suppose that Ω is the unit circle and the boundary is discretized by the points $x_j := (\cos \frac{2\pi j}{N}, \sin \frac{2\pi j}{N})$ $(j = 0, 1, \ldots, N-1)$. Then, as it can easily be checked:

$$B_{kj} = \frac{\partial \Phi}{\partial n_k}(x_k - x_j) = \frac{1}{2\pi} \frac{\langle x_k - x_j, x_k \rangle}{||x_k - x_j||^2} = \frac{1}{4\pi}$$

for $j \neq k$. The desingularization procedure defines the diagonal entries of B to be the same constant, i.e. B has the form:

$$B = cI + \frac{1}{4\pi} w_1 w_1^*,$$

where $w_1 := (1, 1, \ldots, 1)^* \in \mathbf{R}^N$. Denote by

$$w := \frac{w_1}{||w_1||} = \frac{1}{\sqrt{N}} \cdot (1, 1, \ldots, 1)^*,$$

then $P := w \cdot w^*$ is a projector and:

$$B = cI + \frac{N}{4\pi} P$$

Due to the desingularization procedure, the diagonal entries can be calculated explicitly, yielding:

$$B = -\frac{N}{4\pi}(I - P)$$

Table 2 Relative L_2-errors in the domain and condition numbers. Test solution: (14). Pure Dirichlet and pure Neumann boundary conditions

N	8	16	32	64	128	256	512
Dipoles:							
Relative L_2-error (%)	13.50	4.716	1.611	0.6393	0.2864	0.1367	0.0671
Condition number	3.3	3.6	3.9	4.0	4.2	4.4	4.5
Monopoles:							
Relative L_2-error (%)	18.95	8.180	4.027	2.070	1.057	0.5346	0.2685
Condition number	15.8	17.4	18.4	19.2	21.1	30.8	44.5

Moreover, since $C_{kk} = B_{kk}$ and $C_{kj} = -B_{jk}$ for $j \neq k$, the matrix C has a similar form:

$$C = -\frac{N-2}{4\pi}\left(I + \frac{N}{N-2}P\right),$$

which implies that the condition numbers are bounded from above:

$$\mathrm{cond}(C) = 1 + \frac{N}{N-2} \leq 3$$

independently of N (provided that $N \geq 4$).

The underlying idea is as follows. Since the dipole formulation is a discrete form of a double layer potential, in the case of pure Dirichlet problem, (13) is the discrete for of a Fredholm integral equation of the second kind due to the well-known theorem concerning the jump of the double layer potential at the boundary. Similarly, in the case of pure Neumann problem, the monopole formulation results in the system (10), which is again the discrete form of a Fredholm integral equation of the second kind (due to the jump of the normal derivative of the single layer potential at the boundary). The operators of the Fredholm equations can be considered *compact perturbations of the identity*. It is known that in this case the simple conjugate gradient method is particularly efficient: the convergence is in fact superlinear [2]. Thus, it may be expected that this property is preserved in the discrete formulations.

As an example for the above phenomenon, consider again the harmonic test solution (14) defined on the square $[-1,\ 1] \times [-1,\ 1]$. The boundary was discretized by N boundary collocation points in an equidistant manner. Pure Dirichlet as well as pure Neumann boundary conditions were prescribed. In the case of the pure Dirichlet (resp. Neumann) boundary condition, a dipole (resp. monopole) formulation was applied. Note that, in the monopole formulation, the matrix is always singular due to the desingularization procedure; here the first equation in the system (10) was replaced by $\sum_{j=1}^{N} \alpha_j = 0$, which defines the additive constant appearing in the solution of the Neumann problem. Table 2 shows the relative L_2-errors of the approximate solutions on the boundary as well as the corresponding

condition numbers. It can clearly be seen that the accuracy was about the same as previously (cf. Table 1), however, the condition numbers are significantly reduced. Therefore it is worth applying the conjugate gradient method (or a kind of Krylov subspace methods), which makes the solution procedure more economical from computational point of view. This observation is utilized in the next section, where a mixed problem is converted to a sequence of 'pure' problems resulting in an efficient iterative solution technique.

3 Converting Mixed Problems to a Sequence of Pure Problems

First, we briefly recall the main ideas of the technique proposed in [7]. Later, the method will be adopted in the MFS context.

Consider again the 2D model problem (7). Denote by W the following closed subspace in the Sobelev space $H^{1/2}(\Gamma)$:

$$W := \{w \in H^{1/2}(\Gamma) : w|_{\Gamma_D} \equiv 0\},$$

and let P be a (not necessarily orthogonal) projector of the subspace W. Define $P_1 := I - P$ and $P_2 := P^*$. Then P_1 and P_2 are also projectors in the spaces $H^{1/2}(\Gamma)$ and $H^{-1/2}(\Gamma)$, respectively, and can be interpreted as *extension operators* from Γ_D to Γ and from Γ_N to Γ, respectively. Now define the following iteration. Starting from an approximate boundary solution U_0 of (7), let

$$\Delta U_{n+1/2} = 0, \quad U_{n+1/2}|_\Gamma = u_n + P_1(u_D - u_n), \tag{15}$$

$$\Delta U_{n+1} = 0, \quad \frac{\partial U_{n+1}}{\partial n}|_\Gamma = v_{n+1/2} + P_2(v_N - v_{n+1/2}), \tag{16}$$

where $u_n := U_n|_\Gamma$, $v_{n+1/2} := \frac{\partial U_{n+1/2}}{\partial n}|_\Gamma$. Equation (15) is a pure Dirichlet subproblem (the Dirichlet boundary condition along Γ_D is exactly fulfilled), while (16) is a pure Neumann subproblem (the Neumann boundary condition along Γ_N is exactly satisfied).

Remark. The iteration (15)–(16) can be considered a special multiplicative Schwarz method [11, 13]. However, in contrast to the usual techniques, there is no domain decomposition idea in the background.

Let us introduce the Dirichlet-to-Neumann operator (also referred to as Poincaré-Steklov operator, see [15]) J by $JU := \frac{\partial U}{\partial n}$, where $\Delta U = 0$ in Ω and $U|_\Gamma = u$. Then J is an one-to-one mapping from $H^{1/2}(\Gamma)$ onto a one-codimensional subspace of $H^{-1/2}(\Gamma)$, and the above iteration can be expressed as:

$$v_{n+1/2} := J(u_n + P_1(u_D - u_n)), \quad u_{n+1} := J^{-1}(v_{n+1/2} + P_2(v_N - v_{n+1/2})),$$

that is:

$$u_{n+1} := J^{-1}(I - P_2)J(I - P_1)u_n + J^{-1}(P_2 v_N + (I - P_2)JP_1 u_D) \tag{17}$$

If the transition operator

$$A := J^{-1}(I - P_2)J(I - P_1) \tag{18}$$

is a contraction in $H^{1/2}(\Gamma)$ (or, at least, its spectral radius is less than one), the iteration (15)–(16) is convergent.

Remark. In practice, the realization of the operators J and J^{-1} can be performed by solving a pure Dirichlet and a pure Neumann problem, respectively. This can approximately be done by the regularized MFS utilizing the dipole formulation (12) and the monopole formulation (9), respectively.

Based on the strengthened Cauchy inequality and utilizing the fact that for an arbitrary projector P, the equality $||P|| = ||I - P||$ is valid (see e.g. [17]), the following theorem can be deduced:

Theorem 1. *The norm of the operator A defined by (18) can be estimated by:*

$$||A|| \leq \sqrt{1 - \frac{1}{||P||^2}} \cdot ||P|| \cdot ||I - P|| = \sqrt{||P||^2 - 1} \cdot ||P||$$

For the spectral radius of A, the following estimation holds:

$$\rho(A) \leq ||P||^2 - 1$$

For the proof, see [7].
This means that the convergence of the iteration (17) depends on the choice of the projector P, as expected. The iteration may still converge for non-orthogonal projectors as well, provided that the norm of P is not 'too far' from the value 1.

Remark. Let us define the extension operators P_1, P_2 as follows. Let $P_1 u := w|_\Gamma$, where w is the solution of the mixed problem

$$\Delta w = 0 \quad \text{in } \Omega, \qquad w|_{\Gamma_D} = u|_{\Gamma_D}, \quad \frac{\partial w}{\partial n}|_{\Gamma_N} = 0 \tag{19}$$

Similarly, let $P_2 u := \frac{\partial w}{\partial n}|_\Gamma$, where w is the solution of the mixed problem

$$\Delta w = 0 \quad \text{in } \Omega, \qquad w|_{\Gamma_D} = 0, \quad \frac{\partial w}{\partial n}|_{\Gamma_N} = v|_{\Gamma_N} \tag{20}$$

Then obviously, the operator $P := I - P_1$ is a projector of the subspace W. Moreover:

Lemma 1. *The extension P_2 is the adjoint of P.*

Proof. For arbitrary $V \in H^{-1/2}(\Gamma)$, by definition:

$$P_2 V = \frac{\partial U}{\partial n}|_{\Gamma},$$

where

$$\Delta U = 0 \ \text{ in } \Omega, \quad U|_{\Gamma_D} = 0, \ \ \frac{\partial U}{\partial n}|_{\Gamma_N} = V|_{\Gamma_N}$$

Thus, for arbitrary $u \in H^{1/2}(\Gamma)$:

$$(P_2 V)(Pu) = \int_{\Gamma} \frac{\partial U}{\partial n} \cdot (Pu)\, d\Gamma = \int_{\Gamma_D} \frac{\partial U}{\partial n} \cdot (Pu)\, d\Gamma + \int_{\Gamma_N} \frac{\partial U}{\partial n} \cdot (Pu)\, d\Gamma =$$
$$= \int_{\Gamma_N} V \cdot (Pu)\, d\Gamma,$$

since $(Pu)|_{\Gamma_D} = 0$ and $\frac{\partial U}{\partial n}|_{\Gamma_N} = V|_{\Gamma_N}$. Utilizing the equality $(Pu)|_{\Gamma_D} = 0$ once more, we have:

$$(P_2 V)(Pu) = \int_{\Gamma_N} V \cdot (Pu)\, d\Gamma = \int_{\Gamma_N} V \cdot (Pu)\, d\Gamma + \int_{\Gamma_D} V \cdot (Pu)\, d\Gamma =$$
$$= \int_{\Gamma} V \cdot (Pu)\, d\Gamma = V(Pu) = (P^* V)u$$

That is: $(P_2 V)(Pu) = (P^* V)u$. But $Pu = u - P_1 u$, whence

$$(P^* V)u = (P_2 V)u - (P_2 V)(P_1 u)$$

It is sufficient to prove that $(P_2 V)(P_1 u) = 0$, since this implies that $(P^* V)u = (P_2 V)u$ for arbitrary $u \in H^{1/2}(\Gamma)$, $V \in H^{-1/2}(\Gamma)$, that is, $P^* = P_2$. By definition:

$$P_1 u = W|_{\Gamma},$$

where

$$\Delta W = 0 \ \text{ in } \Omega, \quad W|_{\Gamma_D} = u|_{\Gamma_D}, \ \ \frac{\partial W}{\partial n}|_{\Gamma_N} = 0$$

Consequently:

$$(P_2 V)(P_1 u) = \int_{\Gamma} \frac{\partial U}{\partial n} \cdot W\, d\Gamma = \int_{\Gamma} U \cdot \frac{\partial W}{\partial n}\, d\Gamma,$$

where we used Green's second identity. Moreover:

$$(P_2V)(P_1u) = \int_{\Gamma_D} U \cdot \frac{\partial W}{\partial n}\, d\Gamma + \int_{\Gamma_N} U \cdot \frac{\partial W}{\partial n}\, d\Gamma = 0,$$

since $U|_{\Gamma_D} = 0$ and $\frac{\partial W}{\partial n}|_{\Gamma_N} = 0$. This completes the proof.

Lemma 2. *In the case of the above defined extensions P_1, P_2, the transition operator $A = J^{-1}(I - P_2)J(I - P_1)$ is the zero operator.*

Proof. It is sufficient to prove that $(I - P_2)J(I - P_1) = 0$, that is,

$$(I - P_2)JPu = 0$$

for arbitrary $u \in H^{1/2}(\Gamma)$. P is a projector of the subspace W, therefore $(Pu)|_{\Gamma_D} = 0$. We will show that for every $w \in W$, the equality $Jw = P_2Jw$ holds, which proves the lemma. Indeed, $Jw = \frac{\partial W}{\partial n}|_{\Gamma}$, where

$$\Delta W = 0, \qquad W|_{\Gamma} = w,$$

i.e. $W|_{\Gamma_D} = 0$. On the other hand: $P_2Jw = \frac{\partial U}{\partial n}|_{\Gamma}$, where

$$\Delta U = 0, \qquad U|_{\Gamma_D} = 0, \qquad \frac{\partial U}{\partial n}|_{\Gamma_N} = (Jw)|_{\Gamma_N} = \frac{\partial W}{\partial n}|_{\Gamma_N}$$

Thus, $U = W$, i.e. $Jw = P_2Jw$, as stated above.

The immediate consequence of Lemma 2 is that the iteration (15)–(16) results in the exact solution after a single iteration step. Unfortunately, the evaluation of the extension operators P_1, P_2 requires solving special mixed problems, which is as difficult as the solution of the original problem from computational point of view. However, if P_1, P_2 are sufficiently close to the above defined 'ideal' extensions, the operator A might still be a contraction, which assures the convergence of the iteration. In a multi-level context, P_1, P_2 can be defined as some coarse grid approximation of the operators (19)–(20). Note, however, that we have had excellent numerical experiences by using much simpler extension operators as well, based on boundary Shepard interpolation (22), see later.

Now we will show that the underrelaxed version of (15)–(16) remains convergent under much weaker assumptions.

The underrelaxed version is defined by

$$\begin{aligned} v_{n+1/2} &:= J(u_n + \omega \cdot P_1(u_D - u_n)), \\ u_{n+1} &:= J^{-1}(v_{n+1/2} + \omega \cdot P_2(v_N - v_{n+1/2})), \end{aligned} \tag{21}$$

where $\omega > 0$ is a fixed relaxation parameter. The transition matrix of the iteration is now as follows:

$$A = J^{-1}(I - \omega P_2)J(I - \omega P_1)$$

After some manipulations we have:

$$\begin{aligned} A &= (1-\omega)I + \omega \cdot (P - J^{-1}P^*J) + \omega^2 J^{-1}P^*J(I-P) \\ &=: (1-\omega)I + \omega \cdot B + \omega^2 C \end{aligned}$$

Here the operators B and C are bounded in $H^{1/2}(\Gamma)$. The operator B is antisymmetric in the sense that with respect to the scalar product of $H^{1/2}(\Gamma)$, $\langle u, Bv\rangle = -\langle Bu, v\rangle$ holds. Indeed:

$$\langle u, Bv\rangle = \langle u, Pv\rangle - \langle u, J^{-1}P^*Jv\rangle = (Ju)(Pv) - \overline{(Jv)(Pu)},$$

and similarly:

$$\langle Bu, v\rangle = \langle Pu, v\rangle - \langle J^{-1}P^*Ju, v\rangle = \overline{(Jv)(Pu)} - (Ju)(Pv),$$

which proves the antisymmetry. Now let us estimate the norm of the operator A with respect to the $H^{1/2}(\Gamma)$-norm:

$$\begin{aligned} ||Au||^2 &= ||(1-\omega)u + \omega Bu + \omega^2 Cu||^2 \\ &= (1-\omega)^2||u||^2 + 2\omega(1-\omega)\mathrm{Re}\langle Bu, u\rangle + \mathcal{O}(||u||^2 \cdot \omega^2) \end{aligned}$$

Due to the antisymmetry of B, we have: $\mathrm{Re}\langle Bu, u\rangle = 0$, which implies:

$$||Au||^2 = (1-\omega)^2||u||^2 + \mathcal{O}(||u||^2 \cdot \omega^2) = (1-2\omega)||u||^2 + \mathcal{O}(||u||^2 \cdot \omega^2)$$

We have obtained the following theorem:

Theorem 2. *The underrelaxed iteration (21) is convergent for every positive, sufficiently small relaxation parameter ω.*

Consequently, it is expected that if the appearing pure Dirichlet and Neumann subproblems are solved by dipole and monopole formulation, respectively, the sequence of the corresponding discrete problems still result in a convergent iteration.

Extension by boundary Shepard interpolation. Simple and computationally cheap extensions can be constructed by boundary interpolation (more precisely: extrapolation) methods. One of the simplest techniques is based on Shepard interpolation. Consider the boundary interpolation points x_j $(j = 1, \ldots N)$ and define:

Table 3 Norms and spectral radii corresponding to the extension operator based on Shepard boundary interpolation

N	8	16	32	64	128	256	512	1,024
$\|\|P\|\|$	1.315	1.298	1.292	1.290	1.290	1.291	1.291	1.291
$\|\|A\|\|$	0.212	0.269	0.293	0.306	0.315	0.322	0.328	0.333
$\rho(A)$	0.034	0.052	0.059	0.060	0.067	0.074	0.078	0.079

$$(P_1 u)_k := \begin{cases} u_k & (x_k \in \Gamma_D) \\ \dfrac{\displaystyle\sum_{x_j \in \Gamma_D} \dfrac{u_j}{||x_k - x_j||^2}}{\displaystyle\sum_{x_j \in \Gamma_D} \dfrac{1}{||x_k - x_j||^2}} & (x_k \in \Gamma_N) \end{cases} \tag{22}$$

where $u := (u_1, u_2, \ldots, u_N)^*$. The extension operator P_2 can then be defined by $P_2 := (I - P_1)^*$.

For illustration, let Ω be the unit circle and let us discretize the boundary Γ by the equally spaced points $x_1, \ldots, x_N$. Let the Dirichlet part Γ_D (resp. the Neumann part Γ_N) be the upper (resp. lower) half-circle. Table 3 shows the discrete $H^{1/2}(\Gamma)$-norms of the corresponding projector P as well as the transition operator A and its spectral radii with different values of N. The results indicate that, in this case, the iteration is convergent (without underrelaxation) independently of the number of boundary collocation points.

As another illustrative example, consider again the test solution (14) on the square $\Omega := [-1,\ 1] \times [-1,\ 1]$. The boundary collocation points were located in an equidistant way. Along two sides of the square, Dirichlet boundary condition, along the remaining part of the boundary, Neumann boundary condition was prescribed. To approximately solve the mixed problem, the iterative method (15)–(16) was applied with the boundary Shepard extension operator P_1 defined by (22) and $P_2 := (I - P_1)^*$. Table 4 shows the relative L_2-errors (%) with different numbers of boundary points (N), after n_{it} iteration steps. The convergence seems much faster than expected: after 4–5 iterations, the same error levels were achieved as obtained by direct methods (cf. Table 1), independently of the number of boundary collocation points. Moreover, the numerical properties of the appearing matrices are much more advantageous from computational point of view, which allows the use of more efficient solution methods such as the conjugate gradients or other Krylov subspace methods.

Table 4 Relative L_2-errors, iterative solution procedure. Test solution: (14). Mixed boundary conditions

$n_{it} \setminus N$	8	16	32	64	128	256	512
1	21.41	10.16	6.155	4.446	3.633	3.240	3.050
2	17.71	7.991	3.846	1.923	0.9566	0.4647	0.2185
3	17.86	8.040	3.889	1.996	0.9988	0.5048	0.2542
4	17.85	8.039	3.889	1.965	0.9977	0.5037	0.2531
5	17.85	8.039	3.889	1.965	0.9977	0.5038	0.2531
6	17.85	8.039	3.889	1.965	0.9977	0.5038	0.2531

3.1 Multi-level Solution Procedure

The above outlined iteration technique can be speeded up further by using multi-level tools. (For details of multigrid methods, see e.g. [16].) Note that a multi-level method is not necessarily more efficient than a single-level one. It depends on the proper choice of the inter-grid transfer operators (restrictions and prolongations) and the applied smoothing procedure.

In our meshless MFS-based context, let us define a sequence of boundary collocation points:

$$S_k := \{x_1^{(k)}, x_2^{(k)}, \ldots, x_{N_k}^{(k)}\} \qquad (k = 0, 1, \ldots, L),$$

where $N_1 < N_2 < \ldots < N_L$ (L is the maximal level). S_0 is considered to be the coarsest 'grid' (nevertheless, without any grid structure), and S_L is the finest one. The inclusion $S_k \subset S_{k+1}$ is not supposed, however, it is convenient to consider the points of the coarser level to be the 'unification' of some points belonging to the next finer level. At each level, the subproblems must be discretized, and inter-grid transfer operators must be introduced.

In the simplest cascade method, this can be performed as follows. The numbers of boundary collocation points are doubled at each consecutive level, i.e. $N_k = 2N_{k-1}$ ($k = 1, 2, \ldots, L$) and let $x_{2j}^{(k)} := x_j^{(k-1)}$ ($j = 1, \ldots, N_{k-1}$). We consider the coarse-level boundary collocation point $x_j^{(k-1)}$ to be the unification of the fine-level points $x_{2j}^{(k)}$ and $x_{2j+1}^{(k)}$.

In the cascade method, restriction operators are not needed. The prolongation can simply be constructed by 'piecewise constant' way, i.e. if $(\alpha_1^{(k-1)}, \alpha_2^{(k-1)}, \ldots, \alpha_{N_{k-1}}^{(k-1)})$ is a vector of coefficients in the $(k-1)$th level, the prolongated vector is defined by:

$$\alpha_{2j}^{(k)} := \frac{1}{2}\alpha_j^{(k-1)}, \qquad \alpha_{2j+1}^{(k)} := \frac{1}{2}\alpha_j^{(k-1)}$$

Table 5 Relative L_2-errors, multi-level solution procedure. Test solution: (14). Mixed boundary conditions

level	0	1	2	3	4	5	6
N	8	16	32	64	128	256	512
Relative L_2-error (%)	17.86	8.038	3.888	1.965	0.9981	0.5043	0.2532

($j = 1, 2, \ldots, N_{k-1}$, $k = 1, 2, \ldots, L$). As a smoothing procedure, a simple conjugate gradient iteration (applied to the normal equations of (10) and (13)) can be performed. That is, the monopole formulation is applied to the pure Neumann subproblems, while the pure Dirichlet subproblems are treated by the dipole formulation.

Thus, the overall multi-level algorithm is defined as follows:

- Step 0: On the coarsest level S_0, solve the corresponding discrete problem exactly, both in monopole and dipole formulation.
- Step 1: Prolongate the solutions to the next finer level.
- Step 2: Apply several iterations (15)–(16) (or their underrelaxed version (21)) at the current level. To approximately solve the appearing pure Dirichlet and pure Neumann subproblems, use the dipole and the monopole formulation, respectively, and apply several conjugate gradient iterations to the normal equations of (10) and (13), respectively.
- Repeat the procedure from Step 1 until the finest level is reached.

Since both the outer iteration (15)–(16) and the inner conjugate gradient iteration converge rapidly, it is expected that the above multi-level method is more efficient that a single-level method defined on the finest level.

Without going into theoretical details, as a numerical example, consider again the test solution (14) on the square $\Omega := [-1, 1] \times [-1, 1]$. The boundary collocation points were located in an equidistant way and the number of collocation points was doubled at each consecutive level. Along two sides of the square, Dirichlet boundary condition, along the remaining part of the boundary, Neumann boundary condition was prescribed. At each level, only three outer iteration steps (15)–(16) were performed with the boundary Shepard extension operator P_1 defined by (22) and $P_2 := (I - P_1)^*$. In each pure subproblem, five conjugate gradient iterations to the normal equation were applied. Table 5 shows the relative L_2-errors at the different levels (N denotes the number of boundary collocation points at the current level). The results indicate that the multi-level solution algorithm produces almost exactly the same error levels as was shown in Table 4, i.e. the same accuracy can be reached than in the case of single-level solutions. However, the computational cost is significantly reduced due to the fact that the necessary number of the applied iterations is bounded, independently of the actual level.

4 Summary and Conclusions

The Method of Fundamental Solutions was applied to potential problems supplied with mixed boundary conditions. The source and the collocation points were supposed to coincide and were located along the boundary of the domain. This causes singularities in the resulting matrices due to the singularity of the applied fundamental solution. These singularities were removed in different ways. The singularities of the shifted fundamental solutions were eliminated by using truncated fundamental solution or by using the continuous fundamental solution of a fourth-order partial differential operator which approximates that of the original operator. The singularities of the normal derivatives of the shifted fundamental solutions were eliminated by solving an auxiliary pure Dirichlet problem. Similar regularization and desingularization techniques were used for the dipole formulation as well. The dipole formulation results in well-conditioned algebraic systems provided that pure Dirichlet boundary condition is prescribed, while in case of pure Neumann boundary condition, the monopole formulation produces well-conditioned systems. For these pure problems, the familiar conjugate gradient iteration becomes quite efficient due to the fact that the appearing matrices approximate linear operators which are compact perturbations of the identity. In case of mixed boundary condition, the original problem was converted to a sequence of pure problems, the solutions of which converge rapidly to the solution of the original mixed problem. To approximately solve these pure subproblems, the above dipole and monopole formulations were used, which result in a numerically efficient solution technique. The method is embedded in multi-level context in a natural way, which further reduces the numerical complexity. In addition to this, the problem of large and ill-conditioned systems is also avoided.

Acknowledgements The research was partly supported by the European Union (co-financed by the European Social Fund) under the project TÁMOP-4.2.2.A-11/1/KONV-2012-0012.

References

1. C.J.S. Alves, C.S. Chen, B. Šarler, The method of fundamental solutions for solving Poisson problems, in *Proceedings of the 24th International Conference on the Boundary Element Method incorporating Meshless Solution Seminar*, ed. by C.A. Brebbia, A. Tadeu, V. Popov. International Series on Advances in Boundary Elements, vol. 13 (WitPress, Southampton/Boston, 2002), pp. 67–76
2. O. Axelsson, J. Karátson, Superlinearly convergent CG methods via equivalent preconditioning for nonsymmetric elliptic operators. Numer. Math. **99**, 197–223 (2004)
3. C.S. Chen, C.M. Fan, J. Monroe, The method of fundamental solutions for solving elliptic partial differential equations with variable coefficients, in *The Method of Fundamental Solutions – A Meshless Method*, ed. by C.S. Chen, A. Karageorghis, Y.S. Smyrlis (Dynamic Publishers, Atlanta, 2008), pp. 75–105
4. W. Chen, Symmetric bundary knot method. Eng. Anal. Bound. Elem. **26**, 489–494 (2002)

5. W. Chen, L.J. Shen, Z.J. Shen, G.W. Yuan, Boundary knot method for Poisson equations. Eng. Anal. Bound. Elem. **29**, 756–760 (2005)
6. W. Chen, F.Z. Wang, A method of fundamental solutions without fictitious boundary. Eng. Anal. Bound. Elem. **34**, 530–532 (2010)
7. C. Gáspár, Multigrid and multipole techniques in the boundary integral equation methods, in *Boundary Elements: Implementation and Analysis of Advanced Algorithms*, ed. by W. Hackbusch, G. Wittum. Notes on Numerical Fluid Mechanics, vol. 54 (Vieweg-Verlag, Braunschweig/Wiesbaden, 1996), pp. 102–114
8. C. Gáspár, A multi-level regularized version of the method of fundamental solutions, in *The Method of Fundamental Solutions – A Meshless Method*, ed. by C.S. Chen, A. Karageorghis, Y.S. Smyrlis (Dynamic Publishers, Atlanta, 2008), pp. 145–164
9. C. Gáspár, Regularization techniques for the method of fundamental solutions. Int. J. Comput. Methods **10**(2), 1341004-1–1341004-21 (2013)
10. C. Gáspár, A regularized multi-level technique for solving potential problems by the method of fundamental solutions. Eng. Anal. Bound. Elem. (in press, doi: 10.1016/j.enganabound.2014.05.002)
11. M. Griebel, P. Oswald, On the abstract theory of additive and multiplicative Schwarz algorithms. Numer. Math. **70**, 163–180 (1995)
12. A. Karageorghis, G. Fairweather, The method of fundamental solutions for axisymmetric potential problems. Int. J. Numer. Methods Eng. **44**, 1653–1669 (1999)
13. M. Maischak, Multiplicative Schwarz algorithms for the Galerkin boundary element method. SIAM J. Numer. Anal. **38**(4), 1243–1268 (2000)
14. B. Šarler, A modified method of fundamental solutions for potential flow problems, in *The Method of Fundamental Solutions – A Meshless Method*, ed. by C.S. Chen, A. Karageorghis, Y.S. Smyrlis (Dynamic Publishers, Atlanta, 2008), pp. 299–326
15. O. Steinbach, Mixed approximations for boundary elements. SIAM J. Numer. Anal. **38**(2), 401–413 (2000)
16. K. Stüben, U. Trottenberg, *Multigrid Methods: Fundamental Algorithms, Model Problem Analysis and Applications*, in *Lecture Notes in Mathematics*, ed. by W. Hackbusch, U. Trottenberg, vol. 960 (Springer, Berlin, Heidelberg, 1982), pp. 1–176
17. D.B. Szyld, The many proofs on an identity on the norm of oblique projections. Numer. Algorithms **42**, 309–323 (2006)
18. D.L. Young, K.H. Chen, C.W. Lee, Novel meshless method for solving the potential problems with arbitrary domain. J. Comput. Phys. **209**, 290–321 (2005)

Improvements to the Prototype Micro-brittle Model of Peridynamics

Georg C. Ganzenmüller, Stefan Hiermaier, and Michael May

Abstract This paper assesses the accuracy and convergence of the bond-based Peridynamic model with brittle failure, known as the prototype micro-brittle (PMB) model. We investigate the discrete equations of this model, suitable for numerical implementation. It is shown that the widely used discretization approach incurs rather large errors. Motivated by this observation, a correction is proposed, which significantly increases the accuracy by cancelling errors associated with the discretization. As an additional result, we derive equations to treat the interactions between differently sized particles, i.e., a non-homogeneous discretization spacing. This presents an important step forward for the applicability of the PMB model to complex geometries, where it is desired to model interesting parts with a fine resolution (small particle spacings) and other parts with a coarse resolution in order to gain numerical efficiency. Validation of the corrected Peridynamic model is performed by comparing longitudinal sound wave propagation velocities with exact theoretical results. We find that the corrected approach correctly reproduces the sound wave velocity, while the original approach severely overestimates this quantity. Additionally, we present simulations for a crack growth problem which can be analytically solved within the framework of Linear Elastic Fracture Mechanics Theory. We find that the corrected Peridynamics model is capable of quantitatively reproducing crack initiation and propagation.

Keywords Meshless • Simulation • Peridynamics • Crack growth

1 Introduction

Peridynamics (PD), originally devised in 1999 by S. A. Silling [1] is is a relatively new approach to solve problems in solid mechanics. In contrast to the most popular numerical methods for solving continuum mechanics problems, namely the Finite Element Method or the Finite Volume Method, PD does not require a topologically

G.C. Ganzenmüller (✉) • S. Hiermaier • M. May
Fraunhofer Ernst-Mach Institute for High-Speed Dynamics, Freiburg im Breisgau, Germany
e-mail: georg.ganzenmueller@emi.fraunhofer.de

M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations VII*, Lecture Notes in Computational Science and Engineering 100,
DOI 10.1007/978-3-319-06898-5_9

connected mesh of elements. Additionally, PD incorporates the description of damage and material failure from the outset. Within the context of mesh-free methods, Peridynamics can be classified as a Total-Lagrangian collocation method with nodal integration. PD features two classes of interaction models, so called bond-based materials and state-based materials. In the bond-based case, interactions exist as spring-like forces between pairs of particles. The interactions only depend on the relative displacement (and potentially its history) of the interacting particle pair and are thus independent of other particles. This is in contrast to the state-based model where pair-wise interactions also depend on the cumulative displacement state of all other particles within the neighborhoods of the two particles which form the pair.

The scope of this paper is to assess the accuracy and convergence of the linear-elastic, bond-based PD model with brittle failure, known as the prototype micro-brittle (PMB) model in the literature. We investigate the discrete equations of this model, suitable for numerical implementation. It is shown that the widely used discretization approach incurs rather large errors. Motivated by this observation, a new discretization scheme is proposed, which significantly increases the numerical accuracy. As an additional result, we derive equations to treat the interactions between differently sized particles, i.e., a non-homogeneous discretization spacing. This presents an important step forward for the applicability of the PMB model to complex geometries, where it is desired to model interesting parts with a fine resolution (small particle spacing) and other parts with a coarse resolution in order to gain numerical efficiency.

We begin by introducing the basic terminology of bond-based PD. In order to be consistent with the major part of the existing PD literature, we use the following symbols: a coordinate in the reference configuration is denoted with $\boldsymbol{X}$, deformed (current) coordinates are denoted by $\boldsymbol{x}$, such that the displacement is given by $\boldsymbol{u} = \boldsymbol{X} - \boldsymbol{x}$. Bold mathematical symbols like the preceding ones denote vectors, while the same mathematical symbol in non-bold font refers to its Euclidean norm, e.g. $x = |\boldsymbol{x}|$.

The governing equation for a PD continuum is given by

$$W(\boldsymbol{X}, t) = \frac{1}{2} \int_{\mathcal{H}_\delta} \omega(\boldsymbol{X}' - \boldsymbol{X})\, w\left[\boldsymbol{u}(\boldsymbol{X}', t) - \boldsymbol{u}(\boldsymbol{X}, t), \boldsymbol{X}' - \boldsymbol{X}\right] \mathrm{d}V_{\boldsymbol{X}'}, \tag{1}$$

where $W(\boldsymbol{X}, t)$ is the energy density at a point located at $\boldsymbol{X}$ in the reference configuration, and displaced at time t by an amount $\boldsymbol{u}(\boldsymbol{X}, t)$. $w\left[\boldsymbol{u}(\boldsymbol{X}', t) - \boldsymbol{u}(\boldsymbol{X}, t)\right]$ is the *micropotential*, which describes the strain energy due to the relative displacement of a pair of points located at $\boldsymbol{X}$ and $\boldsymbol{X}'$. The assumption that the strain energy density depends only on pairs of interacting volume elements leads to the restriction of a fixed Poisson ratio [2] of $1/3$ in 2D (1/4 in 3D). The function $\omega(\boldsymbol{X}' - \boldsymbol{X})$ is a weight function which modulates the pair interaction strength depending on spatial separation, and $V_{\boldsymbol{X}'}$ is the volume associated with a point.

Referring to Fig. 1, the integration domain $\mathcal{H}_\delta$ is the full disc (full sphere in 3D) around $\boldsymbol{X}$ described by the radial cutoff δ, and is termed the *horizon*. Within the PD

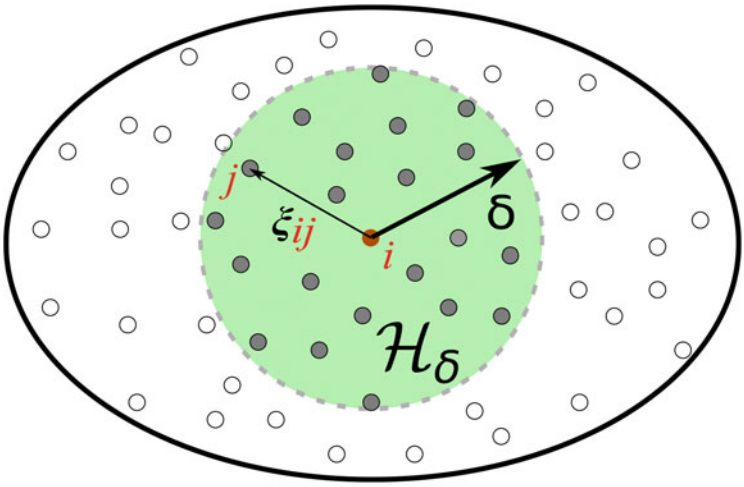


Fig. 1 Peridynamics is a method for solving problems in solid mechanics. A body is discretized with a set of integration nodes, which form the reference configuration. Within this reference configuration, each source node interacts with other nodes that are located within a finite horizon $\mathcal{H}_\delta$, centered on the source node. The interactions are termed bonds. Peridynamics is a non-local method, because not only nearest, or, adjacent, neighbors are considered. The figure above depicts a single source node i with a horizon given by the radial cutoff δ. Bonds exist between node i and all other nodes j which are inside $\mathcal{H}_\delta$. Upon deformation of the bonds, forces are projected along the reference bond vectors $\boldsymbol{\xi}_{ij}$ such that solid material behavior is obtained

picture, the strain energy is conceptually stored in *bonds* that are defined between all pairs of points $(\boldsymbol{X}, \boldsymbol{X}')$ located within $\mathcal{H}_\delta$. Thus, a bond vector in the reference configuration is given by $\boldsymbol{\xi} = \boldsymbol{X}' - \boldsymbol{X}$, and the relative bond displacement due to some deformation at time t is $\boldsymbol{\eta}(t) = \boldsymbol{u}(\boldsymbol{X}', t) - \boldsymbol{u}(\boldsymbol{X}, t)$. The bond distance vector in the current configuration is therefore written as $\boldsymbol{r}(t) = \boldsymbol{\eta}(t) + \boldsymbol{\xi}$.

With this notation, and dropping the explicit dependence on time, Eq. (1) is written in a more compact form as

$$W(\boldsymbol{X}) = \frac{1}{2} \int_{\mathcal{H}_\delta} \omega(\boldsymbol{\xi}) w(\boldsymbol{r}, \boldsymbol{\xi}) \mathrm{d}V_{\boldsymbol{X}'}, \tag{2}$$

The factor of $1/2$ in the above equation arises because each bond is defined twice, once originating at $\boldsymbol{X}$ and pointing to $\boldsymbol{X}'$, and again via its antisymmetric counterpart pointing from $\boldsymbol{X}'$ to $\boldsymbol{X}'$. The forces within the bond-based PD continuum are obtained by taking the derivative of the micropotential with respect to the bond distance vector. The *microforce* between two bonded points is thus

$$\boldsymbol{f}(\boldsymbol{r}, \boldsymbol{\xi}) = -\frac{\partial w(\boldsymbol{r}, \boldsymbol{\xi})}{\partial \boldsymbol{r}}, \tag{3}$$

yielding the acceleration $\boldsymbol{a}(\boldsymbol{X})$ of a point with mass density ρ due to all its neighbors within $\mathcal{H}_\delta$:

$$\rho \boldsymbol{a}(\boldsymbol{X}) = \int_{\mathcal{H}_\delta} \omega(\boldsymbol{\xi}) \boldsymbol{f}(\boldsymbol{r}, \boldsymbol{\xi}) \mathrm{d}V_{X'}. \tag{4}$$

For implementation in a computer code, Eqs. (2) and (4) need to be discretized. This process requires the division of the continuous body to be simulated into a number of distinct nodes with a given subvolume, subject to the constraint that the sum of all subvolumes equals the total volume of the body. These nodes are termed particles henceforth and the Peridynamic bonds exist between these particles. The most straightforward discretization approach is nodal integration, which is used in almost all publications dealing with PD up to date. Referring to Fig. 1, particle i is connected to all neighbors j within the horizon δ. Dropping the explicit dependence on $\boldsymbol{X}$, the discrete expression for the energy density of a particle i reads:

$$W_i = \sum_{j \in \mathcal{H}_\delta} \omega(\xi_{ij}) V_j w_{ij}(\boldsymbol{r}_{ij}, \boldsymbol{\xi}_{ij}), \tag{5}$$

and

$$\boldsymbol{a}_i = \frac{1}{m_i} \sum_{j \in \mathcal{H}_\delta} \omega(\xi_{ij}) V_i V_j \boldsymbol{f}_{ij}(\boldsymbol{r}_{ij}, \boldsymbol{\xi}_{ij}). \tag{6}$$

These discretizations represent simple Riemann sums, i.e., piecewise constant approximations of the true integrals. The object of this work is to quantify the errors incurred by this approach, but before doing so, we introduce a specific form of the pairwise force function which is compatible with linear elastic continuum behavior and supports a brittle fracture mechanism.

2 Linear Elasticity in Peridynamics

In order to establish the link with linear elasticity, i.e., a Hookean solid, Silling [2] introduced the *Prototype Microbrittle Material* (PMB) model, with a microforce that depends linearly on the bond stretch $s = |\boldsymbol{\xi} + \boldsymbol{\eta}|/|\boldsymbol{\xi}|$. The bond stretch can be thought of as a pairwise one dimensional strain description of the material, and a full strain tensor can indeed be derived from an ensemble of bond stretches [3]. A microforce which is linear in s is therefore in agreement with Hooke's law.

Here, we employ the following microforce which:

$$f(s, \xi) = -cs/\xi, \tag{7}$$

with proportionality constant c. The corresponding micropotential is obtained by integrating the microforce w.r.t. displacement.

$$w(s) = -\int f(s,\xi)\mathrm{d}\eta = \frac{1}{2}cs^2. \tag{8}$$

Note that the expressions for the microforce and the micropotential differ from Silling's original work by a factor of ξ. This change is purely for consistency reasons, because, in our opinion, the energy density should not contain a reference to a length scale. The modification will be absorbed into the proportionality constant c which is yet to be determined.

The weight function is chosen as a simple step function,

$$\omega(\xi_{ij}) = \begin{cases} 1 & \text{if } \xi_{ij} \leq \delta \\ 0 & \text{if } \xi_{ij} > \delta \end{cases}, \tag{9}$$

which allows for a compact notation as it can be absorbed into the summation operator of the discretized expressions, i.e., $\sum_{j\in\mathcal{H}_\delta} \omega(\xi_{ij}) = \sum_{j\in\mathcal{H}_\delta} 1$. The effects of using different weight functions have been studied in detail [4]. No significant benefits were observed when using different forms of the weight function for the purpose of simulating structural response problems, however, the weight function affects the dispersion of waves.

Damage and failure are incorporated by keeping track of the history of a bond stretch state. We fail individual bonds by permanently and irreversibly deleting them once they are stretched beyond a critical stretch value s_c.

The remaining constant c is determined by requiring the Peridynamic expression for the energy density, Eq. (2) to be consistent with the result from linear elasticity theory, $W_{el.}$:

$$\frac{1}{2}\int_{\mathcal{H}_\delta} \omega(\xi_{ij})w(s)\mathrm{d}V_{\boldsymbol{X}'} = W_{el.}, \tag{10}$$

In the 3D case of pure dilation or compression, cf. Eq. (32) in the Appendix, we have $W_{el.}^{3D} = 9Ks^2/2$, where K is the bulk modulus and s is the strain along any of the Cartesian directions. Note that for isotropic strain field, the strain and the stretch of any bond coincide. Integrating the Peridynamic energy density expression for this strain field in spherical coordinates, we have

$$\frac{1}{2}\int_{\mathcal{H}_\delta} \omega(\xi_{ij})w(s)\mathrm{d}V = \frac{1}{2}\int_0^\delta\int_0^\pi\int_0^{2\pi} \omega(\xi_{ij})\frac{1}{2}cs^2\xi^2\mathrm{d}\xi\,\sin(\phi)\mathrm{d}\phi\mathrm{d}\theta = \frac{\pi cs^2\delta^3}{6}. \tag{11}$$

Equating this result with the continuum theory expression for the elastic strain energy, the constant c is obtained as:

$$c = \frac{6K}{\pi\delta^3}. \tag{12}$$

This approach of determining c is correct for the continuous integral expressions upon which PD theory is based. However, in combination with the discrete expression given by Eq. (5), the results of a numerical computation of the energy density are inaccurate, as exact analytic integration is combined with piecewise constant approximation of the integrals. Furthermore, the analytic integration performed in Eq. (11) assumes that each node is completely contained in the bulk of the body, such that a spherical integration domain exists around the central node. This assumption is certainly not true at the boundaries of the body. The errors incurred by this approach are rather large and, what is worse, does not converge to zero upon increasing δ. This behaviour would be expected, as more as an increase in δ at fixed particle spacing means that more integration nodes are used to sample the field variables. Before we quantify these errors, we introduce an alternative approach to determine c which relies on exact error cancellation such that the energy density is exactly reproduced for a given strain field.

2.1 An Improved Route for Determining the PMB Proportionality Constant

Instead of deriving the proportionality constant c by exact analytic integration, we propose to use the same integral approximation as is used for discretizing the PD energy density integral or acceleration expression. This means that we use a piecewise constant approximation for Eq. (10), as shown in Fig. 2:

$$\frac{1}{2}\int_{\mathcal{H}_\delta} w(s)\mathrm{d}V_{X'} \approx \frac{1}{2}\sum_{j\in\mathcal{H}_i} w(s)V_j = W_{el.} \tag{13}$$

Inserting the micropotential and the 3D pure dilation result for the continuum strain energy density in the above equation, we obtain the proportionality constant as

$$c_i = \frac{18K}{\sum\limits_{j\in\mathcal{H}_i} V_j}. \tag{14}$$

In this formulation, the dependence of c_i on the horizon δ is now only implicit through the number of particles contributing to the sum in the denominator. A particle at a free surface of a body will have a different number of neighbors compared to a particle in the bulk. This effect is accounted for with our discrete expression for c_i,

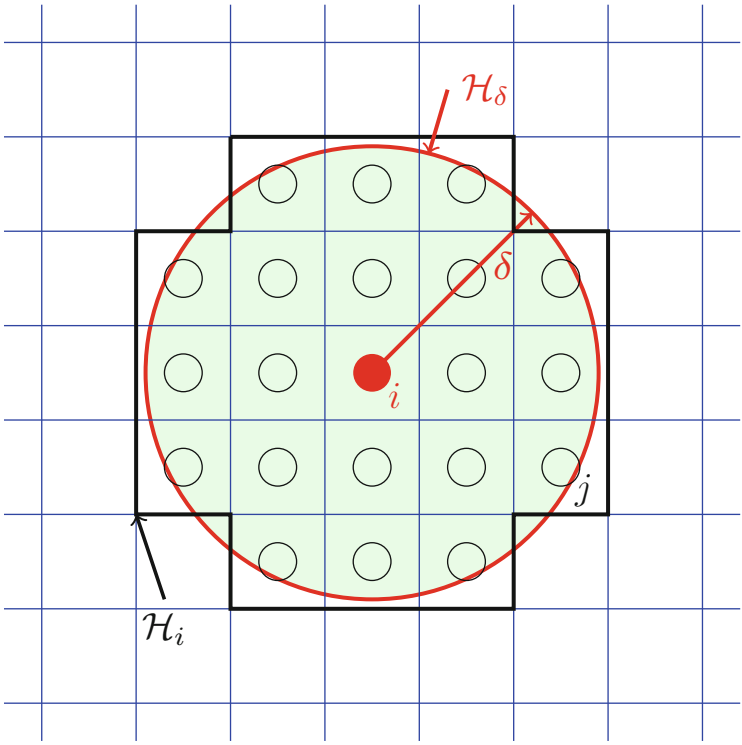


Fig. 2 Piecewise constant approximation of the Peridynamic neighborhood volume. In the original derivation of Peridynamics, the neighborhood boundary $\mathcal{H}_\delta$ is defined as a smooth region in space given by the radial cutoff δ. For a piecewise constant approximation of the integrals of the Peridynamic theory suitable for computer implementation, the neighborhood also needs to be defined in a discrete manner: here, we define the piecewise constant neighborhood approximation as the volume of all particles touched by the radial cutoff δ

as opposed to the original expression, Eq. (12), which is only valid for the bulk. This normalization is similar to a Shepard correction of the shape functions encountered in other meshless methods such as Smooth-Particle Hydrodynamics [5, 6], where it restores C^0 consistency, i.e., the ability to approximate a constant field. At the same time, it is this local dependence which allows us to easily introduce different spatial resolutions and horizons. It is important at this point to discuss the conservation of momentum. In the original formulation of the PMB model, the proportionality constant c is the same for all interacting particles. Therefore, $\boldsymbol{f}_{ij} = -\boldsymbol{f}_{ji}$, and, as the forces are aligned with the distance vector between particles i and j, both linear and angular momentum are conserved. In the approach proposed here, $\boldsymbol{f}_{ij}$ is not necessarily equal to $-\boldsymbol{f}_{ji}$, as the particle volume sum over $\mathcal{H}_i$ is not guaranteed to equal the particle volumes sum over $\mathcal{H}_j$. Thus $c_i \neq c_j$, in general. We therefore enforce symmetry in the following manner:

$$c_{ij} = \frac{c_i + c_j}{2} \tag{15}$$

The full expressions for the potential energy of a particle and its acceleration, as required for implementation in a computer code, are then

$$E_i = \sum_{j \in \mathcal{H}_i} V_i V_j c_{ij} s_{ij}^2, \tag{16}$$

and

$$\boldsymbol{a}_i = \frac{1}{m_i} \sum_{j \in \mathcal{H}_i} V_i V_j c_{ij} s_{ij} \frac{1}{\xi_{ij}} \frac{\boldsymbol{r}_{ij}}{r_{ij}}. \tag{17}$$

We note that the existing body of publications on Peridynamics recognizes that the analytic integration approach for determining the proportionality constant, cf. Eq. (12), is not exact when used together with a discrete set of nodes. Different approaches have been taken to address this problem: Parks et al. [7] describe an algorithm which approximately accounts for the fact that particles near the edge of the horizon have a volume which is only partially within the horizon. Bobaru and Ha [8] describe a similar, yet slightly more accurate algorithm, which rests on the assumption that nodes are placed on quadratic or cubic lattices. Neither of these corrections is able to calculate the sum of volumes enclosed by the horizon exactly. Below, exemplary comparison is made between the algorithm described by Parks et al. and the approach proposed here, which calculates this volume exactly, regardless of whether a regular grid is employed or not.

3 Results

3.1 Comparison of the Original PMB Model with the Improved Model

This section presents two examples to assess the accuracy of the original PMB model and the normalization procedure proposed in this work. We show that the energy density and speed of sound are exactly reproduced using our method, while the original method yields considerable errors. Finally, we investigate a mode-I crack opening example with our modified PD scheme, where a failure criterion based on the Griffith energy release rate correctly reproduces results from Linear Elasticity Fracture Mechanics Theory.

3.1.1 Energy Density

The ability to reproduce the correct strain energy for a homogeneous deformation is the most basic task any simulation method for solid mechanics should be able to handle with good accuracy. We consider a cube of a material under periodic boundary conditions. The bulk modulus is 1 GPa, and the material is discretized using a cubic lattice with spacing $\Delta x = 1$ m. In order to effect a homogeneous deformation, all directions are scaled using a factor of $l = 1.05$, leading to volume change of 15.8 %. We measure the Peridynamic strain energy density, W_{PD} by summing over all bond energies and dividing by the cube volume. The exact strain energy density is calculated using Eq. (32), such that a relative error can be defined:

$$\Delta W = \frac{W_{PD} - W_{el.}^{3D}}{W_{el.}^{3D}} \,. \tag{18}$$

Figure 3 shows the relative errors for a range of different horizon cutoffs $\delta \in [2\Delta x \ldots 6\Delta x]$, such that the number of particles within the horizon varies from 32 to 924, while the total number of particles in the system is kept constant. This type of convergence is denoted *m-convergence* in the Peridynamics literature [9]. Two different methods with analytic expressions for the proportionality constant c are compared against the normalization approach proposed here: the original method, cf. Eq. (12), and the volume correction first presented by Parks et al. [7], which approximately accounts for the fact that finite volume region of particles near the

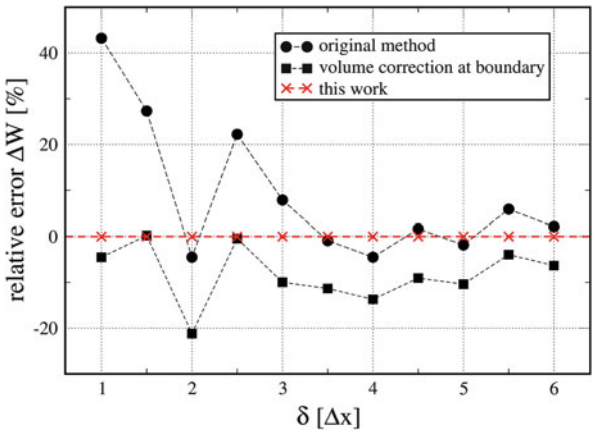


Fig. 3 This graph shows relative errors of the Peridynamic strain energy density for a pure dilation strain field. *Black disks* denote results obtained with the original PMB method [2] which uses analytic integration for the determination of the micropotential proportionality constant. *Black squares* denote the results obtained with the original PMB method and the volume correction approach due to Parks et al. [7]. *Red symbols* show the results obtained using the here proposed normalization approach for the micropotential constant

edge of the horizon are only partially within within the horizon. We observe that the original approach shows relative errors in excess of 30 %. What is worse, is that the errors do not converge monotonously as one increases the horizon. As more particles are within the horizon, the accuracy of the numerical integration should increase, because the strain field is sampled using more integration points. In this particular case, the horizon is the only discretization resolution variable available due to the scale invariance implied by the absence of free surfaces. The volume correction approach due to Parks et al. fares only slightly better, but generally underestimates the strain energy density. We note that Hu et al. [10] describe similar observations for the strain energy density in bond based Peridynamics with the constant micromodulus function. In contrast, the normalization proposed here reproduces the strain energy density exactly for any micromodulus.

3.1.2 Wave Propagation

The second example investigates the propagation of a pressure pulse. To this end, we consider a bar of size $500 \times 4 \times 4$ m^3, discretized using a cubic lattice with $\Delta x = 1$ m. We set $K = 1$ Pa, $\rho = 1$ kg/m^3 and $\delta = 2.5\Delta x$. Periodic boundaries are applied along the y- and z-direction in order to suppress free surface effects. The pulse is initiated by a displacement perturbation of Gaussian shape at one end,

$$\boldsymbol{x} = \boldsymbol{X} + 0.02\ \text{m} \times \exp\left(-\frac{\boldsymbol{X}\cdot\boldsymbol{X}}{100\ \text{m}^2}\right)\boldsymbol{e}_x, \tag{19}$$

where $\boldsymbol{e}_x$ is the unit vector in the Cartesian x-direction. The simulation is then run until the pressure pulse has reached the right end of the bar. The time-step is set to $\Delta t = 0.1$ s, which is stable according to CFL analysis. Following [11], the theoretical value for the longitudinal speed of sound is

$$c_l = \sqrt{\frac{K + \frac{3}{4}G}{\rho}}, \tag{20}$$

where $G = 3K(1-2\nu)/[2(1+\nu)]$ is the shear modulus, and ν is Poisson's ratio. As the 3D Peridynamic model under consideration has a fixed Poisson ratio $\nu = 1/4$ [1], we obtain $c_l = 4.24$ m/s. Figure 4 compares this theoretical prediction with the results of Peridynamics simulation that employ the original analytic integration approach for determining the amplitude constant c of the micropotential, and the normalization approach proposed here. It is evident from this comparison that the original approach severely overestimates the wave propagation speed. This is in agreement with the observation, that the original approach overestimates the energy density, leading to a system which is effectively too stiff. In contrast, the normalization procedure for determining c reproduces the theoretical wave propagation speed very well.

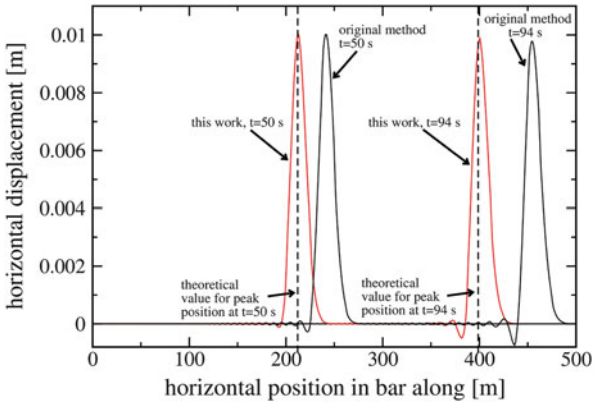


Fig. 4 Sound wave propagation. A horizontally oriented bar of dimensions 500 * 4 * 4 m is loaded using a Gaussian shaped displacement at the left. This initial perturbation causes a Gaussian-shaped pressure pulse to travel to the right at the longitudinal speed of sound. Shown above are theoretical values (*vertical dashed lines*), where the center of the pressure pulse should be located after elapsed time periods of 50 and 94 s, respectively. The results of the Peridynamics simulations, (i) using the original approach, and (ii) using the normalization for the micropotential amplitude are shown as *black* and *red lines*, respectively. We note that the original approach over-predicts the speed of sound by 13 %, while the here proposed normalization approach agrees with the theoretical result within an error margin of less than 1 %

To investigate the performance of the normalization approach in the case of non-uniform particle spacing, we now consider a mesh of the same bar as above, which is generated via a stochastic procedure. We use a Delaunay-based meshing algorithm to generate tetrahedral elements. These elements are subsequently replaced by particles. Each particle is assigned the volume of the tetrahedron it replaces. The particle's mass is obtained from the volume and the mass density, $m = V\rho$. Figure 5 shows a section of the bar in both the tetrahedron and particle representation. To realize a challenging test, the tetrahedral mesh was intentionally generated such that small angles and large variations in the tetrahedron volumes are achieved. The resulting particle configuration is therefore strongly polydisperse with a ratio of smallest to largest radius of 100. Because no characteristic length-scale (such as the lattice spacing above) is now present, we adjust the Peridynamic horizon for each particle separately, such that the neighborhood contains 30 neighbors. Three different initial tetrahedron meshes of different resolutions are used to conduct a convergence study for our PMB normalization approach. The coarsest mesh contains 17,211 tetrahedrons, and two more finely resolved meshes are obtained by repeated splitting of the elements, such that the finest mesh has 70,381 elements.

The results are given in Fig. 6. We observe that pressure pulse is much broader when compared to the results of the uniform particle configuration shown in Fig. 4, and that oscillations travelling behind the main pulse are more pronounced. This is

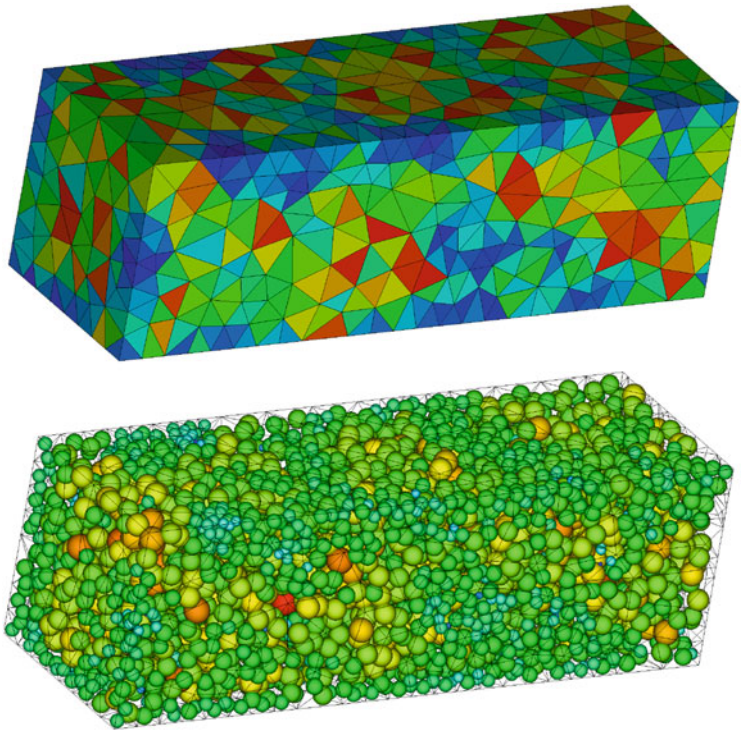


Fig. 5 Generation of non-uniform particle configurations. *Top*: a volume is meshed using regular tetrahedrons. Form this mesh, a particle configuration is obtained by placing particles at the tetrahedron barycenter, and assigning the tetrahedron's volume and mass to the particles. Color coding represents volume, increasing from *blue* to *red*

not surprising, as it is well known that wave propagation is affected by discretization effects: partial reflections occur always when a wave is transmitted between regions of space that are discretized using different resolutions. These reflections cause dispersion and reduction in the observed wave speed propagation speed. As the discretization length scale becomes small compared to the wavelength, these effects disappear. We therefore expect convergence of the location of the pressure pulse to its theoretical position at a given time, and return of its shape back to the initial Gaussian shape, as the particles are more finely resolved. The simulation results shown in Fig. 6 support these statements: as the resolution is enhanced, the wave speed tends towards its theoretical value and the pressure pulse shows less oscillations. We therefore conclude that our approach of handling interactions between Peridynamic particles of different size is correct.

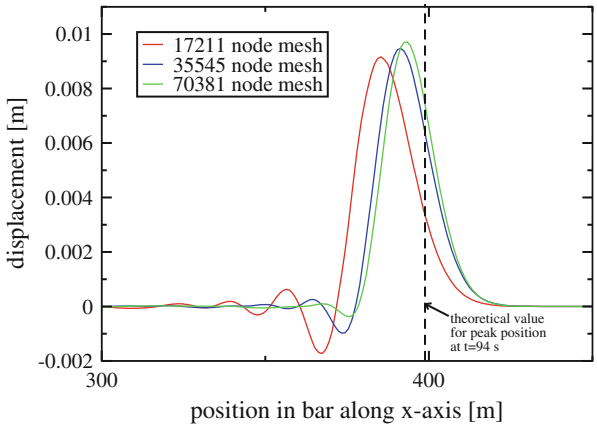


Fig. 6 Propagation of a pressure pulse in a long bar which is discretized using irregular particle positions and polydisperse particle size distributions. The geometry and parameters are the same as for Fig. 4, but instead of a regular mesh we employ the discretization approach via a stochastic tetrahedral mesh outlined in Fig. 5. The *vertical dashed line* indicates the position where the pressure pulse should be, according to the exact wave propagation speed. The Peridynamic simulations show convergence to this exact result upon increasing the number of particles used for discretizing the bar

3.2 Fracture Energy

Traditionally, continuum mechanics is formulated using a set of partial differential equations which describe temporal and spatial evolution. These equations require smooth solutions with well defined gradients. Therefore, discontinuities in the material, such as cracks, cannot emerge naturally within the solution manifold. In contrast, Peridynamics circumvents this problem by employing an integral description for the evolution equations. Due to its simple form, the PMB model in particular is well suited to model arbitrary crack initiation and propagation phenomena. A number of studies have used the PMB model to study crack propagation speed, crack branching as well as coalescence of individual cracks [12–15]. However, to the best of these authors' knowledge, no quantitative assessment of the accuracy of PMB simulations relative to analytic solutions for modelling crack initiation and propagation has been published to date. The main reason for this shortcoming is probably the fact that the original formulation of the PMB model using the analytic integration approach for determining the micropotential amplitude inflicts unacceptably large errors already for the energy density. This implies that no quantitatively correct modelling of crack processes could be carried using the original PMB approach. However, the above cited studies demonstrate that the original PMB model is very well suited to qualitatively model complex crack growth phenomena, including the interaction of multiple cracks with each other. In this section, we demonstrate the our normalization approach for determining the

micropotential amplitude can be used to quantitatively reproduce analytic solutions obtained from Linear Elastic Fracture Mechanics (LEFM) Theory.

A useful crack propagation theory for numerical simulations must be based on criteria which are independent of the discretization length scale. If length scale-dependent measures such as stress are used instead, no convergence of the loads required to propagate a crack can be achieved because finer resolution always implies a higher stress concentrations. One useful criterion is the Griffith energy release rate, i.e., the energy required to separate a body by generating two free surfaces, one to either side of a crack area. The energy release rate is defined as energy divided by area and is therefore an intensive measure for the resistance of a body against cracking. In the discrete setting of a numerical simulation, the energy release rate incorporates the discretization length scale and thus provides a failure criterion which is independent of discretization. This implies that a crack growth simulation based on such a failure criterion can converge upon discretization refinement. A Peridynamic failure criterion based on the Griffith energy release rate has been first published by Silling and Askari [2]. Here, we roughly follow their approach, but restrict ourselves to plane-strain conditions as LEFM Theory provides useful analytic solutions to compare against in this case.

Because PMB interactions are formulated in terms of bond-wise micropotentials, a failure criterion is required which links the micropotential to the energy release rate. Such an expression can be obtained by considering a pure dilation stretch state of a Peridynamic material and summing the energy stored in all those bonds which cross a hypothetical unit fracture surface. The resulting normalized energy per area, which is a function of the bond stretch and the bulk modulus, can be equated with the energy release rate. From this relation a critical bond stretch can be obtained at which the bond should fail in order to yield a given energy release rate. Figure 7 shows how Peridynamic bonds which are connected to a particular central node interact across a hypothetical fracture surface. An interaction volume is defined as as the spatial volume occupied by these bonds. For a given fracture surface, a manifold of interaction volumes exist. The magnitude of these volumes depends on the distance of the central node away from the fracture surface. Thus, we obtain the Peridynamic energy release rate, $G_{I,PD}$, by integrating the product of micropotential and interaction volume over all values of the distance of the central node to the fracture surface. Referring to Fig. 7, this integral is given by:

$$\begin{aligned} G_{I,PD} &= 2 \int_{h=0}^{\delta} w(s)\ V_c(h,\delta)\mathrm{d}h \\ &= 2 \int_{h=0}^{\delta} \left[\frac{1}{2} cs^2\ t\delta^2 \arccos\left(\frac{\delta - h}{\delta}\right) - (\delta - h)\sqrt{2\,\delta h - h^2} \right] \mathrm{d}h \\ &= \frac{2}{3} cs^2 t \delta^3. \end{aligned} \tag{21}$$

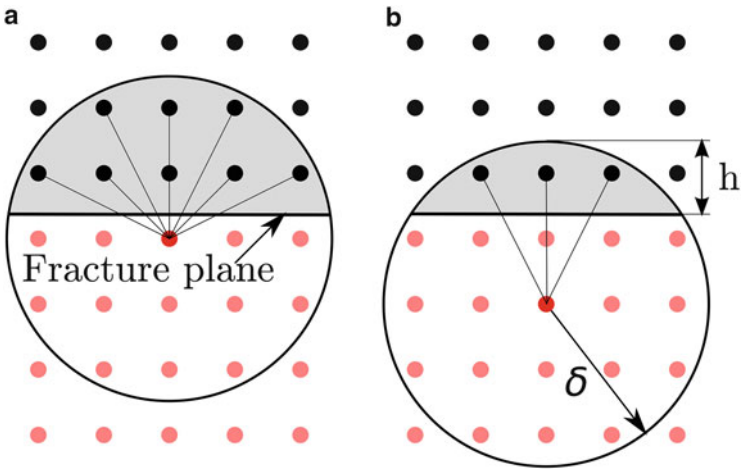


Fig. 7 Peridynamic bond interactions across a hypothetical fracture surface in a 2D plane strain model. Configurations (**a**) and (**b**) show two examples for bonds which traverse a hypothetical fracture surface. The total energy which is set free if the hypothetical fracture surface becomes real is the sum of the energies stored in all the bonds between the *red* and *black particle* half-spaces. While it is in principle possible to enumerate these bonds and perform an explicit summation, this approach is cumbersome in practice. Instead, we consider the interaction volume to either side of the fracture surface, which is given by the plane thickness multiplied with the circular segment (*gray*), $V_c(h,\delta) = t\delta^2 \arccos\left(\frac{\delta-h}{\delta}\right) - (\delta - h)\sqrt{2\,\delta h - h^2}$. The energy density of each configuration is given by the product of the the micropotential and the interaction volume. Finally, the energy release rate is obtained by integrating the energy density over all configurations by varying h

Note that the factor of 2 in front of the integral stems from the fact that we have two interaction volumes, one to either side of the hypothetical fracture surface. The factor t above is the thickness of the plane-strain model. Requiring that the Peridynamic energy release rate matches a specified energy release rate, $G_{I,PD} = G_I$ we obtain the critical bond stretch at failure as:

$$s_c = \sqrt{\frac{3G_I}{2ct\delta^3}}. \tag{22}$$

A useful test for the above expression is delivered by LEFM Theory, which provides analytic solutions that predict the onset of crack growth for some simple models. One such model is a rectangular patch of an elastic material with an existing sharp crack on one side, which is stretched by applying tractions, see Fig. 8. For prescribed values of the energy release rate and the Young's modulus, a critical traction is predicted by LEFM Theory when failure should occur by abrupt propagation of the initial crack through the entire patch. For this geometry, the critical traction that leads to failure is known to be [16]

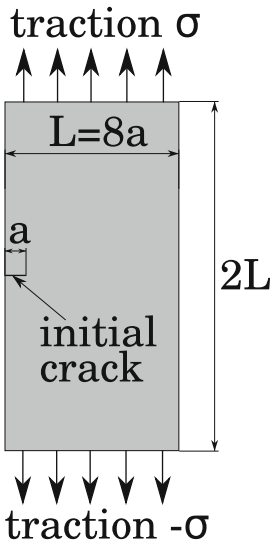


Fig. 8 Sketch of the geometry used for the crack propagation analysis. A rectangular patch of a linear-elastic material is stretched by applying tractions to the top and bottom side. The patch features an initial crack which serves to effect stress concentration at the crack tip. This geometry and loading scenario can be solved analytically solved for a critical traction which causes the crack to grow using Linear Elastic Fracture Mechanic Theory

$$\sigma_F = K_I \frac{1}{\sqrt{\pi\, a}} \left(1.12 - 0.23\, \frac{a}{L} + 10.6\, \frac{a^2}{L^2} - 21.7\, \frac{a^3}{L^3} + 30.4\, \frac{a^4}{L^4} \right)^{-1}. \tag{23}$$

Here, σ_F is the traction applied to the top and bottom of the patch which causes the crack to propagate, a is the initial length of the crack, L is the width of the patch, and K_I is the fracture toughness. In plane strain, the fracture toughness can calculated from the Griffith energy release rate G_I, the Young's modulus of the system, and the Poisson ratio:

$$K_I = \sqrt{\frac{G_I\, E}{1 - \nu^2}} \tag{24}$$

With the values $E = 10^4$ Pa, $\nu = 1/3$, $G_I = 1$ J/m^2, $L = 1$ m and $a = L/8$, we obtain the failure traction as $\sigma_F = 146.9$ Pa. This result will serve as the reference solution against which the normalized PMB model presented in this work will be compared. Peridynamic simulations were carried out using a square lattice discretization of this geometry with seven different lattice constants ranging from 0.005 to 0.04 m, resulting in total particle numbers from 1,250 to 80,000. Tractions were realized by gradually applying opposite forces to the top and bottom row of particles, effecting a gradual stretch of the patch. The forces were ramped up in time such that a displacement velocity 10^4 times slower than the speed of sound in the patch was achieved. Under these conditions, the simulation can be effectively considered quasi-static. Figure 9 shows a snapshot of the simulation with the highest resolution, just before the crack starts to grow. In Fig. 10, the traction values are reported for each resolution, when the crack starts to grow. These data points suggest linear convergence of the critical traction towards the analytic result from above: the

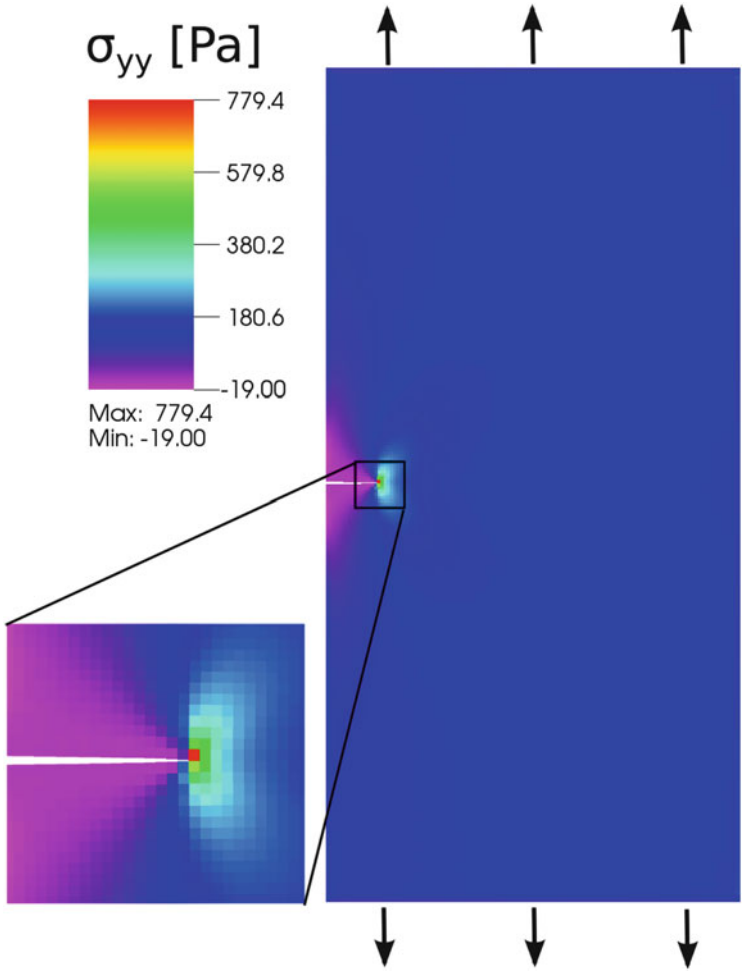


Fig. 9 Peridynamic simulation of crack propagation. Shown is a snapshot of the simulation with the finest resolution. The color-coding represents the yy-component of the stress tensor. The *zoomed-in area* shows the stress concentration at the crack tip

extrapolated infinite-resolution simulation value is 144.4 ± 1.5 Pa, while the analytic result is 146.9 Pa. The agreement between these results is very good and we attribute the remaining difference to the fact that the initial crack does not, depending on the actual particle spacing, align perfectly with the particles. This observation can also explain the scattering of the data points around the linear fit, because, the simulated initial crack is sometimes shorter or longer by one lattice constant when compared to what it should be. Nevertheless, we note that the simple normalized PMB model is highly successful at predicting the correct stress at the crack tip which causes the crack to grow.

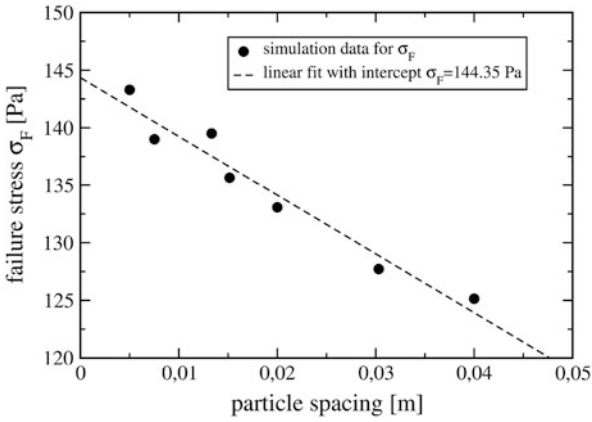


Fig. 10 Convergence of the critical tractions required to cause abrupt crack propagation. As the particle spacing is reduced, linear convergence towards the exact result $\sigma_F = 146.9$ PA is observed

4 Discussion

We have shown that the discrete implementations of the original formulation of the Prototype-Microbrittle Model of linear elasticity in Peridynamics suffers from severe inaccuracies. The origin of this deficiency is traced back to the way how the micropotential proportionality constant is derived. The original approach employs exact analytic integration for this quantity. In a numerical implementation, however, field variables depending on the micropotential are evaluated using non-exact integration rule, e.g., piecewise constant integration via the Riemann sum. The inconsistency between these different integration approaches causes inaccuracies. To resolve this problem, we have modified the PMB model such that the same numerical integration rule is used for determining both the micropotential proportionality constant and the field variables. As an additional result, interactions between particles with different sizes and different Peridynamic horizons can be natively treated using our modification. The correctness of the new approach is validated by simulating the propagation of sound waves, where very good agreement with the theoretical prediction is observed. It is instructive to interpret our modification as a normalization procedure, which performs so well because it effects error cancellation. The modified PMB scheme bears strong similarity to other meshless simulation methods such as Smooth-Particle Hydrodynamics, where such a normalization is known as the Shepard correction. Because Peridynamics is most useful for dealing with material discontinuities, we also consider a crack initiation and propagation example. Here, a patch of an elastic material with a pre-existing crack is pulled apart. Once a critical traction is reached, the stress concentration at the existing crack tip cause the crack to grow abruptly and cause complete separation of the patch. Peridynamic simulations of this experiment with the modified PMB model show linear convergence to the exact critical traction as the discretization resolution is enhanced. Much praise has been granted in advance to Peridynamics as a method specifically apt to handle complex crack growth phenomena. The

simulations reported herein constitute the the first quantitative demonstration that Peridynamics is indeed able to correctly predict failure in agreement with exact analytic solutions.

Appendix

Strain Energy Density

In the continuum theory of linear elasticity, the stress tensor $\boldsymbol{\sigma}$ is obtained from a linear relationship between the stiffness tensor $\boldsymbol{C}$ and the strain tensor $\boldsymbol{\epsilon}$,

$$\sigma_{ij} = C_{ijkl}\epsilon_{kl}. \tag{25}$$

Employing Voigt notation [17] to reduce the dimensionality of the above tensors, the stiffness tensor is expressed as a 6×6 matrix in terms of bulk modulus K and Poisson's ratio ν as,

$$\boldsymbol{C} = \frac{3K}{1+\nu}\begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2-\nu & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2-\nu & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/2-\nu \end{bmatrix}, \tag{26}$$

and the symmetric stress and strain tensors reduce to vectors with six entries:

$$\boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \epsilon_{xy} \\ \epsilon_{xz} \\ \epsilon_{zx} \end{bmatrix}; \; \boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{xz} \\ \sigma_{zx} \end{bmatrix} \tag{27}$$

For a general strain state, the energy density is then obtained from a simple dot-product as

$$W = \frac{1}{2}\boldsymbol{\sigma} \cdot \boldsymbol{\epsilon}. \tag{28}$$

In the following, the volumetric strain energy densities for 3D and 2D plane strain will be derived.

Pure Dilatation Under Plane Strain Conditions

In the case of pure dilatation by an amount s under plane strain conditions, neither shear nor strain along the z-direction is present. The corresponding strain tensor in Voigt notation is

$$\boldsymbol{\epsilon} = \begin{bmatrix} s \\ s \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{29}$$

The plane-strain energy density is therefore

$$W_{el.}^{2D} = \frac{1}{2}\boldsymbol{\sigma} \cdot \boldsymbol{\epsilon} = \frac{9Ks^2}{4}, \tag{30}$$

where the fixed Poisson ratio $\nu = 1/3$, which is applicable to a 2D bond-based Peridynamic model, has been substituted.

Pure Dilatation in 3D

In the case of 3D pure dilatation no shear is present. Thus,

$$\boldsymbol{\epsilon} = \begin{bmatrix} s \\ s \\ s \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{31}$$

and the volumetric energy density is

$$W_{el.}^{3D} = \frac{9Ks^2}{2}, \tag{32}$$

Note that this result is independent of ν.

References

1. S.A. Silling, Reformulation of elasticity theory for discontinuities and long-range forces. J. Mech. Phys. Solids **48**, 175–209 (2000)
2. S.A. Silling, E. Askari, A meshfree method based on the peridynamic model of solid mechanics. Comput. Struct. **83**, 1526–1535 (2005)
3. S.A. Silling, M. Epton, O, Weckner, J. Xu, E. Askari, Peridynamic states and constitutive modeling. J. Elast. **88**, 151–184 (2007)
4. P. Seleson, M.L. Parks, On the role of the influence function in the peridynamic theory. Int. J. Multiscale Comput. Eng. **9**, 689–706 (2011)
5. D. Shepard, A two-dimensional interpolation function for irregularly-spaced data, in *Proceedings of the 1968 23rd ACM National Conference*, New York, 1968, pp. 517–524
6. P.W. Randles, L.D. Libersky, Smoothed particle hydrodynamics. Comput. Methods Appl. Mech. Eng. **139**, 375–408 (1996)
7. M.L. Parks, R.B. Lehoucq, S.J. Plimpton, S.A. Silling, Implementing peridynamics within a molecular dynamics code. Comput. Phys. Commun. **179**, 777–783 (2008)
8. F. Bobaru, Y.D. Ha, Adaptive refinement and multiscale modelling in 2d peridynamics. J. Multiscale Comput. Eng. **9**, 635–659 (2011)
9. F. Bobaru, M. Yang, L.F. Alves, S.A. Silling, E. Askari, J. Xu, Convergence, adaptive refinement, and scaling in 1D peridynamics. Int. J. Numer. Methods Eng. **77**, 852–877 (2009)
10. W. Hu, Y.D. Ha, F. Bobaru Numerical integration in peridynamics, Technical Report, Department of Engineering Mechanics, University of Nebraska-Lincoln, 2010
11. L.E. Kinsler et al., *Fundamentals of Acoustics*, 4th edn. (Wiley, New York, 2000)
12. S.A. Silling, O. Weckner, E. Askari, F. Bobaru, Crack nucleation in a peridynamic solid. Int. J. Fract. **162**, 219–227 (2010)
13. Y.D. Ha, F. Bobaru, Studies of dynamic crack propagation and crack branching with peridynamics. Int. J. Fract. **162**, 229–244 (2010)
14. Y.D. Ha, F. Bobaru, Characteristics of dynamic brittle fracture captured with peridynamics. Eng. Fract. Mech. **78**, 1156–116 (2011)
15. A. Agwai, I. Guven, E. Madenci, Predicting crack propagation with peridynamics: a comparative study. Int. J. Fract. **171**, 65–78 (2011)
16. R.D. Cook, W.C. Young, *Advanced Mechanics of Materials*, 2nd edn. (Prentice-Hall, Englewood Cliffs, 1999)
17. W. Voigt, *Lehrbuch der Kristallphysik: mit Ausschluß der Kristalloptik* (Teubner-Verlag, Leipzig, 1910)

Multiscale Partition of Unity

Patrick Henning, Philipp Morgenstern, and Daniel Peterseim

Abstract We introduce a new Partition of Unity Method for the numerical homogenization of elliptic partial differential equations with arbitrarily rough coefficients. We do not restrict to a particular ansatz space or the existence of a finite element mesh. The method modifies a given partition of unity such that optimal convergence is achieved independent of oscillation or discontinuities of the diffusion coefficient. The modification is based on an orthogonal decomposition of the solution space while preserving the partition of unity property. This precomputation involves the solution of independent problems on local subdomains of selectable size. We deduce quantitative error estimates for the method that account for the chosen amount of localization. Numerical experiments illustrate the high approximation properties even for 'cheap' parameter choices.

Keywords Partition of unity method • Multiscale method • LOD • Upscaling • Homogenization

1 Introduction

In this paper, we present a novel Multiscale Partition of Unity Method for reliable numerical homogenization in the meshfree context.

The Partition of Unity Method (PUM) was introduced by Babuška and Melenk in [6, 28], with the motivation that known singularities of the solution of a given PDE can be embedded into the ansatz space. Examples of Partition of Unity Methods can be found in [11, 15, 16, 20, 24, 30, 35]. Specific realizations of methods that fit

P. Henning
ANMC, Section de Mathématiques, École polytechnique fédérale de Lausanne, Lausanne, Switzerland
e-mail: patrick.henning@epfl.ch

P. Morgenstern (✉) • D. Peterseim
Institut für Numerische Simulation, Rheinische Friedrich-Wilhelms-Universität Bonn, Bonn, Germany
e-mail: morgenstern@ins.uni-bonn.de; peterseim@ins.uni-bonn.de

M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations VII*, Lecture Notes in Computational Science and Engineering 100,
DOI 10.1007/978-3-319-06898-5_10

into the general PUM framework but which are formulated in the context of finite element methods are the Extended Finite Element Method (XFEM, cf. [7, 29]), the Generalized Finite Element Method (GFEM, cf. [9, 10, 12, 23, 33, 34]) and the Stable GFEM presented in [17]. More general surveys on XFEM and GFEM can be found in [3, 13, 32].

In contrast to local singularities (usually due to the shape of the domain), multiscale problems consider the issue of very rough coefficients all over the domain. In order to obtain a reliable numerical approximation to the solution of the multiscale problem, it is typically necessary to 'resolve the coefficient', whereas a simple local averaging of the coefficient leads to wrong approximations. This means that the discrete solution space in which we seek an adequate Galerkin approximation must be able to fully capture the fine structures of the coefficient. Practically, this often leads to very large spaces and therefore to tremendous computational efforts. One approach to overcome this difficulty is to construct a special low dimensional space that incorporates the relevant fine scale features in its basis functions and that exhibits high approximation properties. A locally supported basis of this space can be computed in parallel by solving fine scale problems in small patches. This approach has been studied extensively for Finite Elements in [18, 19, 26, 27].

Other numerical multiscale methods can be found in [1, 4, 14, 21, 22, 25, 36]. In the context of meshfree methods we refer to recent papers [5, 31] where elliptic problems with rough coefficients are treated by introducing special non-polynomial shape functions, i.e., local eigenfunctions in [5] and rough polyharmonic splines in [31].

This paper aims to generalize the mesh-based approach of [18, 19, 26, 27] to general ansatz spaces without the requirement of underlying finite element meshes.

Throughout the paper, our model problem consists of finding a stationary heat distribution in some heterogenous media. Let $A \in L^\infty(\Omega, \mathbb{R}^{d\times d}_{sym})$ be a symmetric coefficient with uniform spectral bounds $\beta \geq \alpha > 0$ in some bounded Lipschitz domain $\Omega \subset \mathbb{R}^d$ for $d = 1, 2, 3$, i.e.,

$$0 < \alpha := \operatorname*{ess\,inf}_{x\in\Omega} \inf_{v\in\mathbb{R}^d\setminus\{0\}} \frac{\big(A(x)v, v\big)}{(v, v)},$$

$$\infty > \beta := \operatorname*{ess\,sup}_{x\in\Omega} \sup_{v\in\mathbb{R}^d\setminus\{0\}} \frac{\big(A(x)v, v\big)}{(v, v)}.$$

This coefficient A may be strongly heterogenous and arbitrarily rough. We consider the prototypical second-order linear elliptic PDE

$$-\operatorname{div} A\nabla u = g \tag{1a}$$

with homogeneous Neumann boundary condition

$$A\nabla u \cdot \nu = 0 \quad \text{on } \partial\Omega, \tag{1b}$$

given the exterior normal vector ν on $\partial\Omega$ and compatible right-hand side $g \in L^2(\Omega)$ such that $\int_\Omega g \, \mathrm{d}x = 0$.

We are looking for the unique (up to a constant) weak solution of problem (1a–1b). This is, for $V := H^1(\Omega)$, find $u \in V/\mathbb{R} = \{v \in V \mid \int_\Omega v \, \mathrm{d}x = 0\}$ with

$$a(u, \phi) := \int_\Omega A\nabla u \cdot \nabla \phi \, \mathrm{d}x = \int_\Omega g\phi \, \mathrm{d}x \quad \text{for all } \phi \in V/\mathbb{R}. \tag{2}$$

2 Abstract Multiscale Partition of Unity

In this section, we propose a Multiscale Partition of Unity Method without restriction to a particular ansatz space or even the existence of a mesh. This method is built upon two abstract (and possibly equal) partitions of unity that will be introduced in Sect. 2.1. Another crucial tool for the design of the method and its error analysis is a quasi-interpolation operator presented in Sect. 2.2. In the third and last subsection, we finally define the novel multiscale partition of unity method based on a localized orthogonal decomposition of V.

2.1 Two Partitions of Unity

The subsequent derivation of the multiscale method is based upon two standard partitions of unity. One partition is regular and spans a coarse space V_c. The other partition may be discontinuous and is solely used for the localization of the corrector problems in Sect. 2.3.

Definition 1 (Partitions of Unity).

(PU 1) Let $\mathcal{J}$ denote a finite index set and $\{\varphi_j \mid j \in \mathcal{J}\}$ a linearly independent Lipschitz partition of unity on Ω, i.e.

$$\sum_{j\in\mathcal{J}} \varphi_j = 1 \quad \text{with} \quad \forall\, j \in \mathcal{J}: \ 0 \le \varphi_j \in W^{1,\infty}(\Omega),$$

$$\text{s.t. for any } \lambda \in \mathbb{R}^{\mathcal{J}}, \ \sum_{j\in\mathcal{J}} \lambda_j \varphi_j = 0 \quad \Leftrightarrow \quad \forall\, j \in \mathcal{J}: \ \lambda_j = 0.$$

We define $\omega_j := \operatorname{supp}(\varphi_j)$ and $H_j := \operatorname{diam}(\omega_j)$ for all $j \in \mathcal{J}$ and $H := \max_{j\in\mathcal{J}} H_j$. The partition of unity functions span a finite dimensional coarse space $V_\mathrm{c} := \operatorname{span}\{\varphi_j \mid j \in \mathcal{J}\}$.

(PU 2) Let $\hat{\mathcal{J}}$ denote a finite index set and $\{\hat{\varphi}_{\hat{\jmath}} \mid \hat{\jmath} \in \hat{\mathcal{J}}\} \subseteq L^\infty(\Omega)$ a bounded and positive partition of unity on Ω, i.e.

$$\sum_{\hat{\jmath}\in\hat{\mathcal{J}}} \hat{\varphi}_{\hat{\jmath}} = 1 \quad \text{on } \Omega \quad \text{and} \quad \hat{\varphi}_{\hat{\jmath}} \geq 0.$$

We define $\hat{\omega}_{\hat{\jmath}} := \operatorname{supp}(\hat{\varphi}_{\hat{\jmath}})$ and $\hat{H}_{\hat{\jmath}} := \operatorname{diam}(\hat{\omega}_{\hat{\jmath}})$ for all $\hat{\jmath} \in \hat{\mathcal{J}}$. The maximum over all $\hat{H}_{\hat{\jmath}}$ is denoted by $\hat{H}$.

Example 1. The abstract definitions of (PU 1) and (PU 2) include the following special cases.

(a) (PU 2) equals (PU 1).
(b) Given some regular simplicial mesh $\mathcal{T}$ with vertices $\mathcal{N} = \mathcal{J}$, the partition (PU 1) is the continuous piecewise affine nodal basis functions φ_z, associated with vertices $z \in \mathcal{N}$. Recall that φ_z is defined by its values $\varphi_z(y) = \begin{cases} 1 & \text{if } y=z \\ 0 & \text{else} \end{cases}$ for vertices $y \in \mathcal{N}$. (PU 2) may be chosen as the characteristic (or 'indicator') functions of the triangles, i.e.

$$\hat{\mathcal{J}} = \mathcal{T} \quad \text{and} \quad \hat{\varphi}_T = \chi_T \text{ for all } T \in \mathcal{T}.$$

Definition 2 (Extension patch). For any patch ω_j in (PU 1) and $k \in \mathbb{N}$, we define the *k-th order extension patch* ω_j^k by

$$\omega_j^k := \bigcup_{x\in\omega_j} \overline{B_{k\cdot H}(x)} = \left\{ x \in \overline{\Omega} \mid \operatorname{dist}(x, \omega_j) \leq k \cdot H \right\}.$$

where $B_{k\cdot H}(x)$ denotes the ball with radius $k \cdot H$ around x and where "dist" denotes the set distance

$$\operatorname{dist}(x, B) := \inf_{b\in B} \|x - b\|.$$

For (PU 2), the extension patches $\hat{\omega}_{\hat{\jmath}}^k$, $k \in \mathbb{N}$ are defined analogously.

The subsequent definition serves only for the proofs. It has no practical relevance for the proposed method.

Definition 3 (Quasi-inclusion). Given two sets $B, C \subseteq \overline{\Omega}$, the set B is *n-quasi-included* in C (shorthand notation: $B \overset{n}{\underset{\sim}{\subseteq}} C$) if

$$\forall\, j_1, \ldots, j_m \in \mathcal{J},\ k_1, \ldots, k_m \in \mathbb{N}: \quad C \subseteq \bigcup_{i=1}^{m} \omega_{j_i}^{k_i} \quad \Rightarrow \quad B \subseteq \bigcup_{i=1}^{m} \omega_{j_i}^{k_i+n}.$$

Note that the shorthand notation allows for quantified transitivity

$$B \overset{n_1}{\underset{\sim}{\subseteq}} C \overset{n_2}{\underset{\sim}{\subseteq}} D \quad \Rightarrow \quad B \overset{n_1+n_2}{\underset{\sim}{\subseteq}} D.$$

2.2 Abstract Quasi-interpolation

Definition 4 (Quasi-interpolation operator). Throughout this paper, let $I : V \to V_c$ denote an abstract quasi-interpolation operator which fulfills the following properties.

(I1) I is linear and continuous.
(I2) $I|_{V_c} : V_c \to V_c$ is an isomorphism.
(I3) There exists a constant C_1 only depending on Ω and the shape of the patches ω_j such that for all $u \in H^1(\Omega)$ and all $j \in \mathcal{J}$

$$\|u - I(u)\|_{L^2(\omega_j)} \leq C_1 H_j \|\nabla u\|_{L^2(\omega_j^1)},$$

and a constant C_2 that further depends on $\max_{j\in\mathcal{J}}\big(H_j \left\|\varphi_j\right\|_{W^{1,\infty}(\Omega)}\big)$ such that

$$\|\nabla I(u)\|_{L^2(\omega_j)} \leq C_2 \|\nabla u\|_{L^2(\omega_j^1)}.$$

(I4) There exists a constant C_3 with same dependencies as C_2 and some $\kappa \in \mathbb{N}$ depending on the overlapping of the supports $\{\omega_j\}_{j\in\mathcal{J}}$ such that for all $v_c \in V_c$ there exists $v \in V$ such that

$$I(v) = v_c, \quad \|\nabla v\|_{L^2(\Omega)} \leq C_3 \|\nabla v_c\|_{L^2(\Omega)}, \text{ and } \operatorname{supp}(v) \overset{\kappa}{\subseteq} \operatorname{supp}(v_c),$$

with the quasi-inclusion $\overset{\kappa}{\subseteq}$ defined above.

A particular quasi-interpolation operator I is given in the subsequent definition.

Example 2 (Clement-type quasi-interpolation [8]). Define a weighted Clément-type quasi-interpolation operator

$$I : V \to V_c, \quad v \mapsto I(v) := \sum_{j\in\mathcal{J}} v_j \varphi_j \quad \text{with } v_j := \frac{(v, \varphi_j)_{L^2(\Omega)}}{(1, \varphi_j)_{L^2(\Omega)}}.$$

This operator obviously satisfies **(I1)** and **(I2)**. The properties **(I3)** have been shown in [8] in the abstract setting of (PU 1). We verify that **(I4)** is satisfied for a particular choice of basis functions. The following result is similar to [26, Lemma 2.1].

Lemma 1. *For a given regular triangulation $\mathcal{T}$ with vertices $\mathcal{N} = \mathcal{J}$ and nodal basis functions $\{\varphi_z\}_{z\in\mathcal{N}}$ as in Example 1(b), the quasi-interpolation operator from Example 2 satisfies **(I4)** with $\kappa = 1$.*

Proof. For any basis function φ_z, we want to find $b_z \in H^1(\Omega)$ with

$$I(b_z) = \varphi_z, \quad |\nabla b_z| \leq C|\nabla \varphi_z| \text{ a.e. in } \Omega \text{ and } \quad \operatorname{supp}(b_z) \subseteq \operatorname{supp}(\varphi_z).$$

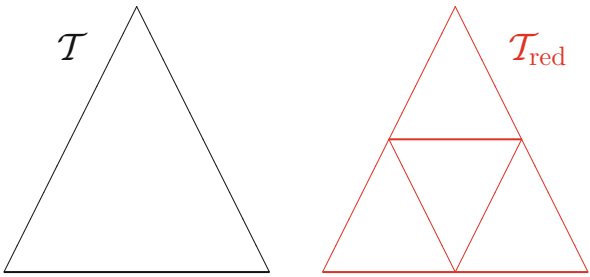


Fig. 1 Sketch of a so-called *red refinement* of a single triangle in $2d$. In general, the red refinement is based on the bisection of all edges and yields at least $d+1$ simplices of same shape and half diameter

Consider the red refinement $\mathcal{T}_{\mathrm{red}}$ of $\mathcal{T}$ (cf. Fig. 1) with nodal basis functions φ_z^{r}. If $\mathrm{nb}(z)$ denotes the set of all neighboring nodes of z in $\mathcal{T}_{\mathrm{red}}$ it can be verified that

$$b_z = (2^{d+1}-1)\varphi_z^{\mathrm{r}} - \tfrac{1}{2}\sum_{y\in\mathrm{nb}(z)} \varphi_y^{\mathrm{r}}$$

satisfies the desired conditions.

To conclude the proof, set

$$w := \sum_{z\in\mathcal{N}} (v_{\mathrm{c}}(z) - I(v_{\mathrm{c}})(z))\, b_z$$

and observe that $v := v_{\mathrm{c}} + w$ satisfies $v_{\mathrm{c}} = I(v)$, with $\mathrm{supp}(v) \overset{1}{\subsetneq} \mathrm{supp}(v_{\mathrm{c}})$ and $\|\nabla v\| \leq (1 + C + C_2 C)\, \|\nabla v_{\mathrm{c}}\|$. □

2.3 *Definition of the Method*

The goal is the construction of a space V_{c}^m that is of the same dimension as the discrete coarse space $V_{\mathrm{c}} = \mathrm{span}\{\varphi_j \mid j \in \mathcal{J}\}$ (cf. Definition 1) but which exhibits high H^1-approximations that are inherited from the L^2-approximation properties of V_{c}. Furthermore, we wish to explicitly construct a partition of unity basis for V_{c}^m.

Under the conditions **(I1)** and **(I2)** on the abstract quasi-interpolation operator, the space V can be written as the direct sum

$$V = V_{\mathrm{c}} \oplus V_{\mathrm{f}}, \quad \text{with } V_{\mathrm{f}} := \{v \in V \mid I(v) = 0\}. \tag{3}$$

The subspace V_{f} contains the fine scale features in V that cannot be captured by the coarse space V_{c}.

Definition 5 (Corrector). For $\hat{\jmath} \in \hat{\mathcal{J}}$ and $m \in \mathbb{N}$, define the *local corrector* $Q_{\hat{\jmath}}^m : V_{\mathrm{c}} \to V_{\mathrm{f}}(\hat{\omega}_{\hat{\jmath}}^m)$ as the mapping of a given $v_{\mathrm{c}} \in V_{\mathrm{c}}$ onto the solution $Q_{\hat{\jmath}}^m(v_{\mathrm{c}}) \in V_{\mathrm{f}}(\hat{\omega}_{\hat{\jmath}}^m) := \{v \in V_{\mathrm{f}} \mid v = 0 \text{ in } \Omega \setminus \hat{\omega}_{\hat{\jmath}}^m\}$ of

$$\int_{\hat{\omega}_{\hat{\jmath}}^m} A\nabla Q_{\hat{\jmath}}^m(v_c)\cdot\nabla w\,\mathrm{d}x = -\int_{\hat{\omega}_{\hat{\jmath}}} \hat{\varphi}_{\hat{\jmath}} A\nabla v_c\cdot\nabla w\,\mathrm{d}x \qquad \text{for all } w\in V_f(\hat{\omega}_{\hat{\jmath}}^m). \tag{4}$$

The global corrector is given by

$$Q^m(v_c) := \sum_{\hat{\jmath}\in\hat{\mathcal{J}}} Q_{\hat{\jmath}}^m(v_c).$$

For sufficiently large m such that $\hat{\omega}_{\hat{\jmath}}^m = \overline{\Omega}$ for all $\hat{\jmath}\in\hat{\mathcal{J}}$, we call $Q^\Omega := Q^m$ the *ideal corrector*.

The parameter m in Definition 5 reflects the locality of the method. The computational cost grows polynomially with m, while the error decays exponentially towards the error of the ideal (not localized) method.

Observe that the corrector problem (4) always yields a unique solution. Existence is clear by the Lax-Milgram theorem, because the zero function is the only constant function in V_f. For any $m\in\mathbb{N}$, the operator Q^m is linear and we denote the corrected discrete space

$$V_c^m := \{v_c + Q^m(v_c) \mid v_c\in V_c\},\quad V_c^\Omega := \{v_c + Q^\Omega(v_c)\mid v_c\in V_c\}. \tag{5}$$

Note that V_c^m (and also V_c^Ω) satisfies

$$V = V_c^m \oplus V_f$$

and that $\{\varphi_j + Q^m(\varphi_j)\mid j\in\mathcal{J}\}$ is a basis of V_c^m. Moreover, the ideal method comes with a-orthogonality of V_c onto V_f, i.e.

$$a(V_c^\Omega, V_f) = 0. \tag{6}$$

Remark 1. The partition of unity property is preserved under correction. To prove this, it suffices to show $\sum_{j\in\mathcal{J}} Q^m(\varphi_j) = 0$. We compute

$$\sum_{j\in\mathcal{J}} Q^m(\varphi_j) = \sum_{j\in\mathcal{J}}\sum_{\hat{\jmath}\in\hat{\mathcal{J}}} Q_{\hat{\jmath}}^m(\varphi_j) = \sum_{\hat{\jmath}\in\hat{\mathcal{J}}} Q_{\hat{\jmath}}^m\Big(\sum_{j\in\mathcal{J}}\varphi_j\Big) = \sum_{\hat{\jmath}\in\hat{\mathcal{J}}} Q_{\hat{\jmath}}^m(1) = 0.$$

Hence $\{\varphi_j + Q^m(\varphi_j)\mid j\in\mathcal{J}\}$ is a partition of unity. This also holds for the ideal corrector Q^Ω.

The Galerkin discretization of (2) with respect to the corrected space V_c^m, $m\in\mathbb{N}$ reads as follows.

Definition 6 (Multiscale Partition of Unity Method). Find $u_c^m\in V_c^m/\mathbb{R}$ such that

$$\int_\Omega A\nabla u_c^m\cdot\nabla v_c\,\mathrm{d}x = \int_\Omega g v_c\,\mathrm{d}x \quad \text{for all } v_c\in V_c^m/\mathbb{R}. \tag{7}$$

The *ideal problem* seeks $u_c^\Omega \in V_c^\Omega/\mathbb{R}$ such that

$$\int_\Omega A\nabla u_c^\Omega \cdot \nabla v_c \,\mathrm{d}x = \int_\Omega g v_c \,\mathrm{d}x \quad \text{for all } v_c \in V_c^\Omega/\mathbb{R}. \tag{8}$$

3 A Priori Error Analysis

In this section, we prove error estimates for the discrete solution of (7). In the first subsection, we consider the ideal case with ansatz space V_c^Ω (cf. (5)). The second subsection yields an error estimate for the localized problem. We will use the notation "$a \lesssim b$" to state the existence of $C > 0$ such that $a \leq Cb$. The hidden constant C may depend on the Poincaré constant $C_{\mathrm{Poinc}}(\Omega)$, on the ratio $\hat{H}/H$ and on the constants C_1, C_2, C_3 and κ from **(I1)**–**(I4)** in Definition 4, but *not* on the data A and g, the spectral bounds α and β (in particular the contrast $\frac{\beta}{\alpha}$) or the patch sizes H and $\hat{H}$.

3.1 Error Estimate for Global Basis Functions

We consider the ideal (but expensive) case of no localization (i.e. $\hat{\omega}_j^m = \overline{\Omega}$) and observe that the proposed method inherits the optimal approximation properties. This estimate is also important in the analysis of the localized method in Sect. 3.2.

Theorem 1 (A priori error estimate for the ideal case). *Let u be the solution of (2). Then the discrete solution u_c^Ω of (8) satisfies*

$$\alpha^{1/2}\big\|\nabla(u_c^\Omega - u)\big\|_{L^2(\Omega)} \leq \big\|A^{1/2}\nabla(u_c^\Omega - u)\big\|_{L^2(\Omega)} \lesssim \alpha^{-1/2} H \|g\|_{L^2(\Omega)}.$$

Proof. Observe that we can replace the test function space $V_c^\Omega/\mathbb{R}$ by V_c^Ω, since we subsequently only consider gradients. Galerkin orthogonality, i.e.

$$a\left(u - u_c^\Omega, v_c\right) = 0 \quad \text{for all } v_c \in V_c^\Omega \tag{9}$$

and (6) imply that $e := u - u_c^\Omega \in V_f$ and therefore $I(e) = 0$. We get

$$\begin{aligned}
\|A^{1/2}\nabla e\|_{L^2(\Omega)}^2 &= a(e,e) \\
&\overset{(9)}{=} a(e,u) = a(u,e) \\
&= \int_\Omega g(e - I(e)) \,\mathrm{d}x
\end{aligned}$$

$$\overset{\textbf{(I3)}}{\lesssim} H\|g\|_{L^2(\Omega)}\|\nabla e\|_{L^2(\Omega)}$$
$$\leq \alpha^{-1/2} H\|g\|_{L^2(\Omega)}\|A^{1/2}\nabla e\|_{L^2(\Omega)}.$$

□

3.2 Error Estimate for Local Basis Functions

In this final subsection, we give error estimates for the localized method. The main result is presented below.

Theorem 2 (A priori error estimates for the localized method). *Assume that $u \in V$ solves (2), then the discrete solution $u_{\mathrm{c}}^m \in V_{\mathrm{c}}^m$ of (7) satisfies*

$$\|\nabla u - \nabla u_{\mathrm{c}}^m\|_{L^2(\Omega)} \lesssim \alpha^{-1}\big(H + \tfrac{\beta}{\alpha} m^{d/2}\tilde{\theta}^m\big)\|g\|_{L^2(\Omega)},$$
$$\|u - u_{\mathrm{c}}^m\|_{L^2(\Omega)} \lesssim \alpha^{-2}\big(H + \tfrac{\beta}{\alpha} m^{d/2}\tilde{\theta}^m\big)^2\|g\|_{L^2(\Omega)},$$

with some generic constant $0 < \tilde{\theta} = \theta^{\lceil \hat{H}/H \rceil} < 1$ and θ depending on the contrast $\frac{\beta}{\alpha}$ (cf. Lemmas 4 and 5 below).

Proof. Let u_{c}^{Ω} be the solution of the ideal problem with correction operator Q^{Ω}, and $u_{\mathrm{c}} \in V_{\mathrm{c}}/\mathbb{R}$ such that $u_{\mathrm{c}}^{\Omega} = u_{\mathrm{c}} + Q^{\Omega}u_{\mathrm{c}}$. As a consequence of **(I3)**, all functions $v \in V_{\mathrm{f}}$ satisfy $\int_{\Omega} v\,\mathrm{d}x = 0$. With $Q^m v_{\mathrm{c}} \in V_{\mathrm{f}}$, we get

$$\begin{aligned}\|\nabla u - \nabla u_{\mathrm{c}}^m\|_{L^2(\Omega)} &\lesssim \min_{v_{\mathrm{c}}^m \in V_{\mathrm{c}}^m/\mathbb{R}} \big\|\nabla u - \nabla v_{\mathrm{c}}^m\big\|_{L^2(\Omega)} \\ &\leq \|\nabla u - \nabla(u_{\mathrm{c}} + Q^m u_{\mathrm{c}})\|_{L^2(\Omega)} \\ &\leq \big\|\nabla u - \nabla u_{\mathrm{c}}^{\Omega}\big\|_{L^2(\Omega)} + \big\|\nabla Q^{\Omega}u_{\mathrm{c}} - \nabla Q^m u_{\mathrm{c}}\big\|_{L^2(\Omega)}.\end{aligned}$$

Lemma 5 will quantify the localization error

$$\big\|\nabla Q^{\Omega}u_{\mathrm{c}} - \nabla Q^m u_{\mathrm{c}}\big\|_{L^2(\Omega)} \lesssim \tfrac{\beta}{\alpha} m^{d/2}\tilde{\theta}^m\Big(\sum_{\hat{\jmath}\in\hat{\mathcal{J}}} \|\nabla Q_{\hat{\jmath}}^{\Omega}u_{\mathrm{c}}\|_{L^2(\Omega)}^2\Big)^{1/2}.$$

This, Theorem 1 and the estimates

$$\sum_{\hat{\jmath}\in\hat{\mathcal{J}}} \|\nabla Q_{\hat{\jmath}}^{\Omega}u_{\mathrm{c}}\|_{L^2(\Omega)}^2 \overset{(4)}{\lesssim} \sum_{\hat{\jmath}\in\hat{\mathcal{J}}} \|\hat{\varphi}_{\hat{\jmath}}\nabla u_{\mathrm{c}}^{\Omega}\|_{L^2(\Omega)}^2 \leq \|\nabla u_{\mathrm{c}}^{\Omega}\|_{L^2(\Omega)}^2 \leq \alpha^{-1} C_{\mathrm{Poinc}}(\Omega)\,\|g\|_{L^2(\Omega)}^2$$

yield the H^1-error estimate. The L^2-error estimate is obtained by a standard Aubin-Nitsche argument. □

To prove Lemma 5, several tools are needed in addition to the preceding results. They will be discussed below.

Lemma 2 (Quasi-inclusion of intersecting patches). *Let $i, j \in \mathcal{J}$ and $\ell, k, m \in \mathbb{N}$ with $k \geq \ell \geq 2$. Then*

$$\text{if} \quad \omega_i^m \cap \left(\omega_j^k \setminus \omega_j^\ell\right) \neq \emptyset \quad \text{then} \quad \omega_i \subseteq \omega_j^{k+m+1} \setminus \omega_j^{\ell-m-1}.$$

Proof. Consider $x \in \omega_i^m \cap \left(\omega_j^k \setminus \omega_j^\ell\right)$ and observe

$$\omega_i \;\subseteq\; \overline{B_{(m+1)H}(x)} \;\subseteq\; \omega_j^{k+m+1} \setminus \omega_j^{\ell-m-1}.$$

□

Definition 7 (Cut-off functions). For all $j \in \mathcal{J}$ and $\ell, k \in \mathbb{N}$ with $k > \ell$, we define the *cut-off function*

$$\eta_j^{k,\ell}(x) = \frac{\operatorname{dist}(x, \omega_j^{k-\ell})}{\operatorname{dist}(x, \omega_j^{k-\ell}) + \operatorname{dist}(x, \Omega \setminus \omega_j^k)}.$$

For $\Omega \setminus \omega_j^k = \emptyset$, we set $\eta_j^{k,\ell} \equiv 0$. Note that $\eta_j^{k,\ell} = 0$ in $\omega_j^{k-\ell}$ and $\eta_j^{k,\ell} = 1$ in $\Omega \setminus \omega_j^k$. Moreover, $\eta_j^{k,\ell}$ is bounded between 0 and 1 and Lipschitz continuous with

$$\left\|\nabla \eta_j^{k,\ell}\right\|_{L^\infty(\Omega)} \leq \frac{1}{\ell \cdot H}. \tag{10}$$

See [2, Theorem 8.5] for existence and boundedness of the weak derivative of Lipschitz-continuous functions.

Remark 2. The Lipschitz bound is shown as follows. For $x \in \mathbb{R}^d$ we have the triangle inequality

$$\operatorname{dist}(x, \omega_j^{k-\ell}) + \operatorname{dist}(x, \Omega \setminus \omega_j^k) \;\geq\; \operatorname{dist}(\omega_j^{k-\ell}, \Omega \setminus \omega_j^k) \;=\; \ell \cdot H.$$

Moreover, any nonemtpy set B in a metric space satisfies Lipschitz continuity of the distance function $\operatorname{dist}(\cdot, B)$ in the sense

$$|\operatorname{dist}(x, B) - \operatorname{dist}(y, B)| \;\leq\; \operatorname{dist}(x, y) \quad \text{for } x, y \in \mathbb{R}^d.$$

Altogether,

$$\begin{aligned}\frac{\left|\eta_j^{k,\ell}(x) - \eta_j^{k,\ell}(y)\right|}{\operatorname{dist}(x, y)} &\leq \frac{1}{\operatorname{dist}(x, y)} \cdot \frac{\left|\operatorname{dist}(x, \omega_j^{k-\ell}) - \operatorname{dist}(y, \omega_j^{k-\ell})\right|}{\ell \cdot H} \\ &\leq \frac{1}{\ell \cdot H}.\end{aligned}$$

A technical issue in our error analysis is that V_f is not invariant under multiplication by such cut-off functions. However, the product $\eta_j^{k,\ell} w$ for $w \in V_\mathrm{f}$ is close to V_f in the following sense.

Lemma 3 (Quasi-invariance of V_f under multiplication by cut-off functions). *Recall κ from **(I4)**. For any given $w \in V_\mathrm{f}$ and cutoff function $\eta_j^{k,\ell}$ with $k > \ell > 0$, there exists $\tilde{w} \in V_\mathrm{f}(\Omega \setminus \omega_j^{k-\ell-\kappa-2}) \subseteq V_\mathrm{f}$ such that*

$$\|\nabla(\eta_j^{k,\ell} w - \tilde{w})\|_{L^2(\Omega)} \lesssim \ell^{-1} \|\nabla w\|_{L^2(\omega_j^{k+2} \setminus \omega_j^{k-\ell-2})}.$$

Proof. We fix the $j \in \mathcal{J}$ and $k \in \mathbb{N}$ and denote $\eta_\ell := \eta_j^{k,\ell}$ and $c_i^\ell := \frac{1}{|\omega_i^1|} \int_{\omega_i^1} \eta_\ell \, \mathrm{d}x$ for $i \in \mathcal{J}$. The property **(I4)**, applied to $I(\eta_\ell w) \in V_\mathrm{f}$, yields $v \in V$ with

$$I(v) = I(\eta_\ell w), \ \|\nabla v\|_{L^2(\Omega)} \lesssim \|\nabla I \eta_\ell w\|_{L^2(\Omega)}, \tag{11}$$

$$\text{and} \quad \operatorname{supp}(v) \overset{\kappa}{\subseteq} \operatorname{supp}\big(I(\eta_\ell w)\big) \overset{1}{\subseteq} \operatorname{supp}(\eta_\ell w) \subseteq \Omega \setminus \omega_j^{k-\ell},$$

$$\text{which yields} \quad \operatorname{supp}(v) \overset{\kappa+1}{\subseteq} \Omega \setminus \omega_j^{k-\ell} \Rightarrow \operatorname{supp}(v) \subseteq \Omega \setminus \omega_j^{k-\ell-\kappa-2}. \tag{12}$$

Note that $\operatorname{supp}\big(I(\eta_\ell w)\big) \overset{1}{\subseteq} \operatorname{supp}(\eta_\ell w)$ is a consequence of **(I3)**, and that (11) implies $I(v - \eta_\ell w) = 0$.

We define $\tilde{w} := \eta_\ell w - v \in V_\mathrm{f}(\Omega \setminus \omega_j^{k-\ell-\kappa-2})$. Using $I(w) = 0$, we obtain for any $i \in \mathcal{J}$

$$\|\nabla I(\eta_\ell w)\|_{L^2(\omega_i)} \overset{\textbf{(I1)}}{=} \big\|\nabla I((\eta_\ell - c_i^\ell) w)\big\|_{L^2(\omega_i)} \overset{\textbf{(I3)}}{\lesssim} \big\|\nabla((\eta_\ell - c_i^\ell) w)\big\|_{L^2(\omega_i^1)}. \tag{13}$$

This gives us

$$\begin{aligned}
\|\nabla I(\eta_\ell w)\|_{L^2(\Omega)}^2 &\lesssim \sum_{i \in \mathcal{J}} \|\nabla I(\eta_\ell w)\|_{L^2(\omega_i)}^2 \\
&\overset{(13)}{\lesssim} \sum_{i \in \mathcal{J}} \big\|\nabla\big((\eta_\ell - c_i^\ell) w\big)\big\|_{L^2(\omega_i^1)}^2 \\
&= \sum_{\substack{i \in \mathcal{J}: \\ \omega_i^1 \cap (\omega_j^k \setminus \omega_j^{k-\ell}) \neq \emptyset}} \big\|\nabla\big((\eta_\ell - c_i^\ell) w\big)\big\|_{L^2(\omega_i^1)}^2 \\
&\overset{(2)}{\lesssim} \sum_{\substack{i \in \mathcal{J}: \\ \omega_i \subseteq \omega_j^{k+2} \setminus \omega_j^{k-\ell-2}}} \big\|\nabla\big((\eta_\ell - c_i^\ell) w\big)\big\|_{L^2(\omega_i^1)}^2 \\
&\lesssim \sum_{\substack{i \in \mathcal{J}: \\ \omega_i \subseteq \omega_j^{k+2} \setminus \omega_j^{k-\ell-2}}} \|(\nabla \eta_\ell)(w - I w)\|_{L^2(\omega_i^1)}^2 + \big\|(\eta_\ell - c_i^\ell) \nabla w\big\|_{L^2(\omega_i^1)}^2.
\end{aligned}$$

Since $\nabla\eta_\ell \neq 0$ only in $\omega_j^k \setminus \omega_j^{k-\ell}$ and $(\eta_\ell - c_i^\ell)\big|_{\omega_i^1} \neq 0$ only if ω_i^1 intersects with $\omega_j^k \setminus \omega_j^{k-\ell}$, we have

$$\begin{aligned}
&\lesssim \sum_{\substack{i\in\mathcal{J}:\\ \omega_i\subseteq\omega_j^{k+1}\setminus\omega_j^{k-\ell-1}}} \|(\nabla\eta_\ell)(w - Iw)\|^2_{L^2(\omega_i)} + \sum_{\substack{i\in\mathcal{J}:\\ \omega_i\subseteq\omega_j^{k+1}\setminus\omega_j^{k-\ell-1}}} \left\|\left(\eta_\ell - c_i^\ell\right)\nabla w\right\|^2_{L^2(\omega_i^1)} \\
&\lesssim H^2 \|\nabla\eta_\ell\|^2_{L^\infty(\Omega)} \|\nabla w\|^2_{L^2(\omega_j^{k+1}\setminus\omega_j^{k-\ell-1})} \\
&\qquad\qquad + \sum_{\substack{i\in\mathcal{J}:\\ \omega_i\subseteq\omega_j^{k+1}\setminus\omega_j^{k-\ell-1}}} \left\|\left(\eta_\ell - c_i^\ell\right)\nabla w\right\|^2_{L^2(\omega_i^1)} \\
&\overset{(10)}{\leq} \ell^{-2} \|\nabla w\|^2_{L^2(\omega_j^{k+2}\setminus\omega_j^{k-\ell-2})}, \qquad (14)
\end{aligned}$$

where we used the Lipschitz bound $\|\eta_\ell - c_i^\ell\|_{L^\infty(\omega_i^1)} \lesssim H \|\nabla\eta_\ell\|_{L^\infty(\omega_i^1)}$. The combination of (11) and (14) readily yields the assertion,

$$\begin{aligned}
\|\nabla(\eta_\ell w - \tilde{w})\|^2_{L^2(\Omega)} = \|\nabla v\|^2_{L^2(\Omega)} &\overset{(11)}{\leq} \|\nabla I(\eta_\ell w)\|^2_{L^2(\Omega)} \\
&\overset{(14)}{\lesssim} \ell^{-2} \|\nabla w\|^2_{L^2(\omega_j^{k+2}\setminus\omega_j^{k-\ell-2})}.
\end{aligned}$$

□

A key result is the following.

Lemma 4 (Exponential decay in the fine scale space). *Consider some fixed $j \in \mathcal{J}$ and let $F \in (V_\mathrm{f})'$ satisfy $F(w) = 0$ for all $w \in V_\mathrm{f}(\Omega \setminus \omega_j^\varrho)$ with $\varrho := \lceil \frac{\hat{H}}{H} \rceil$. Let $p \in V_\mathrm{f}$ be the solution of*

$$a(p, w) = F(w) \qquad \text{for all } w \in V_\mathrm{f}. \qquad (15)$$

Then there exists $0 < \theta < 1$ depending on the contrast $\frac{\beta}{\alpha}$ such that for all positive $k \in \mathbb{N}$ it holds

$$\|\nabla p\|_{L^2(\Omega\setminus\omega_j^k)} \lesssim \theta^k \|\nabla p\|_{L^2(\Omega)}.$$

Proof. We use a cut-off function as in the previous proof and denote $\eta_\ell := \eta_j^{k,\ell}$ with $\ell \leq k - \varrho - \kappa - 2$.

Applying Lemma 3 yields the existence of $\tilde{p} \in V_\mathrm{f}(\Omega \setminus \omega_j^{k-\ell-\kappa-2})$ with the estimate $\|\nabla(\eta_\ell p - \tilde{p})\|_{L^2(\Omega)} \lesssim \ell^{-1} \|\nabla p\|_{L^2(\omega_j^{k+2}\setminus\omega_j^{k-\ell-2})}$. Due to the property $\tilde{p} \in V_\mathrm{f}(\Omega \setminus \omega_j^{k-\ell-\kappa-2})$ and the assumptions on F we also have

$$\int_{\Omega\setminus\omega_j^{k-\ell-\kappa-2}} A\nabla p\cdot\nabla\tilde{p}\,\mathrm{d}x = \int_{\Omega} A\nabla p\cdot\nabla\tilde{p}\,\mathrm{d}x = F(\tilde{p}) = 0. \tag{16}$$

This leads to

$$\begin{aligned}\alpha\,\|\nabla p\|^2_{L^2(\Omega\setminus\omega_j^k)} &\le \int_{\Omega\setminus\omega_j^k} A\nabla p\cdot\nabla p\,\mathrm{d}x \le \int_{\Omega\setminus\omega_j^{k-\ell-\kappa-2}} \eta_\ell A\nabla p\cdot\nabla p\,\mathrm{d}x\\ &= \int_{\Omega\setminus\omega_j^{k-\ell-\kappa-2}} A\nabla p\cdot(\nabla(\eta_\ell p) - p\nabla\eta_\ell)\,\mathrm{d}x.\end{aligned}$$

With (16) and since $p\in V_{\mathrm{f}}$, this is

$$\begin{aligned}&= \int_{\Omega\setminus\omega_j^{k-\ell-\kappa-2}} A\nabla p\cdot\big(\nabla(\eta_\ell p-\tilde{p}) - (p-I(p))\nabla\eta_\ell\big)\,\mathrm{d}x\\ &\lesssim \ell^{-1}\beta\Big(\|\nabla p\|^2_{L^2(\Omega\setminus\omega_j^{k-\ell-\kappa-2})}\\ &\qquad + H^{-1}\|\nabla p\|_{L^2(\Omega\setminus\omega_j^{k-\ell-\kappa-2})}\|p-I(p)\|_{L^2(\Omega\setminus\omega_j^{k-\ell-\kappa-2})}\Big)\\ &\overset{(13)}{\lesssim} \ell^{-1}\beta\,\|\nabla p\|^2_{L^2(\Omega\setminus\omega_j^{k-\ell-\kappa-2})}.\end{aligned}$$

Hence, there exists a constant C independent of mesh size, contrast, number of patch extension layers, such that

$$\|\nabla p\|^2_{L^2(\Omega\setminus\omega_j^k)} \le C\,\ell^{-1}\frac{\beta}{\alpha}\,\|\nabla p\|^2_{L^2(\Omega\setminus\omega_j^{k-\ell-\kappa-2})}. \tag{17}$$

Choose $\ell := \lceil eC\frac{\beta}{\alpha}\rceil$ and observe that successive use of (17) yields

$$\begin{aligned}\|\nabla p\|^2_{L^2(\Omega\setminus\omega_j^k)} &\le e^{-1}\,\|\nabla p\|^2_{L^2(\Omega\setminus\omega_j^{k-\ell-\kappa-2})}\\ &\le e^{-\lfloor\frac{k-\varrho}{\ell+\kappa+2}\rfloor}\,\|\nabla p\|^2_{L^2(\Omega\setminus\omega_j^{\varrho})}\\ &\lesssim e^{-\frac{k}{\ell+\kappa+2}}\,\|\nabla p\|^2_{L^2(\Omega)}.\end{aligned}$$

The choice $\theta := e^{-(\lceil eC\beta/\alpha\rceil+\kappa+2)^{-1}}$ concludes the proof. □

Lemma 5 (Localization error). *For $u_{\mathrm{c}}\in V_{\mathrm{c}}$, the correction operators Q^m and Q^Ω satisfy*

$$\big\|\nabla\big(Q^\Omega u_{\mathrm{c}} - Q^m u_{\mathrm{c}}\big)\big\|_{L^2(\Omega)} \lesssim \tfrac{\beta}{\alpha} m^{d/2}\tilde{\theta}^m\,\big\|Q^\Omega u_{\mathrm{c}}^m\big\|_{L^2(\Omega)}$$

with $\tilde{\theta} := \theta^{\lceil \hat{H}/H\rceil} < 1$ and θ from Lemma 4.

Proof. Recall the definition $Q^m u_c := \sum_{\hat{\jmath}\in\hat{\mathcal{J}}} Q^m_{\hat{\jmath}}(u_c)$ with

$$\int_{\hat{\omega}^m_{\hat{\jmath}}} A\nabla Q^m_{\hat{\jmath}}(u_c)\cdot\nabla w\,\mathrm{d}x = \underbrace{-\int_\Omega \hat{\varphi}_{\hat{\jmath}} A\nabla u_c\cdot\nabla w\,\mathrm{d}x}_{F_{\hat{\jmath}}(w)} \quad \text{for all } w\in V_f(\hat{\omega}^m_{\hat{\jmath}}),\ \ \hat{\jmath}\in\hat{\mathcal{J}}.$$

Note that the right-hand side $F_{\hat{\jmath}}$ of the local problem is zero for $w \in V_f(\Omega\setminus\hat{\omega}_{\hat{\jmath}})$. Consider some fixed $\hat{\jmath}\in\hat{\mathcal{J}}$ and choose $j\in\mathcal{J}$ such that $\omega_j\cap\hat{\omega}_{\hat{\jmath}}\neq\emptyset$. Recall $\varrho = \lceil \frac{\hat{H}}{H}\rceil$, then we have $\hat{\omega}_{\hat{\jmath}}\subseteq\omega_j^\varrho$ and thus $V_f(\Omega\setminus\omega_j^\varrho)\subseteq V_f(\Omega\setminus\hat{\omega}_{\hat{\jmath}})$. Hence $F_{\hat{\jmath}}$ satisfies the conditions from Lemma 4.

Moreover, we get $\omega_j^k\subseteq\hat{\omega}_j^m$ for k satisfying

$$m = \lceil \tfrac{k\cdot H}{\hat{H}}\rceil \le k\lceil\tfrac{H}{\hat{H}}\rceil. \tag{18}$$

Denote $v := Q^\Omega u_c - Q^m u_c \in V_f$ and note that $I(v)=0$. Using the cut-off functions $\eta_j^{k,1}$ from Definition 7, we obtain

$$\alpha\left\|\nabla v\right\|^2_{L^2(\Omega)} \le \sum_{\hat{\jmath}\in\hat{\mathcal{J}}}\Big(\underbrace{\left(A\nabla\left(Q^\Omega_{\hat{\jmath}}u_c - Q^m_{\hat{\jmath}}u_c\right),\nabla(v(1-\eta_j^{k,1}))\right)_{L^2(\Omega)}}_{\mathrm{I}} + \underbrace{\left(A\nabla\left(Q^\Omega_{\hat{\jmath}}u_c - Q^m_{\hat{\jmath}}u_c\right),\nabla(v\eta_j^{k,1})\right)_{L^2(\Omega)}}_{\mathrm{II}}\Big).$$

We bound the term I by

$$\begin{aligned}
\mathrm{I} &\le \beta\left\|\nabla\left(Q^\Omega_{\hat{\jmath}}u_c - Q^m_{\hat{\jmath}}u_c\right)\right\|_{L^2(\Omega)}\left\|\nabla\big(v(1-\eta_j^{k,1})\big)\right\|_{L^2(\omega_j^k)}\\
&\le \beta\left\|\nabla\left(Q^\Omega_{\hat{\jmath}}u_c - Q^m_{\hat{\jmath}}u_c\right)\right\|_{L^2(\Omega)}\left(\left\|\nabla v\right\|_{L^2(\omega_j^k)} + \left\|v\nabla\big(1-\eta_j^{k,1}\big)\right\|_{L^2(\omega_j^k\setminus\omega_j^{k-1})}\right)\\
&\lesssim \beta\left\|\nabla\left(Q^\Omega_{\hat{\jmath}}u_c - Q^m_{\hat{\jmath}}u_c\right)\right\|_{L^2(\Omega)}\left(\left\|\nabla v\right\|_{L^2(\omega_j^k)} + H^{-1}\left\|v-I(v)\right\|_{L^2(\omega_j^k\setminus\omega_j^{k-1})}\right)\\
&\lesssim \beta\left\|\nabla\left(Q^\Omega_{\hat{\jmath}}u_c - Q^m_{\hat{\jmath}}u_c\right)\right\|_{L^2(\Omega)}\left\|\nabla v\right\|_{L^2(\omega_j^{k+1})}.
\end{aligned}$$

Lemma 3 yields the existence of $\tilde{v}\in V_f(\Omega\setminus\omega_j^{k-\kappa-3})$ with

$$\left\|\nabla(v\eta_j^{k,1}-\tilde{v})\right\|_{L^2(\Omega)} \lesssim \left\|\nabla v\right\|_{L^2(\omega_j^{k+2})}.$$

We assume that m is large enough such that $k\ge\varrho+\kappa+3$, then $\tilde{v}\in V_f(\Omega\setminus\hat{\omega}_{\hat{\jmath}})$ and hence

$$\int_\Omega A\nabla\left(Q^\Omega_{\hat{\jmath}}u_c - Q^m_{\hat{\jmath}}u_c\right)\cdot\nabla\tilde{v}\,\mathrm{d}x = 0.$$

It follows that

$$\begin{aligned}\mathrm{II} &= \left(A\nabla\left(Q_{\hat{\jmath}}^{\Omega}u_{\mathrm{c}} - Q_{\hat{\jmath}}^{m}u_{\mathrm{c}}\right), \nabla(v\eta_j^{k,1} - \tilde{v})\right)_{L^2(\Omega)} \\ &\lesssim \beta \left\|\nabla\left(Q_{\hat{\jmath}}^{\Omega}u_{\mathrm{c}} - Q_{\hat{\jmath}}^{m}u_{\mathrm{c}}\right)\right\|_{L^2(\Omega)} \|\nabla v\|_{L^2(\omega_j^{k+2})}.\end{aligned}$$

Combining the estimates for I and II finally yields

$$\begin{aligned}\|\nabla v\|_{L^2(\Omega)}^2 &\lesssim \frac{\beta}{\alpha}\sum_{\hat{\jmath}\in\hat{\mathcal{J}}}\left\|\nabla\left(Q_{\hat{\jmath}}^{\Omega}u_{\mathrm{c}} - Q_{\hat{\jmath}}^{m}u_{\mathrm{c}}\right)\right\|_{L^2(\Omega)} \|\nabla v\|_{L^2(\omega_j^{k+2})} \\ &\lesssim \frac{\beta}{\alpha}\, k^{d/2}\Big(\sum_{\hat{\jmath}\in\hat{\mathcal{J}}}\left\|\nabla\left(Q_{\hat{\jmath}}^{\Omega}u_{\mathrm{c}} - Q_{\hat{\jmath}}^{m}u_{\mathrm{c}}\right)\right\|_{L^2(\Omega)}^2\Big)^{1/2} \|\nabla v\|_{L^2(\Omega)},\end{aligned} \tag{19}$$

provided that $\left|\{i \in \mathcal{J} \mid \omega_i \subseteq \omega_j^{k+2}\}\right| \lesssim k^{d/2}$.

In order to bound $\|\nabla(Q_{\hat{\jmath}}^{\Omega}u_{\mathrm{c}} - Q_{\hat{\jmath}}^{m}u_{\mathrm{c}})\|_{L^2(\Omega)}^2$, we use Galerkin orthogonality for the local problems, which is

$$\left\|\nabla\left(Q_{\hat{\jmath}}^{\Omega}u_{\mathrm{c}} - Q_{\hat{\jmath}}^{m}u_{\mathrm{c}}\right)\right\|_{L^2(\Omega)}^2 \lesssim \inf_{q\in V_{\mathrm{f}}(\omega_j^k)}\left\|\nabla(Q_{\hat{\jmath}}^{\Omega}u_{\mathrm{c}} - q)\right\|_{L^2(\Omega)}^2. \tag{20}$$

(I4) yields the existence of $\tilde{w} \in V_{\mathrm{f}}$ such that

$$I(\tilde{w}) = I((1-\eta_j^{k,1})Q_{\hat{\jmath}}^{\Omega}u_{\mathrm{c}}), \quad \|\nabla\tilde{w}\|_{L^2(\Omega)} \lesssim \|\nabla I((1-\eta_j^{k,1})Q_{\hat{\jmath}}^{\Omega}u_{\mathrm{c}})\|_{L^2(\Omega)},$$
$$\text{and} \quad \operatorname{supp}(\tilde{w}) \overset{\kappa}{\subseteq} \operatorname{supp}((1-\eta_j^{k,1})Q_{\hat{\jmath}}^{\Omega}u_{\mathrm{c}}) \subseteq \omega_j^k.$$

We observe

$$\|\nabla I((1-\eta_j^{k,1})Q_{\hat{\jmath}}^{\Omega}u_{\mathrm{c}})\|_{L^2(\omega_j^{k+\kappa})}^2 = \|\nabla I((1-\eta_j^{k,1})Q_{\hat{\jmath}}^{\Omega}u_{\mathrm{c}})\|_{L^2(\omega_j^{k+1}\setminus\omega_j^{k-2})}^2. \tag{21}$$

With $p_{\hat{\jmath}} := (1-\eta_j^{k,1})Q_{\hat{\jmath}}^{\Omega}u_{\mathrm{c}} - \tilde{w} \in V_{\mathrm{f}}(\omega_j^{k+\kappa})$, we obtain

$$\begin{aligned}\left\|\nabla\left(Q_{\hat{\jmath}}^{\Omega}u_{\mathrm{c}} - Q_{\hat{\jmath}}^{m}u_{\mathrm{c}}\right)\right\|_{L^2(\Omega)}^2 &\overset{(20)}{\lesssim} \left\|\nabla(\eta_j^{k,1}Q_{\hat{\jmath}}^{\Omega}u_{\mathrm{c}} + (1-\eta_j^{k,1})Q_{\hat{\jmath}}^{\Omega}u_{\mathrm{c}} - p_{\hat{\jmath}})\right\|_{L^2(\Omega)}^2 \\ &= \left\|\nabla(\eta_j^{k,1}Q_{\hat{\jmath}}^{\Omega}u_{\mathrm{c}} - \tilde{w})\right\|_{L^2(\Omega)}^2 \\ &\lesssim \|\nabla Q_{\hat{\jmath}}^{\Omega}u_{\mathrm{c}}\|_{L^2(\Omega\setminus\omega_j^{k-2})}^2 + \|\nabla\tilde{w}\|_{L^2(\omega_j^{k+\kappa})}^2 \\ &\lesssim \|\nabla Q_{\hat{\jmath}}^{\Omega}u_{\mathrm{c}}\|_{L^2(\Omega\setminus\omega_j^{k-2})}^2 \\ &\qquad + \|\nabla I((1-\eta_j^{k,1})Q_{\hat{\jmath}}^{\Omega}u_{\mathrm{c}})\|_{L^2(\omega_j^{k+\kappa})}^2\end{aligned}$$

$$
\begin{aligned}
&\overset{(21)}{\lesssim} \|\nabla Q_{\hat{\jmath}}^{\Omega} u_{\mathrm{c}}\|^2_{L^2(\Omega\setminus\omega_j^{k-2})} \\
&\qquad + \|\nabla I((1-\eta_j^{k,1}) Q_{\hat{\jmath}}^{\Omega} u_{\mathrm{c}})\|^2_{L^2(\omega_j^{k+1}\setminus\omega_j^{k-2})} \\
&\overset{\mathbf{(I3)}}{\lesssim} \|\nabla Q_{\hat{\jmath}}^{\Omega} u_{\mathrm{c}}\|^2_{L^2(\Omega\setminus\omega_j^{k-3})} \\
&\overset{\text{Lemma 4}}{\lesssim} \theta^{2(k-3)} \left\|\nabla Q_{\hat{\jmath}}^{\Omega} u_{\mathrm{c}}\right\|^2_{L^2(\Omega)} \\
&\overset{(18)}{\lesssim} \tilde{\theta}^{2m} \left\|\nabla Q_{\hat{\jmath}}^{\Omega} u_{\mathrm{c}}\right\|^2_{L^2(\Omega)}.
\end{aligned} \tag{22}
$$

Combining (19) and (22) proves the lemma. □

4 Numerical Experiment

In this section, we present numerical results for a special realization of the Multiscale Partition of Unity Method. We consider a "coarse" regular triangulation $\mathcal{T}_H$ of Ω, where H denotes the maximum diameter of an element of $\mathcal{T}_H$. By $\mathcal{N}_H$ we denote the set of vertices of the triangulation. We choose the basis functions φ_z as in Example 1(b), i.e., the continuous and piecewise affine nodal basis functions associated with vertices $z \in \mathcal{N} = \mathcal{J}$. The second partition of unity (PU 2) is given by the indicator functions of the elements of the triangulation, i.e. $\{\hat{\varphi}_{\hat{\jmath}} \mid \hat{\jmath} \in \hat{\mathcal{J}}\} := \{\chi_T \mid T \in \mathcal{T}_H\}$. The corrector problems given by (4) are solved with a P_1 Finite Element method on a fine grid with resolution $h = 2^{-8}$. The reference solution u_h is therefore the P_1 Finite Element approximation in a space with mesh size $h = 2^{-8}$.

In order to estimate the accuracy of u_h itself, we performed a second computation for the mesh size $h = 2^{-10}$. The relative L^2-error between the Finite Element approximation on a uniform mesh with resolution $h = 2^{-8}$ and the Finite Element approximation on a uniform mesh with resolution $h = 2^{-10}$ is 0.023. The relative H^1-error is 0.3204. However, we only compute the errors of u_{c}^m with respect to the reference solution (i.e. for $h = 2^{-8}$), since this is the relevant error for investigating the effect of the coarse grid resolution and the decay of the multiscale basis functions on u_{c}^m.

The extension patches $\hat{\omega}_{\hat{\jmath}}^m$ can be defined by using the structure of the coarse grid by setting

$$
\begin{aligned}
\hat{\omega}_{\hat{\jmath}}^0 &:= T_j \in \mathcal{T}_H, \\
\hat{\omega}_{\hat{\jmath}}^m &:= \cup\{T \in \mathcal{T}_H \mid T \cap \hat{\omega}_{\hat{\jmath}}^{m-1} \neq \emptyset\} \quad m = 1, 2, \ldots.
\end{aligned} \tag{23}
$$

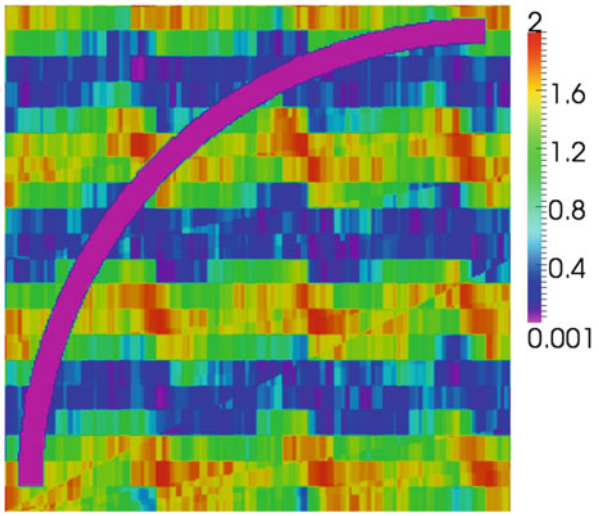


Fig. 2 Plot of the rapidly varying and highly heterogeneous diffusion coefficient a_ε given by Eq. (25), which takes values between 0.01 and 2. The structure is disturbed by an isolating arc (*purple*) of thickness 0.05 and with conductivity 10^{-3}

We consider the following model problem. Let $\Omega :=]0,1[^2$ and $\varepsilon := 0.05$. Find $u_\varepsilon \in V$ with

$$\begin{aligned} -\operatorname{div}\big(a_\varepsilon(x)\nabla u_\varepsilon(x)\big) &= x_1 - \tfrac{1}{2} \qquad \text{in } \Omega \\ \nabla u_\varepsilon(x)\cdot \nu &= 0 \qquad \text{on } \partial\Omega. \end{aligned} \tag{24}$$

The scalar diffusion coefficient a_ε in Eq. (24) is depicted in Fig. 2. It has a contrast of order 10^3 and is constructed from the highly heterogeneous distribution

$$c_\varepsilon(x_1,x_2) := 1 + \tfrac{1}{10}\sum_{j=0}^{4}\sum_{i=0}^{j}\left(\tfrac{2}{j+1}\cos\left(\left\lfloor i x_2 - \tfrac{x_1}{1+i}\right\rfloor + \left\lfloor \tfrac{i x_1}{\varepsilon}\right\rfloor + \left\lfloor \tfrac{x_2}{\varepsilon}\right\rfloor\right)\right)$$

and an isolating arc of radius $r := 0.9$, thickness $\frac{\varepsilon}{2}$ and center $c_0 := (1-\varepsilon,\varepsilon)$. The coefficient a_ε is then given by

$$a_\varepsilon(x) := \begin{cases} 10^{-3} & \text{if } \big||x-c_0|-r\big| < \frac{\varepsilon}{2},\ x_2 > \varepsilon \text{ and } x_1 < 1-\varepsilon \\ (h\circ c_\varepsilon)(x) & \text{else,} \end{cases} \tag{25}$$

$$\text{with } h(t) := \begin{cases} t^4 & \text{for } \frac{1}{2} < t < 1 \\ t^{\frac{3}{2}} & \text{for } 1 < t < \frac{3}{2} \\ t & \text{else.} \end{cases}$$

In our computation, we picked the truncation parameter m (according to (23)) to be in the span between 0 and 2 and the coarse mesh size H to be in the span

Table 1 Results for the relative error between the multiscale partition of unity approximation u_c^m and a reference solution u_h on a fine grid of mesh size $h = 2^{-8} \approx 0.0039 \ll \varepsilon$ which fully resolves the micro structure of the coefficient a_ε. We use the notation $\|u_c^m - u_h\|^{\text{rel}}_{L^2(\Omega)} := \|u_c^m - u_h\|_{L^2(\Omega)} / \|u_h\|_{L^2(\Omega)}$ and analogously the same for $\|u_c^m - u_h\|^{\text{rel}}_{H^1(\Omega)}$. The truncation parameter m determines the patch size and is given by (23)

H	m	$\|u_c^m - u_h\|^{\text{rel}}_{L^2(\Omega)}$	$\|u_c^m - u_h\|^{\text{rel}}_{H^1(\Omega)}$
2^{-1}	0	0.867827	0.93475
2^{-2}	0	0.865630	0.96525
2^{-2}	1	0.167501	0.37387
2^{-3}	1	0.257826	0.61681
2^{-3}	2	0.037841	0.16525
2^{-4}	2	0.063645	0.25613

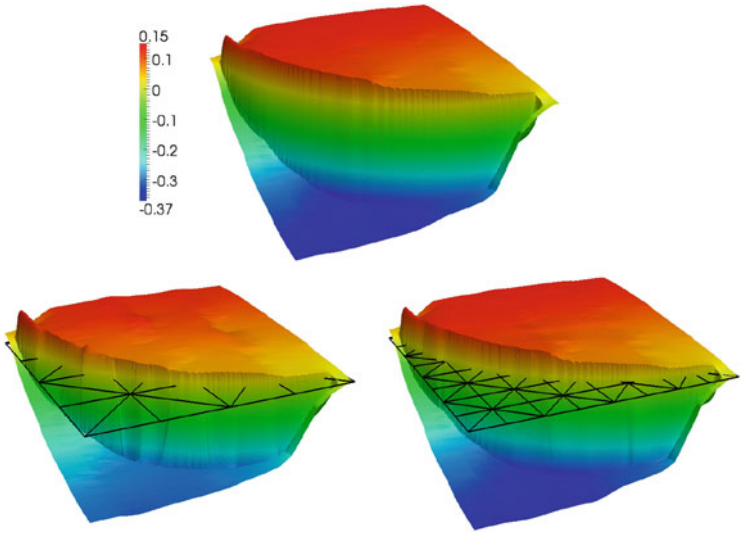


Fig. 3 The *top picture* shows the P_1 finite element reference solution u_h for $h = 2^{-8}$. The *left bottom picture* shows the multiscale approximation u_c^m for $(H, m) = (2^{-2}, 1)$ together with the corresponding coarse grid. This solution already shows the essential features of u_h. The *right bottom picture* shows the multiscale approximation u_c^m for $(H, m) = (2^{-3}, 2)$ together with the corresponding coarse grid

between 2^{-1} (i.e. $h = H^8$) and 2^{-4} (i.e. $h = H^2$). The results are depicted in Table 1. We observe that error stagnates if we decrease only H, without increasing m at the same time. However, already the modification $(H, m) = (2^{-m-1}, m) \mapsto (2^{-m-2}, m+1)$ leads to a dramatic error reduction. Despite the high contrast of order 10^3, we already obtain a highly accurate approximation for $(H, m) = (2^{-3}, 2)$. In this case, the multiscale approximation looks almost identical to the FEM reference solution for $h = 2^{-8}$ (see Fig. 3). Further numerical experiments can be found in [18, 19, 26].

References

1. A. Abdulle, E. Weinan, B. Engquist, E. Vanden-Eijnden, The heterogeneous multiscale method. Acta Numer. **21**, 1–87 (2012). MR 2916381
2. H.W. Alt, *Lineare Funktionalanalysis* (Springer, Berlin/Heidelberg, 2006)
3. I. Babuška, U. Banerjee, J.E. Osborn, Meshless and generalized finite element methods: a survey of some major results, in *Meshfree Methods for Partial Differential Equations*, Bonn, 2001. Lecture Notes in Computational Science and Engineering, vol. 26 (Springer, Berlin, 2003), pp. 1–20. MR 2003426 (2004h:65116)
4. I. Babuška, G. Caloz, J.E. Osborn, Special finite element methods for a class of second order elliptic problems with rough coefficients. SIAM J. Numer. Anal. **31**(4), 945–981 (1994). MR 1286212 (95g:65146)
5. I. Babuška, R. Lipton, Optimal local approximation spaces for generalized finite element methods with application to multiscale problems. Multiscale Model. Simul. **9**(1), 373–406 (2011). MR 2801210 (2012e:65259)
6. I. Babuška, J.M. Melenk, The partition of unity method. Int. J. Numer. Methods Eng. **40**, 727–758 (1996)
7. T. Belytschko, N. Moës, S. Usui, C. Parimi, Arbitrary discontinuities in finite elements. Int. J. Numer. Methods Eng. **50**(4), 993–1013 (2001)
8. C. Carstensen, Quasi-interpolation and a posteriori error analysis in finite element methods. M2AN Math. Model. Numer. Anal. **33**(6), 1187–1202 (1999)
9. C.A. Duarte, I. Babuška, J.T. Oden, Generalized finite element methods for three-dimensional structural mechanics problems. Comput. Struct. **77**(2), 215–232 (2000). MR 1768540 (2001b:74053)
10. C.A. Duarte, D.-J. Kim, Analysis and applications of a generalized finite element method with global–local enrichment functions. Comput. Methods Appl. Mech. Eng. **197**(6–8), 487–504 (2008)
11. C.A. Duarte, J.T. Oden, An h-p adaptive method using clouds. Comput. Methods Appl. Mech. Eng. **139**(1–4), 237–262 (1996)
12. C.A. Duarte, L.G. Reno, A. Simone, A high-order generalized FEM for through-the-thickness branched cracks. Int. J. Numer. Methods Eng. **72**(3), 325–351 (2007). MR 2355178
13. T.-P. Fries, H.-G. Matthies, Classification and overview of meshfree methods. Technical report 2003-3, Technische Universität Braunschweig (2004)
14. A. Gloria, Reduction of the resonance error—Part 1: approximation of homogenized coefficients. Math. Models Methods Appl. Sci. **21**(8), 1601–1630 (2011). MR 2826466
15. M. Griebel, M.A. Schweitzer, A particle-partition of unity method for the solution of elliptic, parabolic, and hyperbolic PDEs. SIAM J. Sci. Comput. **22**(3), 853–890 (2000). (electronic) MR 1785338 (2001i:65105)
16. M. Griebel, M.A. Schweitzer, A particle-partition of unity method. II. Efficient cover construction and reliable integration. SIAM J. Sci. Comput. **23**(5), 1655–1682 (2002). (electronic) MR 1885078 (2003b:65118)
17. V. Gupta, C.A. Duarte, I. Babuška, U. Banerjee, A stable and optimally convergent generalized FEM (SGFEM) for linear elastic fracture mechanics. Comput. Methods Appl. Mech. Eng. **266**(0), 23–39 (2013)
18. P. Henning, A. Målqvist, D. Peterseim, A localized orthogonal decomposition method for semi-linear elliptic problems. ESAIM: Math. Model. Numer. Anal. **eFirst** (2013)
19. P. Henning, D. Peterseim, Oversampling for the multiscale finite element method. Multiscale Model. Simul. **11**(4), 1149–1175 (2013). MR 3123820
20. M. Holst, Application of domain decomposition and partition of unity methods in physics and geometry. Domain decomposition methods in science and engineering, National Autonomous University of Mexico, México, 2003, pp. 63–78 (electronic). MR 2093735

21. T.Y. Hou, X.-H. Wu, A multiscale finite element method for elliptic problems in composite materials and porous media. J. Comput. Phys. **134**(1), 169–189 (1997). MR 1455261 (98e:73132)
22. T.J.R. Hughes, G.R. Feijóo, L. Mazzei, J.-B. Quincy, The variational multiscale method—a paradigm for computational mechanics. Comput. Methods Appl. Mech. Eng. **166**(1–2), 3–24 (1998). MR 1660141 (99m:65239)
23. D.-J. Kim, C.A. Duarte, S.P. Proença, A generalized finite element method with global-local enrichment functions for confined plasticity problems. Comput. Mech. **50**(5), 563–578 (2012) (English)
24. T.J. Liszka, C.A. Duarte, W. Tworzydlo, HP-meshless cloud method. Comput. Methods Appl. Mech. Eng. **139**(1–4), 263–288 (1996)
25. A. Målqvist, Multiscale methods for elliptic problems. Multiscale Model. Simul. **9**(3), 1064–1086 (2011). MR 2831590 (2012j:65419)
26. A. Målqvist, D. Peterseim, Localization of elliptic multiscale problems. Published electronically (in Mathematics of Computation, 2014)
27. A. Målqvist, D. Peterseim, Computation of eigenvalues by numerical upscaling (2012). ArXiv e-prints 1212.0090
28. J.M. Melenk, I. Babuška, The partition of unity finite element method: basic theory and applications. Comput. Methods Appl. Mech. Eng. **139**(1–4), 289–314 (1996)
29. N. Moës, J.E. Dolbow, T. Belytschko, A finite element method for crack growth without remeshing. Int. J. Numer. Methods Eng. **46**(1), 131–150 (1999)
30. J.T. Oden, C.A. Duarte, O.C. Zienkiewicz, A new cloud-based HP finite element method. Comput. Methods Appl. Mech. Eng. **153**(1–2), 117–126 (1998)
31. H. Owhadi, L. Zhang, L. Berlyand, Polyharmonic homogenization, rough polyharmonic splines and sparse super-localization. ESAIM: Math. Model. Numer. Anal. **48**(2), 517–572 (2014)
32. M.A. Schweitzer, Generalizations of the finite element method. Cent. Eur. J. Math. **10**(1), 3–24 (2012). MR 2863778 (2012k:65150)
33. T. Strouboulis, I. Babuška, K. Copps, The design and analysis of the generalized finite element method. Comput. Methods Appl. Mech. Eng. **181**(1–3), 43–69 (2000). MR 1734667 (2000h:74077)
34. T. Strouboulis, K. Copps, I. Babuška, The generalized finite element method. Comput. Methods Appl. Mech. Eng. **190**(32–33), 4081–4193 (2001). MR 1832655 (2002h:65195)
35. C. Wang, Z.-p. Huang, L.-k. Li, Two-grid partition of unity method for second order elliptic problems. Appl. Math. Mech. (English Ed.) **29**(4), 527–533 (2008). MR 2405141 (2009b:65329)
36. E. Weinan, B. Engquist, The heterogeneous multiscale methods. Commun. Math. Sci. **1**(1), 87–132 (2003). MR 1979846 (2004b:35019)

Finite Pointset Method for the Simulation of a Vehicle Travelling Through a Body of Water

Anthony Jefferies, Jörg Kuhnert, Lars Aschenbrenner, and Uwe Giffhorn

Abstract In order to shorten design cycles and reduce the cost of development in the automotive industry, simulation tools are widely used for analysis and testing throughout the vehicle development process. The Finite Pointset Method (FPM) has been applied to predict numerically the fluid motion as a vehicle travels through a body of water. FPM is a purely meshfree approach based on a generalized finite difference formulation. It discretises the flow field over a cloud of zero-dimensional, numerical points using least squares operators, and is therefore particularly suitable for flows with free surfaces. FPM has already successfully been applied to various industrial problems with significant transient surface deformation, including airbag deployment, glass formation, filling and sloshing processes. Ensuring the ability of a vehicle to travel through a body of water without being damaged is necessary in markets where monsoon rain occurs or where off-road driving is important. The development of the process to use FPM as a simulation tool for water travel was carried out in two stages: a two-phase water/air simulation of an operating air intake over a basin of water, and a single phase simulation with a moving vehicle.

Keywords CFD • PDE • FPM • Meshfree methods • General finite difference scheme

1 Introduction

The Finite Pointset Method (FPM) is a meshfree approach to solve numerically partial differential equations (PDE). The method uses a cloud of numerical points, each of which carry necessary numerical data. In order to solve a PDE, FPM needs

A. Jefferies (✉) • L. Aschenbrenner • U. Giffhorn
Volkswagen AG, Berlinerring 2, 38440 Wolfsburg, Germany
e-mail: anthony.jefferies@volkswagen.de

J. Kuhnert
Fraunhofer-Institut für Techno- und Wirtschaftsmathematik, Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany
e-mail: joerg.kuhnert@itwm.fraunhofer.de

M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations VII*, Lecture Notes in Computational Science and Engineering 100,
DOI 10.1007/978-3-319-06898-5_11

to establish approximations of derivatives, which are provided by a specialized least-squares algorithm. The least squares idea turns the whole method into a generalized finite difference method. The purpose of the present article is to give a general picture of the solution algorithms of FPM in fluid and continuum mechanical applications and to present the application of FPM to simulate water travel of a vehicle.

In this article, we will introduce the numerical scheme which is used by the Finite Pointset Method (FPM) in order to solve the classical conservation equations (mass, momentum, energy) in continuum mechanics on a meshfree basis. The material behaviour is governed by general material models. FPM is a meshfree numerical solver in continuum mechanics. It uses clouds of numerical points as a geometrical basis. The point cloud has to be sufficiently dense in the flow domain so as to allow a representative approximation of the continuum. The points move with the velocity of the continuum, i.e. it is a Lagrangian idea. The differential equations of motion are represented without the help of weak formulations. The local spatial derivatives are formed by a FPM-specialized least squares formulation, the time derivatives by simple finite differences [1–5].

In Sect. 2, point cloud management is briefly discussed.

Section 3 introduces the physical model to be solved on the meshfree numerical basis.

Section 4 sketches the numerical model as a whole, taking into account the given differential operators. A short mathematical derivation of the scheme is included, as it is non-classical.

In Sect. 5, we provide the idea how to establish stable differential operators, based on a least squares technique even for distorted point stencils.

Section 6 describes the application of FPM to water travel of a complete vehicle and presents a comparison to experimental results.

2 Pointcloud Management

A very important pre-condition to the method of FPM is the existence of a pointcloud that compactly covers the whole computational domain Ω. The density of the points is given by a function

$$h = h(\boldsymbol{x}, t) \tag{1}$$

i.e. a sufficiently smooth function in space and time. The computational domain is defined to be compactly covered by the point cloud if in any ball with radius $r_{\text{hole}} \cdot h$ inside of Ω, at least one point is found. For practical applications, a value of $r_{\text{hole}} = 0.45$ is chosen, leading to $20 \ldots 50$ points within the neighbour search ball of radius $h(\boldsymbol{x}, t)$.

Another quality constraint is that points are not allowed to assume a distance smaller than $r_{\text{small}} \cdot h$. A practicable value is $r_{\text{small}} = 0.1 \ldots 0.2$.

As the pointcloud moves with the velocity of the continuum, the quality of the pointcloud has to be maintained after each numerical time step. New points are filled if holes occur. Points are deleted, if their distance is less than $r_{\text{small}} \cdot h$.

3 Physical Model: Set of Equations Solved by FPM

The fluid flow, modelled in FPM, is governed by three conservation laws, given in differential form. The most important variables used for modelling are given in Table 1. The operator $\frac{d}{dt} = \frac{\partial}{\partial t} + \boldsymbol{v}^T \cdot \nabla$ represents the material derivative, i.e. the change of some physical quantity along a path of a fluid particle.

3.1 Conservation of Mass

The mass conservation in Lagrangian form is given by

$$\frac{d}{dt}\rho + \rho \cdot \nabla^T \boldsymbol{v} = 0 \,. \tag{2}$$

A change in density takes place only due to compression/decompression of the fluid. For fluids with constant density, (2) reduces to

$$\nabla^T \boldsymbol{v} = 0 \,, \tag{3}$$

a typical constraint for incompressible flows.

Table 1 Nomenclature

$\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right)^T$	Nabla operator
$\boldsymbol{\varepsilon}$	Deformation rate tensor
η	Viscosity
ρ	Density
Δt	Timestep size
$E = c_v T + \frac{1}{2}\boldsymbol{v}^T\boldsymbol{v}$	Total energy
$\boldsymbol{g}$	Vector of body forces
$h(\boldsymbol{x}, t)$	Interaction radius between numerical points
$p = p_{\text{hyd}} + p_{\text{dyn}}$	Pressure consisting of its hydrostatic and dynamic parts
$\boldsymbol{S}$	Deviatoric part of the stress tensor, i.e. $tr(\boldsymbol{S}) = 0$
T	Temperature
$\boldsymbol{v}_i = (u_i, v_i, w_i)^T$	Velocity vector (of numerical points with index i)
$\boldsymbol{x}_i = (x_i, y_i, z_i)^T$	Position (of a numerical point with index i)

3.2 Conservation of Momentum

The momentum conservation, also in Lagrangian form, is given by

$$\frac{d}{dt}(\rho \boldsymbol{v}) + (\rho \boldsymbol{v}) \cdot \nabla^T \boldsymbol{v} = \left(\nabla^T \boldsymbol{S}\right)^T - \nabla p + \rho \cdot \boldsymbol{g} \ . \tag{4}$$

The material derivative of velocity (i.e. total acceleration) is given by the gradient of pressure, body forces as well as the divergence of the stress tensor.

3.3 Conservation of Energy

The principle of conservation of total energy is expressed by the equation

$$\frac{d}{dt}(\rho E) + (\rho E) \cdot \nabla^T \boldsymbol{v} = \nabla^T (\boldsymbol{S} \cdot \boldsymbol{v}) - \nabla^T (p \cdot \boldsymbol{v}) + \left(\rho \cdot g^T \cdot \boldsymbol{v}\right) + \nabla^T (k \cdot \nabla T) \ . \tag{5}$$

The material derivative of global energy is given by the work of the pressure and the stress tensor. Additionally, changes in kinetic energy arise due to movement in the gravitation field. Heat exchange is also taken into account. For the water crossing application, this equation plays a minor role and will not be considered in the numerical scheme developed below.

3.4 Formulation for the Stress Tensor

In fact, the physical model above is general. In order to derive a useful numerical scheme, we split the stress tensor into its viscous and solid parts

$$\boldsymbol{S} = \boldsymbol{S}_{\text{solid}} + \boldsymbol{S}_{\text{visc}} \ . \tag{6}$$

We will not consider the solid part, as it is of no importance for water crossing applications. The viscous stresses are defined as

$$\boldsymbol{S}_{\text{visc}} = \eta \cdot \frac{d\boldsymbol{\varepsilon}}{dt} \ . \tag{7}$$

The strain rate tensor $\frac{d\boldsymbol{\varepsilon}}{dt}$ is given by

$$\frac{d\boldsymbol{\varepsilon}}{dt} = \frac{1}{2}\left[\nabla \boldsymbol{v}^T + \left(\nabla \boldsymbol{v}^T\right)^T\right] - \frac{1}{3}\left(\nabla^T \boldsymbol{v}\right) \cdot \boldsymbol{I} \ , \tag{8}$$

accounting for the local deformation rate of the fluid. The model above requires the viscosity η, which will be subject to turbulent modelling, which however will not be discussed here, either.

4 Numerical Model: Derivation of Time Integration of the Conservation Laws and Stresses

For numerical modelling, we rewrite Eqs. (2) and (4) in primitive form, i.e. in a density and velocity formulation. We obtain

$$\frac{d}{dt}\rho + \rho \cdot \nabla^T \boldsymbol{v} = 0 \,, \tag{9}$$

$$\frac{d}{dt}(\boldsymbol{v}) = \frac{1}{\rho}\left(\nabla^T \boldsymbol{S}\right)^T - \frac{1}{\rho}\nabla p + \boldsymbol{g} \,. \tag{10}$$

For the following numerical model, we assume incompressible flow, i.e. we require $\nabla^T v = 0$. With Eq. (8), we are able to establish an implicit numerical scheme for the meshfree framework of FPM. The scheme is a direct approach, i.e. the differential equations are approximated without the help of weak formulations. The spatial derivatives in the physical model are replaced by the least squares approximations based on the local pointcloud (refer to Sect. 5). We symbolize this by replacing the mathematical differential operators ∇, Δ etc. by their numerical representations $\tilde{\nabla}$, $\tilde{\Delta}$ and so on.

4.1 Integration of the Velocity

We represent the time derivatives by a difference formulation. For simplicity, we keep a first order time approximation, keeping in mind that the following steps would in fact be possible also for at least second order time integration.

$$\frac{\hat{\boldsymbol{v}}^{n+1} - \boldsymbol{v}^n}{\Delta t} = \frac{1}{\rho}\left(\tilde{\nabla}^T \boldsymbol{S}_{\text{visc}}^{n+1}\right)^T - \frac{1}{\rho}\tilde{\nabla}\hat{p} + \boldsymbol{g} \,. \tag{11}$$

The viscous stresses are simply computed by (7), such that it follows

$$\frac{\hat{\boldsymbol{v}}^{n+1} - \boldsymbol{v}^n}{\Delta t} = \frac{1}{\rho}\left(\tilde{\nabla}^T \left(\eta \cdot \frac{d\boldsymbol{\varepsilon}}{dt}\bigg|^{n+1}\right)\right)^T - \frac{1}{\rho}\tilde{\nabla}\hat{p} + \boldsymbol{g} \,. \tag{12}$$

Details about the numerical treatment of the numerical pressure $\hat{p}$ will be given in Sect. 4.2.

Recalling the definition of the strain rate tensor (Eq. (8)), we are able to define

$$\Psi_\eta(\boldsymbol{v}) \equiv \left(\nabla^T\left(\eta \cdot \frac{d\boldsymbol{\varepsilon}}{dt}\Big|\right)\right)^T , \tag{13}$$

knowing, that this term is just a function of the local velocity and its local, spatial derivatives. The full expression is

$$\tilde{\Psi}_\eta^T(\boldsymbol{v}) = \tilde{\nabla}^T\left(\eta\tilde{\nabla}\right)\boldsymbol{v}^T + \left(\tilde{\nabla}\eta\right)^T \cdot \left(\tilde{\nabla}\boldsymbol{v}^T\right)^T + \frac{1}{3}\eta\left(\tilde{\nabla}D\right)^T - \frac{2}{3}\left(\tilde{\nabla}\eta\right)^T D , \tag{14}$$

where $D = \tilde{\nabla}^T\boldsymbol{v}$ is simply the divergence of velocity. If a standard computation is performed with $\tilde{\nabla}\eta = 0$ and $D = 0$, it suffices to evaluate the numerical operator of $\tilde{\nabla}^T\left(\eta\tilde{\nabla}\right)$. We assume $D = 0$. However, for $\tilde{\nabla}\eta \neq 0$, additional evaluations of first order numerical derivatives are necessary. Measurements of computation times show, that cases with $\tilde{\nabla}\eta \neq 0$ run much slower, as the system matrix contains about twice as much non-zero entries. As the viscosity contains the turbulent effects, this property is significant for the water travelling applications.

Using definition (13), Eq. (12) is expressed as

$$\left(I - \frac{\Delta t}{\rho}\tilde{\Psi}_\eta\right)\left(\hat{\boldsymbol{v}}^{n+1}\right) = \boldsymbol{v}^n - \frac{\Delta t}{\rho}\tilde{\nabla}\hat{p} + \Delta t \cdot \boldsymbol{g} . \tag{15}$$

In order to find the solution for the velocity at time step $n+1$, FPM solves exactly this equation. On the left hand side, we find terms depending on time level $n+1$, on the right hand sight, there are terms of time level n. By its character, Eq. (15) is a time implicit scheme for the velocity. The left hand side operator $\left(I - \frac{\Delta t}{\rho}\tilde{\Psi}_\eta\right)$ gives rise to the construction of a big sparse matrix, each line of which containing the local, discrete representation of this operator. The right hand side appears as a load vector. Hence, Eq. (15) represents a big sparse linear system. The solutions are the velocity components associated with the appropriate particle location.

With $\hat{\boldsymbol{v}}^{n+1}$, we do not compute the true solution of the velocity but a first guess. $\hat{\boldsymbol{v}}^{n+1}$ might be in need of further correction, as it might not meet all constraints. For example, the divergence $\tilde{\nabla}^T\hat{\boldsymbol{v}}^{n+1}$ might not be the correct value. The correction of $\hat{\boldsymbol{v}}^{n+1}$ towards the correct velocity $\boldsymbol{v}^{n+1}$ is directly connected to the computation of the correction pressure, which is examined in Sect. 4.2.

4.2 Solution to the Pressure

In order for the numerical model to produce a correct approximation of the velocity (15), we need a good model for the pressure $\hat{p}$. If $\hat{p}$ is wrong, especially the divergence of the velocity $\tilde{\nabla}^T\hat{v}^{n+1}$ produced by (15) will most probably be wrong.

On the other hand, the pressure at time level $n+1$ is not known in advance. In order to provide at least a good guess of $\hat{p}$, it is worthwhile to recall how pressure and velocity are connected. We recall Eq. (10),

$$\frac{d}{dt}v = \frac{1}{\rho}\left(\nabla^T \boldsymbol{S}\right)^T - \frac{1}{\rho}\nabla p + \boldsymbol{g}\ ,$$

and study it in more detail. Application of the divergence operator results in

$$\frac{d}{dt}\left(\nabla^T \cdot v\right) + \Phi\left(v\right) = \nabla^T\left(\frac{1}{\rho}\left(\nabla^T \boldsymbol{S}_{\text{visc}}\right)^T\right) - \nabla^T\left(\frac{1}{\rho}\nabla p\right) + \nabla^T \boldsymbol{g} \tag{16}$$

where $\Phi\left(v\right)$ turns out to be of the form

$$\Phi\left(\boldsymbol{v}\right) = \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 + \left(\frac{\partial w}{\partial z}\right)^2 + 2\left(\frac{\partial u}{\partial y}\right)\cdot\left(\frac{\partial v}{\partial x}\right) + \ldots\ .$$

Since the velocity field is known throughout the computational domain, we are free to establish a Poisson-type equation in order to solve for the pressure field:

$$\nabla^T\left(\frac{1}{\rho}\nabla p\right) = -\frac{d}{dt}\left(\nabla^T \boldsymbol{v}\right) - \Phi\left(\boldsymbol{v}\right) + \nabla^T\left(\frac{1}{\rho}\left(\nabla^T \boldsymbol{S}\right)^T\right) + \nabla^T \boldsymbol{g}. \tag{17}$$

We split the pressure into a hydrostatic part and into a dynamic part. The hydrostatic part is formed from body forces and inner stresses

$$\nabla^T\left(\frac{1}{\rho}\nabla p_{\text{hyd}}\right) = \nabla^T \boldsymbol{g}\ . \tag{18}$$

The dynamic part arises due to the movement of the fluid (i.e. the dynamic part vanishes as the fluid velocity becomes zero)

$$\nabla^T\left(\frac{1}{\rho}\nabla p_{\text{dyn}}\right) = -\frac{d}{dt}\left(\nabla^T \boldsymbol{v}\right) - \Phi\left(\boldsymbol{v}\right) + \nabla^T\left(\frac{1}{\rho}\left(\nabla^T \boldsymbol{S}\right)^T\right). \tag{19}$$

Equations (18) and (19) provide a natural splitting of the pressure, which will be used for the numerical scheme.

We define the pressure $\hat{p}$ of the solution model for the velocity (15) as

$$\hat{p} = p_{\text{hyd}}^{n+1} + p_{\text{dyn}}^{n}. \tag{20}$$

By virtue of Eq. (18), the numerical solution of the hydrostatic pressure at the future time step is straight forward:

$$\tilde{\nabla}^T\left(\frac{1}{\rho}\tilde{\nabla} p_{\text{hyd}}^{n+1}\right) = \tilde{\nabla}^T \boldsymbol{g}\ . \tag{21}$$

It is, however, not straight forward to determine the dynamic pressure at time level $n+1$, as we would need the velocity $\boldsymbol{v}^{n+1}$, as yet unknown. Therefore, p^n_{dyn} of the previous time step is taken into account. We thus introduce an error concerning the pressure. In order to fix this situation, we employ a correction pressure, which is applied as a posteriori formulation.

Suppose we have computed the solution $\hat{\boldsymbol{v}}^{n+1}$ to the numerical model (15). We reproduce a divergence free velocity by finding a correction term $\boldsymbol{\varepsilon}^{n+1}$ similar to Chorin's projection idea [6]. Our model (15) would produce a slightly different solution, if the correction pressure is employed additionally in the sense

$$\left(I - \frac{\Delta t}{\rho}\tilde{\Psi}_\eta\right) \cdot \boldsymbol{v}^{n+1} = \boldsymbol{v}^n - \frac{\Delta t}{\rho}\tilde{\nabla}\hat{p} - \frac{\Delta t}{\rho}\tilde{\nabla}\boldsymbol{\varepsilon}^{n+1} + \Delta t \cdot \boldsymbol{g} \ . \tag{22}$$

Subtracting Eq. (15) from (22) yields

$$\boldsymbol{v}^{n+1} - \hat{\boldsymbol{v}}^{n+1} - \frac{\Delta t}{\rho}\tilde{\Psi}_\eta\left(\boldsymbol{v}^{n+1} - \hat{\boldsymbol{v}}^{n+1}\right) = -\frac{\Delta t}{\rho}\tilde{\nabla}\boldsymbol{\varepsilon}^{n+1} \ . \tag{23}$$

We apply the divergence operator to (23) and obtain

$$\left(\tilde{\nabla}^T\boldsymbol{v}^{n+1}\right) - \left(\tilde{\nabla}^T\hat{\boldsymbol{v}}^{n+1}\right) - \tilde{\nabla}^T\left(\frac{\Delta t}{\rho}\tilde{\Psi}_\eta\left(\boldsymbol{v}^{n+1} - \hat{\boldsymbol{v}}^{n+1}\right)\right) = -\tilde{\nabla}^T\left(\frac{\Delta t}{\rho}\tilde{\nabla}\boldsymbol{\varepsilon}^{n+1}\right) \ . \tag{24}$$

Moreover we assume the term

$$\tilde{\nabla}^T\left(\frac{\Delta t}{\rho}\tilde{\Psi}_\eta\left(\boldsymbol{v}^{n+1} - \hat{\boldsymbol{v}}^{n+1}\right)\right) \approx 0 \ , \tag{25}$$

such that we can neglect it. One can always achieve this by adjusting the size of the time step. Thus, from (24) it follows

$$\tilde{\nabla}^T\left(\frac{\Delta t}{\rho}\tilde{\nabla}\boldsymbol{\varepsilon}^{n+1}\right) = -\left(\tilde{\nabla}^T\boldsymbol{v}^{n+1}\right)_{\text{desired}} + \left(\tilde{\nabla}^T\hat{\boldsymbol{v}}^{n+1}\right) \ . \tag{26}$$

The equation above represents a Poisson-type equation to solve for the correction pressure $\boldsymbol{\varepsilon}^{n+1}$. That, however, requires the knowledge of the two right hand side terms. $\tilde{\nabla}^T\hat{\boldsymbol{v}}^{n+1}$ is evaluated by numerically differentiating the existing (i.e. known) preliminary velocity. Furthermore, we require

$$\left(\tilde{\nabla}^T\boldsymbol{v}^{n+1}\right)_{\text{desired}} \stackrel{!}{=} 0 \ ,$$

as we are concerned with incompressible flows for the current project. Compressible or weakly compressible flows will not be discussed here.

4.3 Coupled Pressure Velocity Solution

Velocity and pressure can be coupled by a penalty formulation. We enhance the numerical scheme for the velocity (15) by the correction pressure and solve it together with the scheme (26) which penalizes incorrect values of the divergence of velocity

$$\begin{aligned} \left(I - \frac{\Delta t}{\rho}\tilde{\Psi}_{\eta}\right)\left(\hat{\boldsymbol{v}}^{n+1}\right) + \frac{\Delta t}{\rho}\tilde{\nabla}\boldsymbol{\varepsilon}^{n+1} &= \boldsymbol{v}^{n} - \frac{\Delta t}{\rho}\tilde{\nabla}\hat{p} + \Delta t \cdot \hat{\boldsymbol{g}} \\ \left(\tilde{\nabla}^{T}\hat{\boldsymbol{v}}^{n+1}\right) - \tilde{\nabla}^{T}\left(\frac{\Delta t_{\text{virt}}}{\rho}\tilde{\nabla}\boldsymbol{\varepsilon}^{n+1}\right) &= \left(\tilde{\nabla}^{T}\boldsymbol{v}^{n+1}\right)_{\text{desired}} \end{aligned} \tag{27}$$

This powerful combination of velocity and pressure provides more numerical stability. The virtual time step size is flexible. Choosing $\Delta t_{\text{virt}} = \Delta t$ would result in an implicit Chorin projection, $\Delta t_{\text{virt}} = 0$ results in a direct condition on the value of $\tilde{\nabla}^{T}\hat{\boldsymbol{v}}^{n+1}$. Theoretically, the best choice is $\Delta t_{\text{virt}} = 0$, which, however, provides poorly conditioned linear systems which are hard to be solved by iterative methods. Nevertheless, Δt_{virt} should be chosen as small as possible, for this we introduce the definition

$$\Delta t_{\text{virt}} = A_{\text{virt}} \cdot \Delta t \ . \tag{28}$$

with the nondimensional, universal numerical parameter A_{virt}. The smaller A_{virt} is chosen, the more tight the numerical velocity is forced towards the desired divergence, however solving the linear system becomes more time consuming. Practically, a choice of $A_{\text{virt}} = 0.1 \ldots 0.01$ provides good conditioning of the linear system, and a solution which is very close to the desired divergence of velocity, i.e. perfect for the application of water travel.

5 Construction of the Differential Operators

The numerical schemes above are based on the meshfree differential operators. This section gives a sketch how to establish these operators in general. Let us suppose there is a pointcloud being sufficiently dense. The positions of the particles are given by

$$\boldsymbol{x}_i = \begin{pmatrix} x_i & y_i & z_i \end{pmatrix}^{T}, i = 1 \ldots N \ , \tag{29}$$

where N is the number of particles in the pointcloud. Suppose furthermore that any function f is given only at the discrete particle locations, i.e.

$$f_i \equiv f\left(x_i\right) \ . \tag{30}$$

The vector of discrete function values is given by

$$\boldsymbol{f} \equiv \left(f_1, f_2, \ldots, f_N\right)^T \ . \tag{31}$$

We call the numerical differential operators in FPM those vectors which provide an approximation of some derivative in the sense

$$\partial^*_{num} f\left(x_i\right) = \tilde{\partial}^* f\left(x_i\right) = \tilde{\partial} f_i = \sum_{j=1}^{N} c^*_{ij} \cdot f_j = \left(\boldsymbol{c}^*_i\right)^T \cdot \boldsymbol{f} \ . \tag{32}$$

The star (*) in the equation above is a placeholder for all the numerous differential operators needed by FPM, which are

$$\begin{aligned} \boldsymbol{c}^0_i &= \text{the numerical operator for function approximation} \\ \boldsymbol{c}^x_i &= \text{the numerical operator for the x-derivative} \\ \boldsymbol{c}^y_i &= \text{the numerical operator for the y-derivative} \\ \boldsymbol{c}^z_i &= \text{the numerical operator for the z-derivative} \\ \boldsymbol{c}^{\Delta}_i &= \text{the numerical operator for the Laplacian} \end{aligned} \tag{33}$$

just to name those most often appearing. Our intention is to develop operators which are, as shown in (32), independent of the underlying function values. Having operators which work generally for all given functions will save a lot of computation time.

We also introduce a weight function which switches on the particular neighbours close to some particle i by

$$w_{ij} = w\left(r\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)\right) = w\left(r_{ij}\right) \ . \tag{34}$$

The distance function $r\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)$ is given by

$$r\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = 2\frac{\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_2}{\left(h\left(\boldsymbol{x}_i\right) + h\left(\boldsymbol{x}_j\right)\right)} \ . \tag{35}$$

The interaction radius $h\left(\boldsymbol{x}_i\right)$ is defined in (1), however we disregard the dependence on t since the operators are constructed at a fixed, discrete time. It actually rules the local density of the particles, i.e. the mean distance between particles. We provide a weight function which becomes 0 if $r > 1$ and which becomes 1 if $r = 0$ and which is n-times continuously differentiable, for example

$$w(r) = \begin{cases} \left(1 - r^2\right)^{\gamma}, & if \ r \leq 1 \\ 0, & \text{otherwise} \end{cases} \ . \tag{36}$$

We can group the discrete weights in the weight matrix

$$\boldsymbol{W}_i = \begin{pmatrix} w_{i1} & & & 0 \\ & w_{i2} & & \\ & & \ddots & \\ 0 & & & w_{iN} \end{pmatrix} \tag{37}$$

having the discrete weights on its diagonal and zeros otherwise.

5.1 General Least Squares Procedure for Operators

We are searching for the operator $\boldsymbol{c}_i$ fulfilling the least squares criterion

$$\frac{1}{2}\left\|\boldsymbol{W}_i^{-1}\cdot\boldsymbol{c}_i\right\|^2 = \frac{1}{2}\boldsymbol{c}_i^T\cdot\boldsymbol{W}_i^{-1T}\cdot\boldsymbol{W}_i^{-1}\cdot\boldsymbol{c}_i \stackrel{!}{=} \frac{1}{2}\boldsymbol{c}_i^T\cdot\boldsymbol{W}_i^{-2}\cdot\boldsymbol{c}_i = \min \tag{38}$$

under the consistency constraints

$$\boldsymbol{K}_i^T\cdot\boldsymbol{c}_i = \boldsymbol{b}\ . \tag{39}$$

For simplicity, we omit the star and, instead of writing $\boldsymbol{c}_i^*$, we simply employ $\boldsymbol{c}_i$. The matrix $\boldsymbol{K}_i$ represents test functions (given as discrete values set for the particles) for which the numerical operator shall give a distinct value. As for example, the numerical operator for the x-derivative $\boldsymbol{c}_i^x$ shall deliver 0 if operating on a constant function $\boldsymbol{k}_i^1 = (1,1,\ldots,1)^T$ or a quadratic function $\boldsymbol{k}_i^3 = \left(\left(x_1-x_i\right)^2,\left(x_2-x_i\right)^2,\ldots,\left(x_N-x_i\right)^2\right)^T$, but it shall deliver 1 if operating on the linear function $\boldsymbol{k}_i^2 = \left(x_1-x_i, x_2-x_i,\ldots,x_N-x_i\right)^T$. In other words, we have the conditions (among others!)

$$\begin{aligned} \left(\boldsymbol{k}_i^1\right)^T\cdot\boldsymbol{c}_i^x &= 0 \\ \left(\boldsymbol{k}_i^2\right)^T\cdot\boldsymbol{c}_i^x &= 1 \\ \left(\boldsymbol{k}_i^3\right)^T\cdot\boldsymbol{c}_i^x &= 0 \end{aligned} \tag{40}$$

In general, the columns of matrix $\boldsymbol{K}_i$ represent M discrete test functions in the sense

$$\boldsymbol{K}_i \equiv \left(\boldsymbol{k}_i^1\ \boldsymbol{k}_i^2\ \boldsymbol{k}_i^3\ \ldots\ \boldsymbol{k}_i^M\right)\ . \tag{41}$$

Usually, a convenient choice for the $\boldsymbol{k}_i^p, p = 1\ldots M$ is to use the monomials up to a certain order, i.e. the numerical operator reproduces these functions exactly. In addition to that, numerical stability can be achieved by determining the action of the operator on the delta function as well, i.e. we choose

$$\boldsymbol{k}_i^M = \left(\delta_{1i}\ \delta_{2i}\ \ldots\ \delta_{Mi}\right)^T\ . \tag{42}$$

The right hand side vector $\boldsymbol{b}$ consequently contains the corresponding values to be delivered by the operator if applied to the test functions.

The minimization problem (38) together with the constraints (39) can be solved using Lagrangian multipliers in the sense

$$\boldsymbol{W}_i^{-2} \cdot \boldsymbol{c}_i + \sum_{k=1}^{M} \lambda_k \cdot \boldsymbol{k}_i^k = 0 \,. \tag{43}$$

Let us transform Eq. (43) into

$$\boldsymbol{c}_i + \sum_{k=1}^{M} \lambda_k \cdot \boldsymbol{W}_i^2 \cdot \boldsymbol{k}_i^k = 0. \tag{44}$$

and multiply from left with the p-th condition vector $\boldsymbol{k}_i^p$

$$\left(\boldsymbol{k}_i^p\right)^T \cdot \boldsymbol{c}_i + \sum_{k=1}^{M} \lambda_k \left(\boldsymbol{k}_i^p\right)^T \cdot \boldsymbol{W}_i^2 \cdot \boldsymbol{k}_i^k = 0, p = 1 \ldots M \,, \tag{45}$$

which, by virtue of the consistency conditions (39) transforms to

$$b_i^p + \sum_{k=1}^{M} \lambda_k \left(\boldsymbol{k}_i^p\right)^T \cdot \boldsymbol{W}_i^2 \cdot \boldsymbol{k}_i^k = 0, p = 1 \ldots M \,. \tag{46}$$

Equation (46) provides M equations for the same number of unknowns λ_k. In a more compact manner, Eq. (44) can be written as

$$\boldsymbol{c}_i + \boldsymbol{W}_i^2 \cdot \boldsymbol{K}_i \cdot \lambda_i = 0 \,, \tag{47}$$

where λ_i is the vector of all M values of lambda.

Equation (47) immediately shows that the operator to be searched for is a linear combination of the test functions together with the weight potential. Multiplying from left with the matrix of test functions gives

$$\boldsymbol{K}_i^T \cdot \boldsymbol{c}_i + \boldsymbol{K}_i^T \cdot \boldsymbol{W}_i^T \cdot \boldsymbol{W}_i \cdot \boldsymbol{K}_i \cdot \lambda_i = 0 \,, \tag{48}$$

which, by virtue of (39), provides

$$b_i + \boldsymbol{K}_i^T \cdot \boldsymbol{W}_i^T \cdot \boldsymbol{W}_i \cdot \boldsymbol{K}_i \cdot \lambda_i = 0 \,. \tag{49}$$

Equation (49) provides an $M \times M$ linear system to be solved for λ_i which then serves for providing the solution

$$\boldsymbol{c}_i = -\boldsymbol{W}_i^2 \cdot \boldsymbol{K}_i \cdot \lambda_i \,. \tag{50}$$

The computationally most expensive part in solving for the differential operators is the production of the matrix $\boldsymbol{K}_i^T \cdot \boldsymbol{W}_i^T \cdot \boldsymbol{W}_i \cdot \boldsymbol{K}_i$ together with the solution of the linear system (49). This embeds one risk which lies in the fact that we produce the square of the matrix $\boldsymbol{W}_i \cdot \boldsymbol{K}_i$. This matrix is usually well conditioned. Occasionally, its conditioning becomes bad, usually when the local pointcloud becomes distorted. Better numerical conditioning of the local system is achieved if we solve the system (39) by a numerical least norm approach instead of the Lagrange multiplier formulation used above. The procedure sketched in this section will fail if the pointcloud locally deforms to a less dimensional manifold, i.e. if (for a 3D computation) the particles are nearly placed in a plane or on a line etc. This happens for example if water squirts off the free surface. In this case, we strictly employ pseudo-inverse solutions to the system (49) in order to keep the differential operators in good condition.

6 Water Travel

Internal combustion engines rely on a constant supply of air in order to provide the necessary oxygen for the combustion process inside the engine cylinders. In passenger vehicles, this air is taken from the atmosphere through an air intake. The position of the opening of the air intake is typically situated behind the air inlet grill at the front of the vehicle, as high as possible. It is positioned here so that the air available is at the lowest possible temperature, since air in the engine compartment or in the underbody may already have been heated by the engine cooling system or other processes. It is also designed so that a degree of water ingress will be deflected without entering the engine air intake system. Significant water ingress into the engine can damage or destroy it. Packaging requirements must also be taken into account, since components such as heat exchangers and the engine fan are also normally positioned behind the air inlet grill.

Experimental water travel testing of complete vehicles is carried out regularly by Volkswagen. However, it is challenging to observe directly the flow characteristics of water close to the air intake during such testing, as it is difficult to place a camera so that it can record the water motion without itself getting wet and with sufficient light to capture satisfactory images.

In the early stages of vehicle development, particularly those with configurations which have not previously been developed for series production, a physical prototype of a vehicle is not available. A passenger vehicle ought to be able to drive through a depth of water, without the engine being damaged due to water ingress. In climate zones subject to monsoon weather conditions this is particularly relevant, as the requirement to drive on flooded streets occurs often. Off-road vehicles have more stringent requirements in this regard, where there is the enhanced stipulation that such vehicles are able to travel through and also stop and start in shallow rivers. The increasing electrification in the automotive industry adds to traditional electric components such as the electric generator which are sensitive to contact with water. In order to analyse and test possible vehicle design configurations it is desirable to be

able to conduct numerical simulations of vehicle water travel in a timely and physically accurate manner. Furthermore, simulations are a cost effective means of evaluating a vehicle configuration without the expense of producing a physical prototype.

Previous work at Volkswagen on simulating a complete vehicle travelling through a body of water include a Finite Volume (FV) based solution of the Navier Stokes equations with Volume of Fluid (VOF) modelling of the water/air interface, and Smoothed Particle Hydrodynamics (SPH) simulations. FV suffered from relatively long run times due to the very fine meshing required accurately to capture the water/air interface. The requirements for the quality of the surface mesh were also stricter than those for the particle based methods, thus necessitating longer mesh preparation. For numerical stabilisation when using SPH, an artificially high numerical viscosity is required, thus yielding an unphysical solution which does not exhibit separation of water droplets from the continuous water phase to the degree experimentally observed.

6.1 Water Travel Vehicle Simulation and Experiment

A simulation was carried out with FPM of a vehicle travelling through a body of water. The vehicle studied is a generation 6 Volkswagen Golf with a 2.01 103 kW turbo diesel engine. The geometry of the water basin is that of the experimental test basin at Volkswagen's development centre in Wolfsburg. The water is represented by 3.5×10^6 particles. In the simulation, the vehicle moves through a test basin of water with a velocity profile derived from an experiment. The water has a depth at rest of 30 cm above the water basin. The simulation is a one phase simulation with only the water phase present. The vehicle and the water basin were both represented geometrically by watertight triangulated meshes. The geometrical scope of the vehicle is typical for a complete underhood simulation, with the vehicle exterior shell and the external representation of all components of the engine compartment, underbody and wheel arches with rotating wheels. This setup aims to simulate a test run for which experimental results are available.

6.2 Results

Figure 1 shows the position of the water surface after 4 s of travel. Phemonologically, the flow exhibits several features also present in the experiment. At the front of the vehicle, a wave is beginning to form in front of the bumper, with spray forming at the top of the wave. Behind the vehicle, a V-shaped wake has developed, choppy directly behind the vehicle, with the surface gradually smoothening as the vehicle moves further away.

Figure 2 displays the wave building at the front of the vehicle in the simulation and experiment after 4 s of vehicle travel. Figure 2 exhibits spray building on a wave front at a height with little variance across the vehicle in both simulation and experiment. Figure 3 shows the rear of the vehicle after 4 s of travel in the simulation

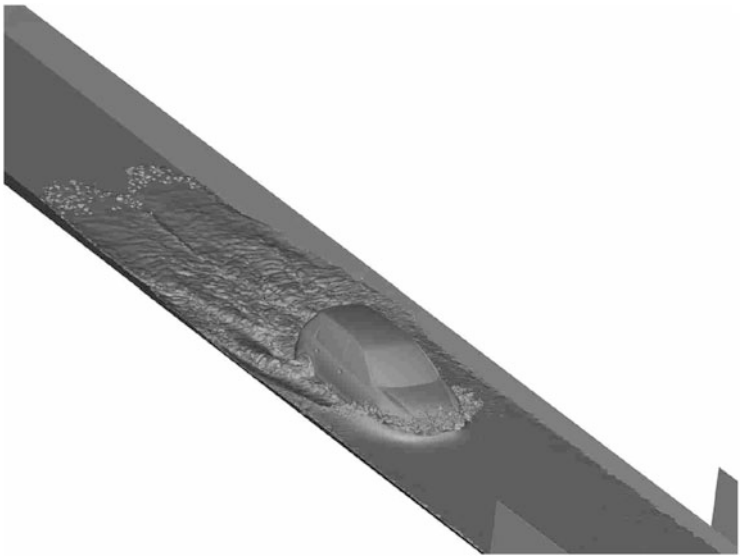


Fig. 1 Simulation: Isometric view of the vehicle in the test basin at 4 s

Fig. 2 Experiment (*left*) and simulation (*right*): Front view of the vehicle at 4 s

Fig. 3 Experiment (*left*) and simulation (*right*): Rear view of the vehicle at 4 s

and the experiment. Qualitatively, Fig. 3 exhibits several features in the simulation, such as the reduced depth of water to the left of the vehicle doors, the side of the bow wave forming over the front left wing of the vehicle, a wave from the right wheel arch extending to the edge of the basin and a channel of water flow from the drive train tunnel forming behind the centre of the vehicle, which can also be observed in the experiment.

7 Conclusion

FPM was used to simulate water flow in an air intake placed in a water basin. The wave formation and the detachment of water droplets is captured by the simulation.

FPM was successfully employed to simulate water motion in a basin. The simulation captures several qualitative features which are also observed in the experiment. The next step will be to attempt to simulate vehicle water travel with 2-phase modelling for water and air, analogous to the simulation of the air intake. Although there is a strong expectation that the computational resources available will continue to increase, further optimisation of the solution algorithms so as to reduce memory requirements and computational time is desirable in order to use these resources as effectively as possible. Further work will investigate the pressure distribution on the components in order to predict damage during travel through water. A fluid-structure interaction simulation will also be developed, in order to calculate the motion of the vehicles components during water travel. The buoyancy of the vehicle, suspension motion of the vehicle during water travel, as well as predicting the motion of the vehicle through water as a function of the engine power output and drag force on the vehicle due to water are all features which would improve the predictive value of the simulations.

References

1. J. Kuhnert, Meshfree numerical schemes for time dependent problems in fluid and continuum mechanics, in *Advances in PDE Modeling and Computation*, ed. by S. Sundar (Ane Books, New Delhi, 2014)
2. I. Ostermann, J. Kuhnert, D. Kolymbas, C.-H. Chen, I. Polymerou, V. Smilauer, C. Vrettos, D. Chen, Meshfree generalized finite difference methods in soil mechanics part I: theory. Int. J. Geomath. (2013). doi:10.1007/s13137-013-0048-7
3. A. Tramecon, J. Kuhnert, *Simulation of Advanced Folded Airbags with VPS PAMCRASH/FPM: Development and Validation of Turbulent Flow Numerical Simulation Techniques Applied to Curtain Bag Deployments*, SAE International (2012). doi:10.4271/2013-01-1158
4. E. Uhlmann, R. Gerstenberger, J. Kuhnert, Cutting simulation with the meshfree finite pointset method, in *14th CIRP Conference on Modeling of Machining Operations (CIRP CMMO)*, Turin, 2013

5. J. Kuhnert, Finite Pointset Method (FPM): meshfree flow solver with applications to elasto-plastic material laws, in *Proceedings of the International Conference on Particle-Based Methods, Particles 2009*, Barcelona, ed. by E. Oñate, D.R.J. Owen
6. A.J. Chorin, A numerical method for solving incompressible flow problems. J. Comput. Phys. **2**, 12–26 (1967)

Meshfree Finite Differences for Vector Poisson and Pressure Poisson Equations with Electric Boundary Conditions

Dong Zhou, Benjamin Seibold, David Shirokoff, Prince Chidyagwai, and Rodolfo Ruben Rosales

Abstract We demonstrate how meshfree finite difference methods can be applied to solve vector Poisson problems with electric boundary conditions. In these, the tangential velocity and the incompressibility of the vector field are prescribed at the boundary. Even on irregular domains with only convex corners, canonical nodal-based finite elements may converge to the wrong solution due to a version of the Babuška paradox. In turn, straightforward meshfree finite differences converge to the true solution, and even high-order accuracy can be achieved in a simple fashion. The methodology is then extended to a specific pressure Poisson equation reformulation of the Navier-Stokes equations that possesses the same type of boundary conditions. The resulting numerical approach is second order accurate and allows for a simple switching between an explicit and implicit treatment of the viscosity terms.

Keywords Meshfree • Finite-differences • Navier-Stokes • Incompressible • Vector Poisson equation • Pressure Poisson equation • Reformulation • Manufactured solution • High-order

D. Zhou (✉) • B. Seibold
Temple University, 1805 North Broad Street, Philadelphia, PA 19122, USA
e-mail: dong.zhou@temple.edu; seibold@temple.edu

D. Shirokoff
McGill University, 805 Sherbrooke Street West, Montreal, QC H3A 0B9, Canada
e-mail: david.shirokoff@mail.mcgill.ca

P. Chidyagwai
Loyola University Maryland, 4501 N. Charles Street, Baltimore, MD 21210, USA
e-mail: pchidyagwai@loyola.edu

R.R. Rosales
Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA
e-mail: rrr@math.mit.edu

M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations VII*, Lecture Notes in Computational Science and Engineering 100,
DOI 10.1007/978-3-319-06898-5_12

1 Introduction

The numerical approximation of vector fields that are incompressible is often times challenging because incompressibility, $\nabla \cdot \mathbf{u} = 0$, is a global constraint that may not fit within the framework of simple discretization approaches of the complete problem. The instationary incompressible Navier-Stokes equations (NSE) represent a prime example, in which the time-evolution of the velocity field is given, but not of the pressure (which is a Lagrange multiplier associated with $\nabla \cdot \mathbf{u} = 0$). As a consequence, there is no single canonical way to advance the NSE forward in time. Similarly, in electrostatics the electric field and the magnetic potential are solutions to vector Poisson equations with divergence constraints.

One methodology to circumvent the incompressibility constraint inside the computational domain is to formulate a different problem that imposes $\nabla \cdot \mathbf{u} = 0$ as a boundary condition instead. Under certain circumstances, this new problem has the same solution as the original problem, while at the same time giving rise to new numerical approximation methods. For electrostatic problems (see Sect. 2) this approach is employed and analyzed in [15,22], and in the context of incompressible fluid flows (see Sect. 3) it has been proposed in [29]. The specific boundary conditions for these problems consist of enforcing the tangential component(s) of the solution, together with the condition $\nabla \cdot \mathbf{u} = 0$. Due to their occurrence in electrostatics, they are often called (perfect) *electric boundary conditions* (EBC), a terminology that we follow in this paper.

A fundamental question is what are simple numerical approaches to approximate the solutions to vector-valued problems with EBC. In [29] an immersed boundary staggered grid approach has been proposed. While convergent, this approach is definitely not a simple or canonical method. A seemingly more natural approach for problems on irregular domains is the standard nodal finite element method (FEM). Interestingly, for the class of problems at hand, nodal FEM can exhibit a Babuška paradox, i.e., for domains with curved boundaries the sequence of FEM approximations can converge to a wrong solution (see Sect. 2.2). Domains with re-entrant corners (see Sect. 2.1) pose additional challenges; however, this last aspect is not the focus of this paper. The convergence of the FEM can be recovered by converting to a mixed FEM formulation, as conducted by the authors in a companion paper. However, the simplicity of nodal FEM does not carry over to mixed FEM.

The objective of this paper is to present an alternative for problems with EBC, namely meshfree finite differences (FD), and to demonstrate that these methods solve the problems at hand effectively and in a comparatively canonical fashion. Meshfree FD are generalizations of traditional grid-based FD that apply to clouds of points without any connectivity between them. The meshfree FD methodology is presented in Sects. 2.3 and 2.4, and its application to the vector Poisson equation with EBC is shown in Sect. 2.5. As we shall demonstrate, a structural advantage of meshfree FD is that the approximation of a general PDE boundary value problem is quite straightforward: any differential operator, whether in the domain's interior or on the boundary, is approximated via a meshfree FD stencil. Thus, the problem

is directly transformed into a finite dimensional system, in which each individual equation corresponds to the governing condition that holds at a particular point of the cloud.

We demonstrate (in Sect. 2.5) that for the vector Poisson equation with EBC, meshfree finite differences do not exhibit the Babuška paradox, and furthermore that there is no conceptual problem to obtain higher-order accuracy (we test the method up to third order convergence).

We then move on to time-dependent problems. First, the meshfree FD method is extended to the vector heat equation with EBC (see Sect. 3.2), where an explicit or an implicit time-stepping can be conducted. Then, by adding nonlinear convective terms and a pressure, the approach is further extended to the incompressible Navier-Stokes equations (NSE). Specifically, we consider a pressure Poisson equation (PPE) reformulation of the Navier-Stokes equations. The idea of PPE reformulations (see Sect. 3) is that an operator function $p = P(\mathbf{u})$ is formulated that yields (via the solution of a Poisson equation) the pressure p to any given velocity field $\mathbf{u}$ that solves the NSE (see also Sect. 3 for a brief discussion of the advantages of PPE reformulations). Here, we focus on a specific PPE reformulation, proposed in [29], which prescribes EBC for the fluid velocity, the motivation for which is outlined in Sect. 3.1. We demonstrate how a meshfree FD approximation for the full PPE reformulation can be constructed (see Sect. 3.3), and show computational results for a resulting numerical scheme that is second order accurate in space and time, and that allows for a choice of an explicit or an implicit treatment of the viscosity (see Sect. 3.4).

2 Vector Poisson Equation

The vector Poisson equation (VPE) arises, for instance, in problems in electrostatics. The electric field satisfies $\nabla \cdot \mathbf{E} = \rho$, where $\rho = \rho(x)$ is the (normalized) charge density. Using the fact that $\nabla \times \mathbf{E} = 0$, this implies the VPE $\Delta \mathbf{E} = \nabla \rho$. Moreover, if the boundaries of the domain are perfect conductors, then the vector field is perpendicular to the boundary, i.e., $\mathbf{n} \times \mathbf{E} = 0$, where $\mathbf{n}$ is the outer surface normal vector. Another example is the magnetic potential, which satisfies the VPE $\Delta \mathbf{A} = -\mathbf{J}$, where $\mathbf{J} = \mathbf{J}(x)$ is the (normalized) electric current density. The Coulomb gauge yields $\nabla \cdot \mathbf{A} = 0$, and the boundary condition $\mathbf{n} \times \mathbf{A} = 0$ represents a zero magnetic field (see [8] for more details). Motivated by the structure of these examples, we here consider the VPE

$$\begin{cases} -\Delta \mathbf{u} = \mathbf{f} & \text{in } \Omega \\ \nabla \cdot \mathbf{u} = 0 & \text{on } \Omega \\ \mathbf{n} \times \mathbf{u} = \mathbf{n} \times \mathbf{g} & \text{on } \partial\Omega \ . \end{cases} \tag{1}$$

As motivated in Sect. 1, it can be desirable to remove the divergence condition that holds in the whole domain. In the following, we outline how this can be achieved.

2.1 Electric Boundary Conditions

Let Ω be a bounded, simply connected domain with Lipschitz boundary $\partial\Omega$. Moreover, in this paper we restrict to domains with boundaries $\partial\Omega$ which are piecewise C^2 and convex (see Remark 1). We denote by $\mathbf{n}$ the outward unit normal vector along the boundary (that is defined almost everywhere). The vector Poisson equation (VPE) with electric boundary conditions (EBC) takes the form

$$\begin{cases} -\Delta\mathbf{u} = \mathbf{f} & \text{in } \Omega \\ \nabla\cdot\mathbf{u} = 0 & \text{on } \partial\Omega \\ \mathbf{n}\times\mathbf{u} = \mathbf{n}\times\mathbf{g} & \text{on } \partial\Omega \end{cases} \tag{2}$$

where the source is incompressible, i.e., $\nabla\cdot\mathbf{f} = 0$. Note that in contrast to problem (1), problem (2) possesses no source-free condition in the domain's interior. Instead, $\nabla\cdot\mathbf{u} = 0$ is specified as an additional boundary condition. Clearly, any solution of (1) is also a solution of (2). Moreover...

Lemma 1. *If the solution to* (2) *is in* $H^2(\Omega)$*, then it is also a solution to* (1)*.*

Proof. Define $\phi = \nabla\cdot\mathbf{u}$. Then ϕ is a (weak) solution of the problem

$$\begin{cases} \Delta\phi = 0 & \text{in } \Omega \\ \phi = 0 & \text{on } \partial\Omega\ , \end{cases}$$

which has the unique solution $\phi \equiv 0$. Hence $\nabla\cdot\mathbf{u} = 0$ in Ω.

Remark 1. As shown in [17], the assumption of Lemma 1 is satisfied for the domains considered in this paper. However, it is not satisfied if the domain Ω has re-entrant (i.e., non-convex) corners. In such a case, the physically relevant (i.e., source-free) solution to (1) is not in H^1, while problem (2) possesses a solution in H^1, however, one that does not satisfy $\nabla\cdot\mathbf{u} = 0$ inside Ω. In this paper we exclude this possibility, and for the domains considered here (see above) the problems (1) and (2) are in fact equivalent (see [17] for a proof).

2.2 Nodal Finite Elements and Babuška Paradox

Among possible approaches to numerically approximate problem (2) on an irregular domain, standard nodal-based finite elements (FE) are one of the first ideas that would come to a numerical analyst's mind. Below, we derive two possible variational formulations (Sect. 2.2.1), and then use these to prove the possibility of the Babuška paradox (Sect. 2.2.2). Its actual occurrence is then demonstrated via an numerical example (Sect. 2.2.3).

2.2.1 Variational Formulations

In order to conduct a FE approximation, a variational formulation of the VPE (2) must be introduced. It is natural to work with the affine Hilbert space of vector-valued H^1 functions that satisfy the tangential boundary condition in (2),

$$H^1_{\mathbf{g}t}(\Omega)^N = \{\mathbf{u} \in H^1(\Omega)^N : \mathbf{n} \times (\mathbf{u} - \mathbf{g})|_{\partial\Omega} = 0\} .$$

Moreover, let $H^1_{0t}(\Omega)^N$ denote the associated homogeneous (i.e., $\mathbf{g} = 0$) Hilbert space. There are then two equivalent weak formulations of (2). To obtain the first formulation, we use the identity $-\Delta\mathbf{u} = \nabla \times (\nabla \times \mathbf{u}) - \nabla(\nabla \cdot \mathbf{u})$ and follow the standard procedure of multiplying the first equation in (2) by a test function $\mathbf{v} \in H_{0t}(\Omega)^N$, integrating by parts, and applying the boundary conditions to the boundary integral to obtain

$$\langle \mathbf{f}, \mathbf{v} \rangle = \int_\Omega -\Delta\mathbf{u} \cdot \mathbf{v} \, \mathrm{d}x = a(\mathbf{u}, \mathbf{v}) - \int_{\partial\Omega} (\nabla \cdot \mathbf{u})(\mathbf{n} \cdot \mathbf{v}) \, \mathrm{d}S , \tag{3}$$

where the bilinear form is

$$a(\mathbf{u}, \mathbf{v}) = \int_\Omega (\nabla \times \mathbf{u}) \cdot (\nabla \times \mathbf{v}) + (\nabla \cdot \mathbf{u})(\nabla \cdot \mathbf{v}) \, \mathrm{d}x .$$

Based on this, the first variational formulation of (2) reads as: Given $\mathbf{f} \in L^2(\Omega)^N$ with $\nabla \cdot \mathbf{f} = 0$, find $\mathbf{u} \in H_{\mathbf{g}t}(\Omega)^N$ such that for each $\mathbf{v} \in H_{0t}(\Omega)^N$

$$\textbf{(VP1)} \qquad a(\mathbf{u}, \mathbf{v}) = \langle \mathbf{f}, \mathbf{v} \rangle .$$

Note that due to (3), the condition $(\nabla \cdot \mathbf{u})_{\partial\Omega} = 0$ arises as a natural boundary condition. It is this formulation **(VP1)** that we implement in the numerical test in Sect. 2.2.3.

In obtaining the second variational formulation, we restrict the derivation to the case $\mathbf{g} = 0$, because this case is enough to show that the Babuška paradox can arise. Assume for a moment that $\mathbf{u} \in H^2(\Omega)^N$. Then, using the fact that $\Delta\mathbf{u} = \nabla \cdot (\nabla\mathbf{u})$, we multiply the left hand side of (2) by $\mathbf{v} \in H_{0t}(\Omega)^N$ and integrate by parts to obtain

$$\langle \mathbf{f}, \mathbf{v} \rangle = \int_\Omega -\Delta\mathbf{u} \cdot \mathbf{v} \, \mathrm{d}x = \int_\Omega \nabla\mathbf{u} \cdot \nabla\mathbf{v} \, \mathrm{d}x - \int_{\partial\Omega} \mathbf{v} \cdot \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{n}} \, \mathrm{d}S . \tag{4}$$

Combining (3) and (4), and using that $\mathbf{n} \times \mathbf{v} = 0$ on $\partial\Omega$, we can rewrite $a(\mathbf{u}, \mathbf{v})$ as a new bilinear form

$$b(\mathbf{u}, \mathbf{v}) = \int_\Omega \nabla\mathbf{u} \cdot \nabla\mathbf{v} \, \mathrm{d}x + \int_{\partial\Omega} (\nabla \cdot \mathbf{u} - \mathbf{n} \cdot \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{n}})(\mathbf{n} \cdot \mathbf{v}) \, \mathrm{d}S .$$

Moreover, since $\mathbf{n}\times\mathbf{u} = 0$ on $\partial\Omega$, we can expand the divergence on the boundary as

$$\nabla\cdot\mathbf{u} = \mathbf{n}\cdot\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{n}} + \kappa\mathbf{n}\cdot\mathbf{u} \qquad \text{on } \partial\Omega\ , \tag{5}$$

where κ is the local curvature which is defined almost everywhere. Using (5), we can write $b(\mathbf{u},\mathbf{v})$ as

$$b(\mathbf{u},\mathbf{v}) = \langle\nabla\mathbf{u},\nabla\mathbf{v}\rangle + \int_{\partial\Omega}\kappa\mathbf{u}\cdot\mathbf{v}\,\mathrm{d}S\ ,$$

thus giving rise to a different variational formulation: Given $\mathbf{f}\in L^2(\Omega)^N$ with $\nabla\cdot\mathbf{f} = 0$, find $\mathbf{u}\in H_{0t}(\Omega)^N$ such that for each $\mathbf{v}\in H_{0t}(\Omega)^N$

(VP2) $\qquad b(\mathbf{u},\mathbf{v}) = \langle\mathbf{f},\mathbf{v}\rangle\ .$

Clearly, by construction the bilinear forms are equal, $a(\mathbf{u},\mathbf{v}) = b(\mathbf{u},\mathbf{v})$, for functions in $H^2(\Omega)^N$. In fact, as shown in [17], the equality also holds if the functions are in $H^1(\Omega)^N$. Moreover, the standard theory shows that the bilinear forms are coercive and continuous on $H_{0t}(\Omega)^N$ (since Ω is simply connected) so that by the Lax-Milgram theorem there is a unique solution to **(VP1)** and **(VP2)**. Moreover the variational problems **(VP1)** and **(VP2)** have the same solution.

2.2.2 Babuška Paradox

Using the just derived weak formulations, we prove the possible occurrence of the Babuška paradox. Note that other proofs have been provided before, such as in [31]. One new aspect in the proof presented here is the isolation of the actual limit problem for the nodal FEM approximation, given below.

Theorem 1. *When solving the vector Poisson equation* (2) *using a nodal FEM implementation of* ***(VP1)****, one may encounter the Babuška paradox.*

Proof. Suppose that $\mathbf{u}_h$ solves **(VP1)** using nodal elements, a triangular mesh, and a regular polygonal domain Ω_h. Here h denotes the diameter of the largest mesh element, so that $\Omega_h\to\Omega$ (in the appropriate sup-norm sense) as $h\to 0$. Now, under the current assumptions on the domain Ω, the problems **(VP1)** and **(VP2)** have the same weak solution. Then by the equivalence of the two problems, $\mathbf{u}_h$ also solves **(VP2)**. For any given mesh, however, the boundary of Ω_h has flat sides with $\kappa = 0$. Consequently, the weak solution $\mathbf{u}_h$ solves **(VP2)** with $\kappa = 0$, i.e.,

$$\int_{\Omega_h}\nabla\mathbf{u}\cdot\nabla\mathbf{v}\,\mathrm{d}x = \int_{\Omega_h}\mathbf{f}\cdot\mathbf{v}\,\mathrm{d}x$$

for each test function $\mathbf{v}_h$. Therefore the solutions $\mathbf{u}_h$ converge to the function $\mathbf{u}^*$ which solves the limit problem

$$\int_{\Omega} \nabla \mathbf{u}^* \cdot \nabla \mathbf{v} \, \mathrm{d}x = \langle \mathbf{f}, \mathbf{v} \rangle$$

for each $\mathbf{v} \in H_{0t}(\Omega)^N$. In other words, the nodal FEM solutions $\mathbf{u}_h$ converge to a solution where κ is artificially set to zero, or equivalently to a problem where one replaces the boundary condition $\nabla \cdot \mathbf{u} = 0$ with $\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{n}} = 0$. Hence, for an arbitrary domain (with boundaries that are at least partially curved), generally $\mathbf{u}^*$ does not equal the true solution of the problem **(VP1)**.

2.2.3 Manufactured Solution Test Case

We now demonstrate the Babuška paradox via a numerical example. On the 2D domain $\Omega = \{(x-0.5)^2 + (y-0.5)^2 < 0.5^2\} \cup (0,1) \times (0,0.5)$, we consider the VPE (2). We employ the method of manufactured solutions, i.e., we prescribe the incompressible solution

$$\mathbf{u}(x,y) = \begin{pmatrix} \pi \sin(2\pi y) \sin^2(\pi x) \\ -\pi \sin(2\pi x) \sin^2(\pi y) \end{pmatrix}, \tag{6}$$

and set the forcing $\mathbf{f} = -\Delta \mathbf{u}$ and the boundary velocity $\mathbf{g} = \mathbf{u}$, so that the solution of the VPE (2) recovers the prescribed solution (6).

We consider two triangulations (of different resolutions) that approximate the domain Ω via polygons with straight edges. To define normal vectors at the boundary vertices, we adopt the method introduced in [9], which obtains normal vectors as suitable averages of normal vectors at the edges connecting to the boundary vertex. Using standard nodal-based finite elements, or Lagrange FE [6], we implement the tangential boundary condition in an essential fashion (by choosing the solution space $H^1_{\mathbf{g}t}(\Omega)^N$), and leave the normal component unprescribed, with the idea that $\nabla \cdot \mathbf{u} = 0$ follows naturally (see above). We use quadratic C^0 elements.

The results of the two FEM approximations are shown in Fig. 1. The true vector field (6) is shown by red arrows, and the approximate FEM solution is given by black arrows. In each case, the mesh that is used to conduct the respective computation is shown in the background. The right panel shows a computation with a mesh that is twice as fine as the one in the left panel. As one can see, the relevant features of the prescribed solution (6) are well-resolved already on the coarse mesh. Moreover, the two numerical solutions (on the two meshes) are almost identical, and hence they can be interpreted as converged (in the eye-norm) to the limit ($h \to 0$) solution $\mathbf{u}^*$ of the nodal FEM. Clearly the FEM solution $\mathbf{u}^*$ is different from the true solution $\mathbf{u}$, thus confirming (and certainly visualizing) the occurrence of the Babuška paradox.

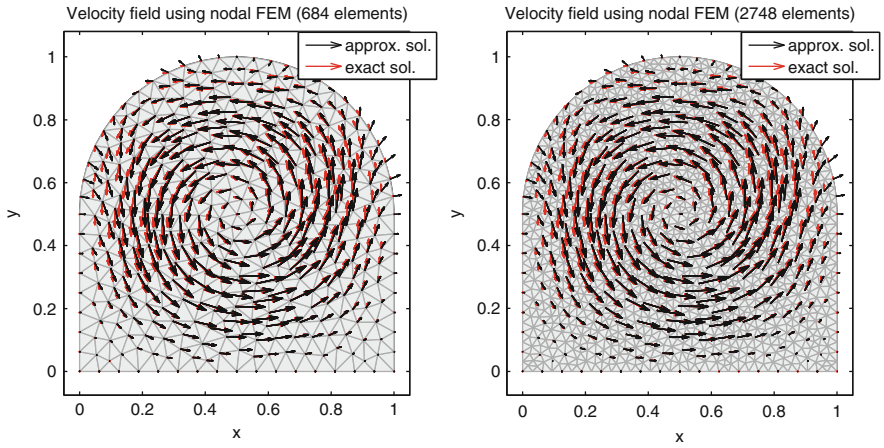


Fig. 1 True vector field (*red arrows*) and numerical approximations (*black arrows*) obtained via nodal FEM. The numerical solutions obtained with two different mesh resolutions are almost identical, thus the numerics are essentially converged. Yet, the FEM solution differs from the true solution

Remark 2. Within the framework of FEM, the Babuška paradox can be overcome by moving to a mixed FEM formulation (cf. [1, 2, 26, 31]). The idea of mixed FEM for the VPE (2) is introduce an additional variable, $\sigma = \nabla \times \mathbf{u}$, and transform $-\Delta \mathbf{u} = \mathbf{f}$ into two equations: $\sigma = \nabla \times \mathbf{u}$ and $\nabla \times \sigma - \nabla(\nabla \cdot \mathbf{u}) = \mathbf{f}$. This framework allows one to use Raviart-Thomas elements [26] for the approximate vector field $\mathbf{u}^h$ (and standard nodal elements (2D) or Nédélec elements [23] (3D) for σ^h), and to incorporate the tangential velocity boundary condition $\mathbf{n} \times (\mathbf{u} - \mathbf{g}) = 0$ as boundary integrals into the weak formulation, rather than into the solution space. In a companion paper, we apply high-order mixed FEM to the pressure Poisson equation reformulations of the Navier-Stokes equations devised in [29]. While mixed FEM overcome the Babuška paradox, this framework is clearly not as simple as nodal FEM, or as meshfree FD, described below.

2.3 *Meshfree Finite Difference Method*

Meshfree finite differences (FD) generalize classical FD that are defined on regular grids: at a given point, a differential operator of a smooth function is approximated via a combination of function values at nearby points. The selection of points and the corresponding weights are called the *stencil*. In the same way as grid-based FD, meshfree FD can be derived in two ways: as derivatives of suitable local interpolants of the data (cf. [7, 20, 21]), or via Taylor expansion of the solution (cf. [28]). Here we outline the second methodology.

Consider a *point cloud* that consists of interior points (inside Ω) and boundary points (on $\partial\Omega$); see Fig. 2 for an example. For a point x_i, let a neighborhood B_i be defined. Here we employ circular neighborhoods, i.e., $B_i = \{j : \|x_j - x_i\| \leq r\}$, where r is an appropriately chosen radius (see below). However, many other types of neighborhoods are possible [27]. Now define the relative coordinates $\bar{x}_{ij} = x_j - x_i$ and Taylor-expand the solution $u(x)$ around x_i:

$$u(x_j) = u(x_i) + \nabla u(x_i) \cdot \bar{x}_{ij} + \tfrac{1}{2}\nabla^2 u(x_i) : (\bar{x}_{ij} \cdot \bar{x}_{ij}^T) + \text{h.o.t.}$$

Note that in the quadratic term, the matrix scalar product $A : C = \sum_{i,j} A_{ij} C_{ij}$ and the outer product of $\bar{x}_{ij}$ with itself are used. While here we stop at the quadratic term, the expansion can of course be carried out further (or less far). A linear combination (with weights a_{ij}) of nearby solution values yields

$$\begin{aligned} \sum_{j \in B_i} a_{ij} u(x_j) = u(x_i) \sum_{j \in B_i} a_{ij} + \nabla u(x_i) \cdot \sum_{j \in B_i} a_{ij} \bar{x}_{ij} \\ + \nabla^2 u(x_i) : \tfrac{1}{2} \sum_{j \in B_i} a_{ij} (\bar{x}_{ij} \cdot \bar{x}_{ij}^T) + \text{h.o.t.} \end{aligned} \tag{7}$$

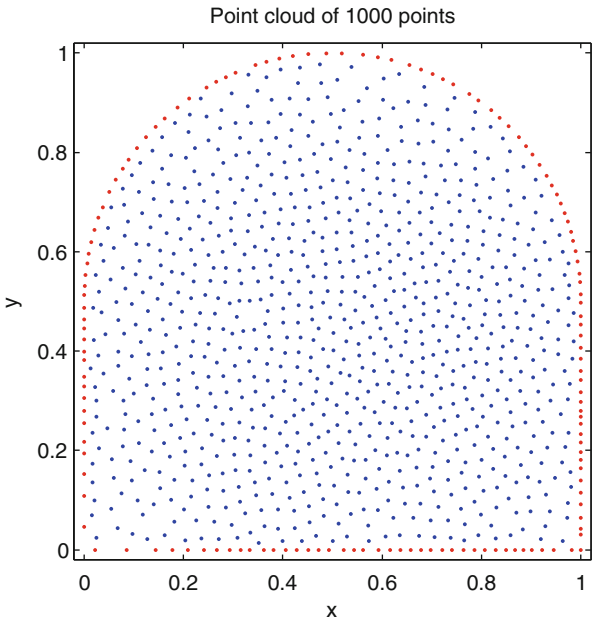


Fig. 2 Point cloud for the computational domain Ω with 1,000 points. The boundary points are shown in *red* and the interior points in *blue*

If (7) is supposed to approximate a given differential operator applied to the solution, then the stencil weights a_{ij} must satisfy certain constraints. For instance, for (7) to approximate $\Delta u(x_i)$, it is required that

$$\sum_{j \in B_i} a_{ij} = 0 \,, \quad \sum_{j \in B_i} \bar{x}_{ij} a_{ij} = 0 \,, \text{ and } \quad \sum_{j \in B_i} (\bar{x}_{ij} \cdot \bar{x}_{ij}^T) a_{ij} = 2I \,,$$

which in 2D gives rise to the linear system of constraints

$$\underbrace{\begin{pmatrix} \bar{x}_{i,j_1} & \dots & \bar{x}_{i,j_{m_i}} \\ \bar{y}_{i,j_1} & \dots & \bar{y}_{i,j_{m_i}} \\ \bar{x}_{i,j_1}^2 & \dots & \bar{x}_{i,j_{m_i}}^2 \\ \bar{x}_{i,j_1}\bar{y}_{i,j_1} & \dots & \bar{x}_{i,j_{m_i}}\bar{y}_{i,j_{m_i}} \\ \bar{y}_{i,j_1}^2 & \dots & \bar{y}_{i,j_{m_i}}^2 \end{pmatrix}}_{V_i} \cdot \underbrace{\begin{pmatrix} a_{i,j_1} \\ \vdots \\ \vdots \\ a_{i,j_{m_i}} \end{pmatrix}}_{\mathbf{a}_i} = \underbrace{\begin{pmatrix} 0 \\ 0 \\ 2 \\ 0 \\ 2 \end{pmatrix}}_{\mathbf{b}} \,. \tag{8}$$

Here, the stencil vector $\mathbf{a}_i$ does not contain the diagonal entry a_{ii}. Its value is obtained as $a_{ii} = -\sum_{j \in B_i \setminus \{i\}} a_{ij}$. Moreover, $m_i = |B_i| - 1$ is the number of neighbors of x_i. If the radius r is chosen large enough that $m_i \geq 5 \;\; \forall\, i$, and if the point cloud generation (see Sect. 2.4) ensures that no pathological point configurations arise (see [27] for examples), then system (8) always has a solution, and the resulting approximation is (at least) first order accurate.

If $m_i > 5$, system (8) in general has infinitely many solutions. One way (employed here) to single out a unique solution is via a weighted least-squares (WLSQ) minimization problem

$$\min \sum_{j \in B_i \setminus \{i\}} \frac{a_{ij}^2}{w_{ij}} \,, \text{ s.t. } V_i \cdot \mathbf{a}_i = \mathbf{b} \tag{9}$$

where the weights are decreasing with the distance, $w_{ij} = \|x_j - x_i\|_2^{-\beta}$ (here we choose $\beta = 2$). The solution of (9) is

$$\mathbf{a}_i = W_i V_i^T (V_i W_i V_i^T)^{-1} \cdot \mathbf{b} \,.$$

where $W = \text{diag}(w_{i,1}, \dots, w_{i,j_{m_i}})$. Note that an alternative approach (not employed here) would be to solve system (8) in an ℓ^1 sense, i.e.,

$$\min \sum_{j \in B_i \setminus \{i\}} \frac{a_{ij}}{w_{ij}} \,, \text{ s.t. } V_i \cdot \mathbf{a}_i = \mathbf{b} \,, \; \mathbf{a}_i \geq 0 \,,$$

which would generate optimally sparse stencils [28].

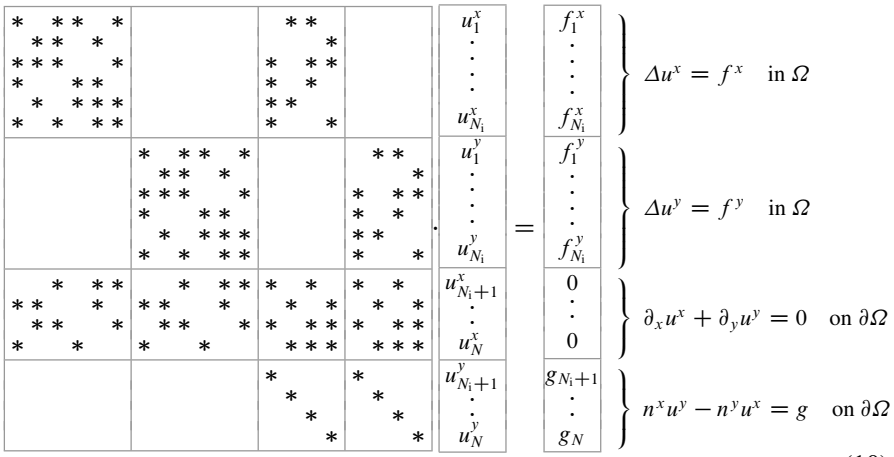
Other differential operators are approximated in an analogous fashion. For instance, a first order approximation to $\partial_x u$ is obtained by setting

$$V_i = \begin{pmatrix} \bar{x}_{i,j_1} & \dots & \bar{x}_{i,j_{m_i}} \\ \bar{y}_{i,j_1} & \dots & \bar{y}_{i,j_{m_i}} \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

and a second order approximation to $\partial_x u$ is obtained by setting

$$V_i = \begin{pmatrix} \bar{x}_{i,j_1} & \dots & \bar{x}_{i,j_{m_i}} \\ \bar{y}_{i,j_1} & \dots & \bar{y}_{i,j_{m_i}} \\ \bar{x}^2_{i,j_1} & \dots & \bar{x}^2_{i,j_{m_i}} \\ \bar{x}_{i,j_1}\bar{y}_{i,j_1} & \dots & \bar{x}_{i,j_{m_i}}\bar{y}_{i,j_{m_i}} \\ \bar{y}^2_{i,j_1} & \dots & \bar{y}^2_{i,j_{m_i}} \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Applying this procedure to each equation (at all interior points) and boundary condition (at all boundary points) of the vector Poisson equation (2) leads (here is 2D) to the linear system



(10)

In this system, $\mathbf{u} = (u^x, u^y)$, $\mathbf{f} = (f^x, f^y)$, and the function $g = \mathbf{n} \times \mathbf{g}$ at the boundary, and $g_i = \mathbf{n}_i \times \mathbf{g}_i$. Moreover, the total number of points is N, and the number of interior points is N_i. In the sparse block matrix, the first two rows of blocks correspond to the two components of the Poisson equation at the interior points; the third block row encodes the $\nabla \cdot \mathbf{u} = 0$ boundary condition; and the fourth block row represents the $\mathbf{n} \times (\mathbf{u} - \mathbf{g}) = 0$ condition. The first two block columns corresponds to the two vector field components at the interior points $x_1, \dots, x_{N_\mathrm{i}}$;

and likewise the last two block columns correspond to $\mathbf{u} = (u^x, u^y)$ at the boundary points $x_{N_i}, \ldots, x_N$. Each empty block is devoid of nonzero entries, and each block with stars possesses some nonzero entries, corresponding to the neighboring points of each central point.

2.4 Point Cloud Generation

The generation of point clouds is conducted similarly to the mesh generation code DistMesh [24]. The domain Ω is specified via a level set function ϕ, chosen so that $\Omega = \{x : \phi(x) < 0\}$. Hence, ϕ also defines the boundary $\partial\Omega = \{x : \phi(x) = 0\}$ and surface normal vectors $\mathbf{n} = \nabla\phi/|\nabla\phi|$. The access to the level set function allows an immediate check whether a given location/point is inside or outside of the domain. Moreover, if ϕ is a signed distance function (i.e., $|\nabla\phi| = 1$ a.e.), one even has immediate access to a location's distance to the domain boundary.

The generation of a cloud of N points is initiated by placing N points randomly inside Ω. After that, the points are moved according to repulsive forces experienced from nearby points, i.e., the point x_i moves according to

$$\dot{x}_i = \sum_{j \in B_i \setminus \{i\}} \min\{\|x_j - x_i\|_2^{-2}, v_{\max}\}\,, \tag{11}$$

where B_i is a circular neighborhood around x_i, and $v_{\max}$ is some upper bound on the repulsion. The law of motion (11) is further constrained by $x_i \in \bar{\Omega}$, i.e., no point can ever move beyond the domain boundary to leave the domain. A simple way to implement this constraint is to allow points to slightly leave $\bar{\Omega}$, but then to immediately project them back onto $\partial\Omega$, using the normal $\nabla\phi/|\nabla\phi|$ that is defined also outside of $\bar{\Omega}$. Finally, points that are inside Ω but too close to the boundary are also projected onto $\partial\Omega$, thus preventing interior points from being too close to the boundary (see [28] for why this would be undesirable). The law of motion (11) is then applied to all points until the amount of motion has fallen below a given threshold. The resulting point clouds are unstructured, and tend to be quite uniform (i.e., the ratio between the minimum distance between points and the radius of the largest ball that contains no points (cf. [20]) is quite large). An example of a point cloud associated with the domain defined in Sect. 2.2.3 is shown in Fig. 2.

For the definition of a "mesh" resolution h of a given point cloud, a variety of possible choices exists [20]. In fact, the point cloud generation algorithm presented above ensures suitable regularity of the point clouds, so that as $N \to \infty$, the mesh size (as defined in [20]) goes to zero, while the separation approaches a positive value. Therefore, for the purpose of quantifying the resolution of the point cloud, here we use a simple averaged concept of resolution, defined as follows. One type of configuration in which nearby points are equidistant is an optimal sphere packing,

which in 2D is a hexagonal lattice, i.e., adjacent points form equilateral triangles. Each point is a corner of six triangles, and each triangle is shared by three points. One can therefore associate to each point 1/3 of each of the six triangles, resulting in an area per point of $A = \frac{\sqrt{3}}{2}h^2$, if the spacing between points is h. Counting the area of the N_i interior points full, and of the N_b boundary points half, and equating the total "point area" with the area of the domain, $\lambda(\Omega)$, we obtain the expression $h = \sqrt{\frac{4\lambda(\Omega)}{\sqrt{3}(2N_\mathrm{i}+N_\mathrm{b})}}$ for the resolution.

2.5 Numerical Results

We consider the same manufactured solution test problem as studied in Sect. 2.2.3. On the domain $\Omega = \{(x-0.5)^2+(y-0.5)^2 < 0.5^2\} \cup (0,1)\times(0,0.5)$ (see Figs. 1 and 2), point clouds of various numbers of points are generated, so that convergence studies can be conducted. For each point cloud, the vector Poisson equation (2) is discretized via the procedure described in Sect. 2.3. We conduct the meshfree FD approximation for three different orders: first order accuracy, i.e., the Taylor expansion in (7) is carried out up to the quadratic term for $\Delta\mathbf{u}$ and up to the linear term for $\nabla\cdot\mathbf{u}$; and second and third order accuracies, for which the Taylor expansions in (7) are carried out further accordingly.

The numerical approximations obtained for the different orders and mesh resolutions are then compared to the true solution in the maximum norm, taken over all points. We consider the errors in the vector field $\mathbf{u}$ itself, as well as its Jacobian $\nabla\mathbf{u}$ (which is important for calculating stresses at the boundary when $\mathbf{u}$ represents a velocity field) and its divergence $\nabla\cdot\mathbf{u}$ (which by the equivalence of problems (1) and (2) should be close to zero). All derivative quantities are obtained from the vector field $\mathbf{u}$ via meshfree FD stencils of fourth order. Hence, if an order of less than 4 is observed, we know that this is the true accuracy of the numerical result.

The error convergence of these quantities is shown in Fig. 3, for the approximation orders 1 (left panel), 2 (middle panel), and 3 (right panel). The results show that all approaches converge as $h \to 0$, and the convergence orders equal the local approximation orders. In particular, the convergence orders of the derivative quantities are the same as those of the vector field itself. This is an important property that finite difference methods commonly exhibit, and that is in contrast to finite element methods that frequently lose an order of accuracy when a derivative quantity is evaluated (in the sense of functions).

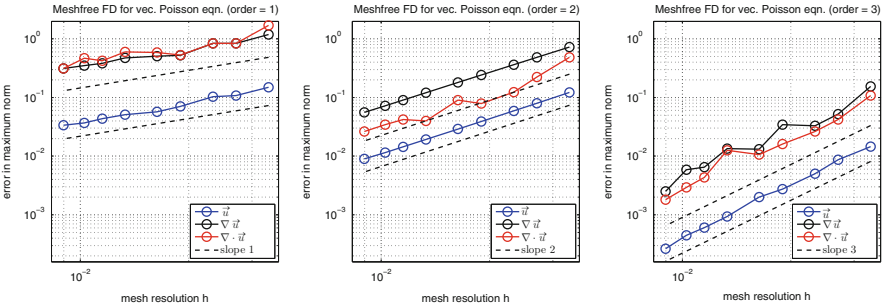


Fig. 3 Error convergence for first (*left*), second (*middle*), and third (*right*) order meshfree FD approximations. The errors are measured in the maximum norm. The results show that kth order meshfree FD stencils result in kth order convergence rates for the solution of the VPE, and its derivatives

3 Pressure Poisson Equation Reformulation of the Navier-Stokes Equations

We consider the time-dependent incompressible Navier-Stokes equations (NSE)

$$\begin{cases} \partial_t \mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\nabla p + \nu\Delta\mathbf{u} + \mathbf{f} & \text{in } \Omega\times(0,T) \\ \nabla\cdot\mathbf{u} = 0 & \text{in } \Omega\times(0,T) \\ \mathbf{u} = \mathbf{g} & \text{on } \partial\Omega\times(0,T) \\ \mathbf{u} = \mathring{\mathbf{u}} & \text{on } \Omega\times\{t=0\}\,, \end{cases} \tag{12}$$

on a domain Ω as specified in Sect. 2.1, with compatibility conditions

$$\mathring{\mathbf{u}} = \mathbf{g} \quad \text{on } \partial\Omega\times\{t=0\} \qquad \text{(continuity between i.c. and b.c.)}$$

$$\nabla\cdot\mathring{\mathbf{u}} = 0 \quad \text{in } \Omega \qquad \text{(incompressible i.c.)}$$

$$\int_{\partial\Omega} \mathbf{n}\cdot\mathbf{g}\,\mathrm{d}x = 0\,. \qquad \text{(inflow = outflow)}$$

Due to the lack of a time evolution of the pressure, there is no single canonical way to numerically advance (12) forward in time. One class of approaches to do so is based on approximating the time derivative in the momentum equation, and solving for $\mathbf{u}$ and p in a fully coupled fashion. This methodology is accurate, but also costly, because a large system must be solved that possesses a saddle point structure. An alternative class of approaches decouples the pressure solve from the velocity update. This methodology was first proposed in the form of projection methods [5,30], and later employed in approaches based on pressure Poisson equation (PPE) reformulations of the NSE [12–14,16,29].

A fundamental difference between projection methods (see [11] for an overview) and PPE reformulations is that projection methods are based on a fractional step

approach in which the time-evolution of the velocity field and its projection onto the space of incompressible fields are alternated. In contrast, PPE reformulations derive an equation for the pressure that replaces the incompressibility constraint in (12) by a global pressure function $p = P(\mathbf{u})$ that is designed so that the solutions of the PPE reformulation are identical to the solutions of the original NSE. As a consequence, numerical methods based on PPE reformulations are structurally easy to extend to high order accuracy in time. In addition, they do not suffer from poor spatial accuracy near boundaries (see [29]). A difficulty of PPE reformulations is that the Poisson equation for the pressure can involve complicated expressions, whose interaction with the velocity field equation is not always easy to understand and analyze. Another important property of PPE reformulations is that, unlike the original NSE, they are also defined if the initial conditions are not incompressible, see Sect. 3.1.

3.1 PPE Reformulation with Electric Boundary Conditions

In this paper, we are concerned with the particular PPE reformulation of the NSE proposed in [29]. Its fundamental difference from previously proposed PPE reformulations [12,13,16] is that the velocity field satisfies electric boundary conditions, i.e., incompressibility and the tangential flow are prescribed at the boundary. In turn, the normal velocity is enforced via a relaxation term in the pressure equation (see [29] for a discussion on the choice of λ). The PPE reformulation consists of the momentum equation

$$\begin{cases} \partial_t \mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\nabla P(\mathbf{u}) + \nu\Delta\mathbf{u} + \mathbf{f} & \text{in } \Omega\times(0,T) \\ \nabla\cdot\mathbf{u} = 0 & \text{on } \partial\Omega\times(0,T) \\ \mathbf{n}\times\mathbf{u} = \mathbf{n}\times\mathbf{g} & \text{on } \partial\Omega\times(0,T) \\ \mathbf{u} = \mathring{\mathbf{u}} & \text{on } \Omega\times\{t=0\}\,, \end{cases} \tag{13}$$

where $P(\mathbf{u})$ is a solution of the associated pressure Poisson equation

$$\begin{cases} \Delta p = \nabla\cdot(\mathbf{f} - (\mathbf{u}\cdot\nabla)\mathbf{u}) & \text{in } \Omega \\ \frac{\partial p}{\partial \mathbf{n}} = \mathbf{n}\cdot(\mathbf{f} - \partial_t\mathbf{g} + \nu\Delta\mathbf{u} - (\mathbf{u}\cdot\nabla)\mathbf{u}) + \lambda\mathbf{n}\cdot(\mathbf{u}-\mathbf{g}) & \text{on } \partial\Omega\,. \end{cases} \tag{14}$$

While a variety of modifications and additions can be applied to these equations (cf. [29,32]), here we study the equations exactly in the given form (with one small caveat regarding the compatibility of the pressure boundary conditions, see below).

If the initial conditions are not incompressible, i.e., $\nabla\cdot\mathring{\mathbf{u}} \neq 0$, then the solution of (13) relaxes towards a solutions of (12), for the following reason. Let $\phi = \nabla\cdot\mathbf{u}$. Then the application of $\nabla\cdot$ to the momentum equation in (12), and the use of the first equation in (14) yields that ϕ satisfies the heat equation with homogeneous Dirichlet boundary conditions

$$\begin{cases} \partial_t \phi = \nu \Delta \phi & \text{in } \Omega \times (0,T) \\ \phi = 0 & \text{on } \partial\Omega \times (0,T) \\ \phi = \nabla \cdot \mathring{\mathbf{u}} & \text{on } \Omega \times \{t=0\}\,. \end{cases} \tag{15}$$

This property is of great relevance. It means that in PPE reformulations, there is no need to impose a discrete incompressibility principle. If, due to numerical approximation errors, the numerical solution starts to drift away from the $\nabla \cdot \mathbf{u} = 0$ manifold, Eq. (15) ensures that it is pulled back towards incompressibility.

At the same time, the fact that the numerical solution may not be exactly incompressible, implies that the compatibility condition in the pressure Poisson equation (14) may be violated. Specifically, (14) has a solution if

$$\int_\Omega (\lambda + \nu\Delta)\phi \,\mathrm{d}x - \int_{\partial\Omega} (\partial_t + \lambda)\mathbf{g}\cdot\mathbf{n}\,\mathrm{d}S = 0\,, \tag{16}$$

and due to numerical approximation errors (or because a problem with $\nabla \cdot \mathring{\mathbf{u}} \neq 0$ is considered) this condition may be violated. However, whenever this occurs, the solution of the augmented system (22), described in Sect. 3.3 projects the right hand side of (14) in a way that the solvability condition is satisfied.

Below, we first generalize the meshfree finite difference methods developed in Sect. 2 for the vector Poisson equation to the vector heat equation (Sect. 3.2). Then, we extend the methodology to the PPE reformulation (Sect. 3.3).

3.2 Meshfree Finite Differences for the Vector Heat Equation

Before moving to the full PPE reformulation (13), we first generalize the numerical scheme developed in Sect. 2.3 for the vector Poisson equation (2) to the vector heat equation (VHE) that describes the evolution of a vector field $\mathbf{u}(x,t)$ via the system

$$\begin{cases} \partial_t \mathbf{u} = \nu\Delta\mathbf{u} + \mathbf{f} & \text{in } \Omega \times (0,T) \\ \nabla \cdot \mathbf{u} = 0 & \text{on } \partial\Omega \times (0,T) \\ \mathbf{n} \times \mathbf{u} = \mathbf{n} \times \mathbf{g} & \text{on } \partial\Omega \times (0,T) \\ \mathbf{u} = \mathring{\mathbf{u}} & \text{on } \Omega \times \{t=0\}\,. \end{cases} \tag{17}$$

Here the forcing $\mathbf{f}(x,t)$ and the initial data $\mathring{\mathbf{u}}(x)$ are incompressible, i.e., $\nabla \cdot \mathbf{f} = 0$ and $\nabla \cdot \mathring{\mathbf{u}} = 0$. We discretize (17) in time via ImEx (Implicit-Explicit) schemes [3]. Specifically, we use the first-order scheme

$$\tfrac{1}{\Delta t}\left(\mathbf{u}(t+\Delta t) - \mathbf{u}(t)\right) = \mathbf{R}(\mathbf{u}(t+\Delta t)) + \mathbf{Q}(\mathbf{u}(t)) \tag{18}$$

and the second-order two-stage Runge-Kutta scheme

$$\begin{aligned} \tfrac{1}{\Delta t}\left(\mathbf{u}^* - \mathbf{u}(t)\right) &= \gamma \mathbf{R}(\mathbf{u}^*) + \gamma \mathbf{Q}(\mathbf{u}(t)) \\ \tfrac{1}{\Delta t}\left(\mathbf{u}(t+\Delta t) - \mathbf{u}(t)\right) &= \gamma \mathbf{R}(\mathbf{u}(t+\Delta t)) + (1-\gamma)\mathbf{R}(\mathbf{u}^*) \\ &\quad + \delta \mathbf{Q}(\mathbf{u}(t)) + (1-\delta)\mathbf{Q}(\mathbf{u}^*)\,, \end{aligned} \tag{19}$$

where $\gamma = 1 - \frac{1}{2}\sqrt{2}$ and $\delta = 1 - \frac{1}{2\gamma}$. For the VHE (17), an explicit first-order scheme (forward Euler) is obtained by setting $\mathbf{R}(\mathbf{u}) = 0$ and $\mathbf{Q}(\mathbf{u}) = \nu\Delta\mathbf{u} + \mathbf{f}$ in (18). In turn, setting $\mathbf{R}(\mathbf{u}) = \nu\Delta\mathbf{u}$ and $\mathbf{Q}(\mathbf{u}) = \mathbf{f}$ yields semi-implicit schemes of first order via (18), and of second order via (19). Moreover, the boundary conditions are always treated implicitly, so that they are satisfied by the new state (at the end of each Runge-Kutta stage).

The schemes that treat $\Delta\mathbf{u}$ implicitly lead to relatively simple modifications of the linear system (10) of the VPE (17). For instance, the first-order scheme,

$$\begin{cases} \frac{1}{\Delta t}\left(\mathbf{u}(t+\Delta t) - \mathbf{u}(t)\right) = \nu\Delta\mathbf{u}(t+\Delta t) + \mathbf{f}(t) & \text{in } \Omega \\ \nabla\cdot\mathbf{u}(t+\Delta t) = 0 & \text{on } \partial\Omega \\ \mathbf{n}\times\mathbf{u}(t+\Delta t) = \mathbf{n}\times\mathbf{g}(t+\Delta t) & \text{on } \partial\Omega \\ \mathbf{u}(0) = \mathring{\mathbf{u}} & \text{in } \Omega\,, \end{cases}$$

amounts to the following modifications of system (10): (i) The vector of unknowns becomes the new vector field at time $t + \Delta t$. (ii) In the right hand side vector, the function g is evaluated at time $t + \Delta t$. (iii) In the top two block rows of the system matrix, multiply all entries by ν and add $1/\Delta t$ to the diagonal entries. (iv) Add $1/\Delta t$ times the solution at time t to the top two blocks of the right hand side vector. Everything else remains unchanged.

In the forward Euler case, the update at the interior points becomes explicit, while the boundary conditions are still implicit. Hence, to advance the solution from time t to $t + \Delta t$, one first updates at each interior point explicitly

$$\mathbf{u}(x_i, t+\Delta t) = \mathbf{u}(x_i, t) + \Delta t\, \nu \sum_{j\in B_i} a_{ij}\mathbf{u}(x_j, t) + \Delta t\, \mathbf{f}(x_i, t)$$

according to (7) and (9), and after that solves a small linear system for the boundary points that results from the two bottom block rows of system (10), where the left two block columns are brought to the right hand side (using the just updated interior point values).

To study these numerical schemes, we generalize the manufactured solution from Sect. 2.2.3 to the time-dependent case. We set $\nu = 1$, prescribe the incompressible solution

$$\mathbf{u}(x, y, t) = \begin{pmatrix} \pi\cos(t)\sin(2\pi y)\sin^2(\pi x) \\ -\pi\cos(t)\sin(2\pi x)\sin^2(\pi y) \end{pmatrix}, \tag{20}$$

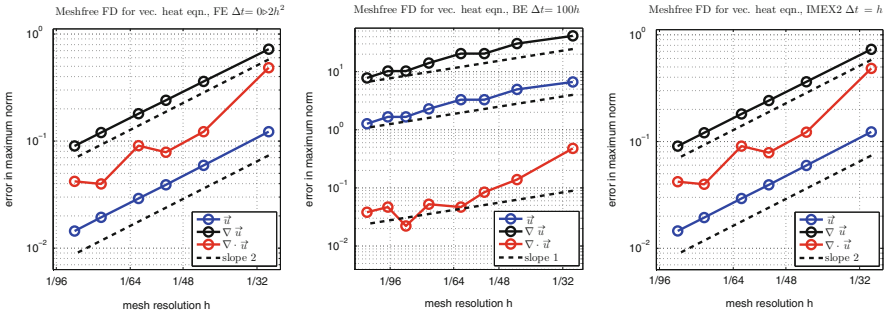


Fig. 4 Error convergence for the vector heat equation, using a spatially second order meshfree finite difference discretization. *Left panel*: using forward Euler time stepping with $\Delta t = 0.2h^2$, confirming the expected $O(h^2)$ convergence order. *Middle panel*: using backward Euler time stepping with $\Delta t = 100h$, yielding the expected $O(h)$ convergence due to temporal errors. *Right panel*: using a second-order ImEx scheme with $\Delta t = h$, confirming the expected $O(h^2)$ convergence order

and calculate the forcing $\mathbf{f} = \partial_t \mathbf{u} - \nu \Delta \mathbf{u}$, the boundary velocity $\mathbf{g} = \mathbf{u}$, and the initial conditions $\mathring{\mathbf{u}} = \mathbf{u}(t = 0)$ accordingly. Using this test case, we first determine, via numerical experiments, the maximum time step that the forward Euler scheme admits to be stable. We find $\Delta t \leq \frac{Ch^2}{\nu}$, where for the given point clouds, C always lies between 0.2 and 0.3. Then, we study the convergence orders of the numerical schemes. The results are shown in Fig. 4. We use a second order meshfree FD approximation in space, and conduct five kinds of time stepping: forward Euler and backward Euler with $\Delta t = 0.2h^2$ (left panel; the plots of forward vs. backward Euler are indistinguishable); backward Euler with $\Delta t = h$ (not shown; due to very small temporal errors, the convergence looks like second order); backward Euler with $\Delta t = 100h$ (middle panel; the temporal error is visible and yields the expected drop to first order); and the second-order ImEx scheme (19) (right panel). From these results, we can see that the solution, and its derivatives, are in fact second-order accurate in space. Moreover, explicit and implicit time stepping (of first and second order) can be conducted without problems; and the temporal errors are relatively small.

3.3 Meshfree Finite Differences for the PPE Reformulation

Structurally the PPE reformulation (13) is the same as the vector heat equation (17), "just" with the nonlinear term $\mathbf{N}(\mathbf{u}) = (\mathbf{u} \cdot \nabla)\mathbf{u}$ and the pressure term $\nabla P(\mathbf{u})$ added to the time evolution. We treat both of these terms, as well as the forcing, explicitly. The first-order ImEx time-stepping (18) gives rise to the update rule

$$\tfrac{1}{\Delta t}\left(\mathbf{u}(t + \Delta t) - \mathbf{u}(t)\right) = -\mathbf{N}(\mathbf{u}(t)) - \nabla P(\mathbf{u}(t)) + \nu \Delta \mathbf{u}(t + \theta \Delta t) + \mathbf{f}(t)\ , \qquad (21)$$

where $\theta \in \{0, 1\}$ allows to switch between an explicit/implicit treatment of viscosity. In the forward Euler case ($\theta = 0$), stability requires $\Delta t = O(h^2)$, and thus the scheme's accuracy is $O(h^2)$ overall. In the semi-implicit case ($\theta = 1$), one can choose larger time steps, and consequently the temporal accuracy is not sufficient. We therefore use instead the second-order ImEx time-stepping (19) with $\mathbf{R}(\mathbf{u}) = \nu \Delta \mathbf{u}$ and $\mathbf{Q}(\mathbf{u}) = \mathbf{f} - \mathbf{N}(\mathbf{u}) - \nabla P(\mathbf{u})$, which allows for time steps $\Delta t = O(h)$ and yields an $O(h^2)$ accurate scheme.

The Jacobi matrix $\nabla \mathbf{u}$ needed for the nonlinear terms is approximated very simply via point-centered meshfree finite differences, via the methodology described in Sect. 2.3. Clearly, such a centered treatment of advection is not the most effective choice for high Reynolds numbers (i.e., $\nu \ll 1$). And in fact, meshfree FD are quite easily amenable to an upwind treatment (e.g., by centering the approximation around a position $x_i - \beta \mathbf{u}(x_i)$, where β is a suitably chosen parameter). However, for the purpose of demonstrating the convergence of meshfree FD methods for the PPE reformulation (13), the central treatment of $\mathbf{N}(\mathbf{u})$ is sufficient.

The pressure $P(\mathbf{u})$ results from the solution of the pressure Poisson equation (14). We discretize this problem via the same meshfree FD method described in Sect. 2.3, with one important deviation from the standard procedure. The right hand side of the boundary conditions in (14) requires the evaluation of $\Delta \mathbf{u}$ at the boundary $\partial \Omega$. While straightforward meshfree FD for $\Delta \mathbf{u}$ yield accurate approximations inside the domain, it turns out that low accuracy (in the form of bounded but noticeable spatial oscillations along $\partial \Omega$) is achieved when using the same procedure at a boundary point. The reason is that the Laplacian is an operator that naturally "likes" to use data around the approximation point (cf. [28]); however, at the boundary, data in such a configuration is not accessible. We therefore employ a different approach that remedies the problem: we use the meshfree approximation of $\mathbf{w} = \Delta \mathbf{u}$ at the interior points (as calculated for the viscosity), and extrapolate this field $\mathbf{w}$ to the boundary points, using moving least squares (MLS) interpolation [18]. This aspect is visualized in Fig. 5: the black dots are the approximation errors when approximating $\Delta \mathbf{u}$ at $\partial \Omega$ via meshfree FD; the red dots are the errors obtained when using MLS interpolation.

Since (14) is a Neumann problem, its discretization leads to a linear system $A \cdot p = r$, in which the Poisson matrix A has corank 1. In fact, because in the meshfree FD expansion (7) the first term must vanish for any differential operator, the kernel of A is $e = (1, \ldots, 1)^T$. In line with the approach described in [12], we solve the augmented system

$$\begin{pmatrix} A & e \\ e^T & 0 \end{pmatrix} \cdot \begin{pmatrix} p \\ \alpha \end{pmatrix} = \begin{pmatrix} r \\ 0 \end{pmatrix} , \tag{22}$$

whose unique solution is the one satisfying $A \cdot p = r - \alpha e$, where the new right hand side is the projection of r onto the range of A. Moreover, a unique solution is singled out by the condition $e^T \cdot p = 0$. This approach in particular addresses the possibility that the PPE compatibility condition (16) may be not satisfied exactly.

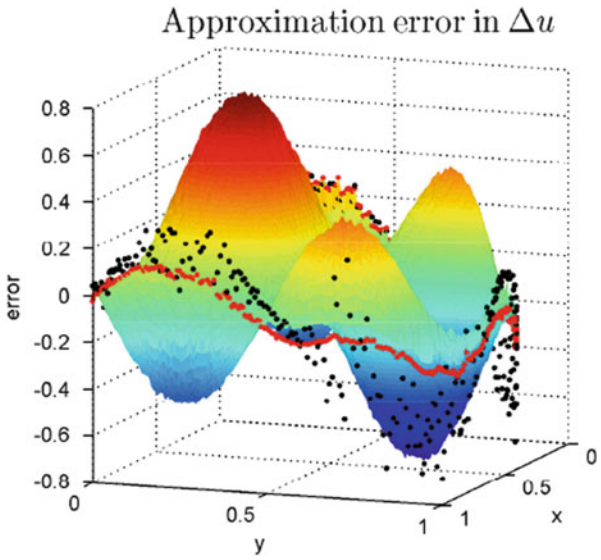


Fig. 5 Error function when approximating Δu using meshfree FD. *Black dots*: use FD directly at boundary points. *Red dots*: extrapolate function Δu from interior to the boundary

The gradient of the resulting pressure is then approximated via standard meshfree FD at all interior points.

The maximum admissible time step of the numerical scheme is determined by the viscosity term in the explicit case, i.e., $\Delta t = O(h^2)$, and by the nonlinear advection term in the semi-implicit case, i.e., $\Delta t = O(h)$. These stability time step restrictions are in line with those observed for the numerical method presented in [29]. However, they are different from the phenomenon observed and analyzed in [14, 25] for a different PPE reformulation and a different numerical discretization. In that study, the parabolic scaling $\Delta t = O(h^2)$ is observed to be required for stability, even if viscosity is treated implicitly.

3.4 Numerical Results

In order to investigate the convergence of the numerical scheme developed in Sect. 3.3, we use the same manufactured solution (20) as for the VHE, set $\nu = 1$, and calculate the pressure $p(x, y, t) = -\cos(t)\cos(\pi x)\sin(\pi y)$, the forcing $\mathbf{f} = \partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla P(\mathbf{u}) - \nu \Delta \mathbf{u}$, and the boundary velocity $\mathbf{g} = \mathbf{u}$ accordingly. We use a spatially second order meshfree FD scheme (with the special treatment of $\Delta \mathbf{u}|_{\partial\Omega}$, see Sect. 3.3), and two types of time stepping: (a) forward Euler with $\Delta t = 0.2h^2$ (i.e., viscosity is treated explicitly); and second-order ImEx with $\Delta t = 0.2h$ (i.e., viscosity is treated implicitly). In all cases the boundary relaxation

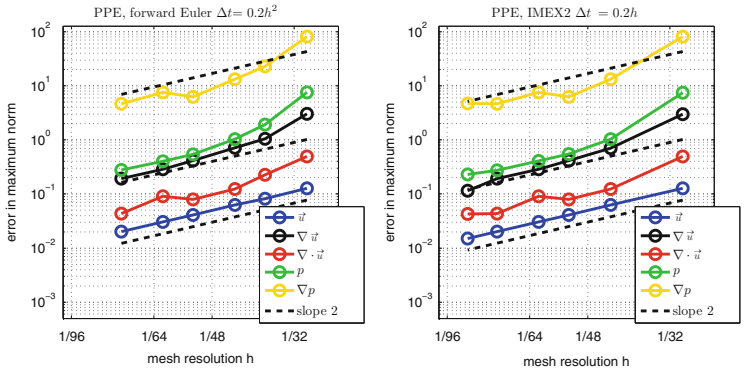


Fig. 6 Error convergence of the velocity field, the pressure, and their derivatives, for the PPE reformulation of the NSE. A spatially second order meshfree FD discretization is used. *Left panel*: forward Euler time stepping (with $\Delta t = 0.2h^2$). *Right panel*: ImEx2 time stepping (with $\Delta t = 0.2h$). In both cases, the expected $O(h^2)$ convergence is confirmed

value is chosen $\lambda = 30$. The numerical results, shown in Fig. 6, demonstrate that we obtain an overall second order convergence rate for all quantities of interest: the velocity field, its gradient, the divergence, the pressure, and the pressure gradient.

Moreover, to demonstrate the applicability of the numerical methodology, we conduct the standard benchmark lid-driven cavity test [4] for Reynolds number 100, i.e., $\nu = 0.01$. On the domain $\Omega = (0, 1)^2$, the velocity field is zero at the boundaries, except for the tangential velocity at $y = 1$, which is 1. The initial velocity field is zero everywhere except for $y = 1$, where it equals the boundary condition. The numerical approach used here is the same forward Euler-based scheme as in the manufactured solution test. The results of the steady-state profile ($t = 20$) are shown in Fig. 7. The velocity field (left panel) is depicted in the form of normalized vectors at the approximation points. The large center vortex and the two vortices in the bottom corners are captured. Moreover, a comparison (right panel) of the velocity through the centerlines of the cavity with reference data [10] shows a good agreement even on this not very highly resolved point cloud. Note that one particularity of the PPE reformulation (13) is that the flow through the boundary need not necessarily be exactly zero due to numerical approximation errors. This is why a flow through the boundaries is visible in the scaled quiver plot. However, the actual flow through the boundary, at least an h away from the top corners, is very small (less than 10^{-3} in the given example). The fact that the flow through the boundary is negligible is also visible in the right panel of Fig. 7, in which the solid curve at $y = 0$ and the dashed curve at $x \in \{0, 1\}$ are indistinguishable from 0. Moreover, computational results on two different resolutions indicate that the total flow through the boundary decreases with h.

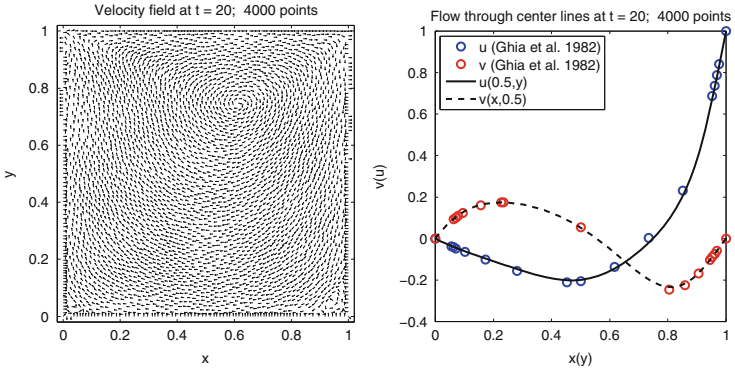


Fig. 7 Lid-driven cavity test with Re = 100 using a second-order meshfree scheme with 4,000 points. *Left*: normalized velocity field at time $T = 20$. *Right*: plot of the flow normal to the two centerlines of the cavity compared with reference data [10]

4 Conclusions and Outlook

The results in this paper show that meshfree finite differences (FD) provide a relatively straightforward methodology to approximate the solutions of vector-valued elliptic, parabolic, and fluid flow problems with electric boundary conditions (EBC), on domains without re-entrant corners. This is in contrast to finite element methods (FEM), whose simplest version, nodal-based FEM on triangular elements, fails at generating the correct solution. Instead, a Babuška paradox arises, which is shown to arise from the fact that nodal-based FEM do not capture the domain boundary's curvature.

For the vector Poisson equation, meshfree FD lead to a linear system that discretizes the Laplace operator at interior points, and the divergence operator at boundary points, in a natural and very systematic fashion. The same methodology is shown to yield first, second, and third order convergent numerical schemes. Analogous statements hold for the vector-heat equation. Implicit time stepping is a straightforward extension of the vector Poisson case; explicit time stepping is a bit more interesting (because boundary conditions remain implicit), but poses no conceptual complication.

The extension of the methods to a PPE reformulation of the Navier-Stokes equation with EBC is, again, conceptually not complicated. There is one challenge that must be overcome, namely the approximation of the Laplacian of the velocity field at the domain boundary. Once this issue has been addressed, a second-order accurate numerical scheme is obtained in which the pressure solve and the viscosity solve are decoupled. As a consequence, one can choose between an explicit and an implicit treatment of viscosity.

Being FD approaches, the numerical schemes yield the values of the velocity field $\mathbf{u}$ at the approximation points only. However, meshfree stencils can be

employed to also calculate spatial derivatives of the velocity field. For instance, the velocity gradient $\nabla \mathbf{u}$ is crucial in computing forces and stresses acting on the boundary. Our investigation of the accuracy of these derivative quantities reveals that, for all studied problems, they show no degradation in order: a kth order scheme yields kth order convergence in $\mathbf{u}$, and also in $\nabla \mathbf{u}$.

While the results demonstrate the potential of meshfree FD for these types of problems, they also give rise to further questions. One important question is whether domains with re-entrant corners can also be treated. At first glance, one would think "no", because of the lack of smoothness of the solutions on such domains. However, FD methods are known to be able to yield correct answers even for certain problems that lack smoothness (such as hyperbolic conservation laws [19]). This work also gives rise to a number of questions regarding the high-order accurate meshfree FD approximations. First, extensions of the numerical schemes for the PPE reformulation to convergence orders higher than two are of interest. Second, to avoid excessively large stencils for higher approximation orders, it is of interest whether the numerical schemes would work equally well (or better) if the meshfree FD approximations were obtained in a different fashion, such as via radial basis functions, compact FD, or deferred correction.

Acknowledgements P. Chidyagwai, R. R. Rosales, B. Seibold, and D. Zhou wish to acknowledge support by the National Science Foundation. This work was supported through grants DMS–1115269 and DMS-1115278. Furthermore, R. R. Rosales would like to acknowledge partial support by NSF through grant DMS–1318942, and B. Seibold would like to acknowledge partial support by NSF through grants DMS–1318641 and DMS–1318709. D. Shirokoff acknowledges partial funding by NSERC.

References

1. D.N. Arnold, R.S. Falk, J. Gopalakrishnan, Mixed finite element approximation of the vector Laplacian with Dirichlet boundary conditions. Math. Models Methods Appl. Sci. **22**(9), 1250024 (2012)
2. D.N. Arnold, R.S. Falk, R. Winther, Finite element exterior calculus: from Hodge theory to numerical stability. Bull. Am. Math. Soc. **47**(2), 281–354 (2010)
3. U.M. Ascher, S.J. Ruuth, R.J. Spiteri, Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. Appl. Numer. Math. **25**(2), 151–167 (1997)
4. O.R. Burggraf, Analytical and numerical studies of the structure of steady separated flows. J. Fluid Mech. **24**(1), 113–151 (1966)
5. A.J. Chorin, Numerical solution of the Navier-Stokes equations. Math. Comput. **22**, 745–762 (1968)
6. R. Courant, Variational methods for the solution of problems of equilibrium and vibrations. Bull. Am. Math. Soc. **49**(1), 1–23 (1943)
7. L. Duarte, T. Liszka, W. Tworzyako, HP-meshless cloud method. Comput. Methods Appl. Mech. Eng. **139**(1–7), 263–288 (1996)
8. F. El Dabaghi, O. Pironneau, Stream vectors in three dimensional aerodynamics. Numer. Math. **48**(5), 561–589 (1986)

9. M.S. Engelman, R.L. Sani, The implementation of normal and/or tangential boundary conditions in finite element codes for incompressible fluid flow. Int. J. Numer. Methods Fluids **2**, 225–238 (1982)
10. U. Ghia, K.N. Ghia, C.T. Shin, High-resolutions for incompressible flow using the Navier-Stokes equations and a multigrid method. J. Comput. Phys. **48**(3), 387–411 (1982)
11. J.L. Guermond, P. Minev, J. Shen, An overview of projection methods for incompressible flows. Comput. Methods Appl. Mech. Eng. **195**(44–47), 6011–6045 (2006)
12. W.D. Henshaw, A fourth-order accurate method for the incompressible Navier-Stokes equations on overlapping grids. J. Comput. Phys. **113**(6), 13–25 (1994)
13. W.D. Henshaw, H.-O. Kreiss, L.G.M. Reyna, A fourth-order accurate difference approximation for the incompressible Navier-Stokes equations. Comput. Fluids **23**(4), 575–593 (1994)
14. W.D. Henshaw, N.A. Petersson, A split-step scheme for the incompressible Navier-Stokes equations, in *Numerical Simulation of Incompressible Flows*, ed. by M.M. Hafez, vol. 2502 (World Scientific, River Edge, 2003), pp. 108–125
15. B.N. Jiang, J. Wu, L.A. Povinelli, The origin of spurious solutions in computational electromagnetics. J. Comput. Phys. **125**(1), 104–123 (1996)
16. H. Johnston, J.-G. Liu, A finite difference method for incompressible flow based on local pressure boundary conditions. J. Comput. Phys. **180**(1), 120–154 (2002)
17. U. Kangro, R. Nicholaides, Divergence boundary conditions for vector Helmholtz equations with divergence constraints. ESAIM Math. Model. Numer. Anal. **33**(3), 479–492 (1999)
18. P. Lancaster, K. Salkauskas, Surfaces generated by moving least squares methods. Math. Comput. **37**, 141–158 (1981)
19. R.J. LeVeque, *Numerical Methods for Conservation Laws*, 2nd edn. (Birkhäuser, Basel/Boston, 1992)
20. D. Levin, The approximation power of moving least–squares. Math. Comput. **67**, 1517–1531 (1998)
21. T. Liszka, J. Orkisz, The finite difference method at arbitrary irregular grids and its application in applied mechanics. Comput. Struct. **11**, 83–95 (1980)
22. I.D. Mayergoyz, A new point of view on the mathematical structure of Maxwell's equations. IEEE Trans. Magn. **29**(2), 1315–1320 (1993)
23. J.-C. Nédélec, Mixed finite elements in $\mathbb{R}^3$. Numer. Math. **35**(3), 315–341 (1980)
24. P.-O. Persson, G. Strang, A simple mesh generator in MATLAB. SIAM Rev. **46**(2), 329–345 (2004)
25. N.A. Petersson, Stability of pressure boundary conditions for Stokes and Navier-Stokes equations. J. Comput. Phys. **172**(1), 40–70 (2001)
26. P.-A. Raviart, J.-M. Thomas, A mixed finite element method for 2-nd order elliptic problems, in *Mathematical Aspects of Finite Element Methods*. Volume 606 of Lecture Notes in Mathematics (Springer, Berlin, 1977), pp. 292–315
27. B. Seibold, M-Matrices in meshless finite difference methods. Dissertation, Department of Mathematics, University of Kaiserslautern, 2006
28. B. Seibold, Minimal positive stencils in meshfree finite difference methods for the Poisson equation. Comput. Methods Appl. Mech. Eng. **198**(3–4), 592–601 (2008)
29. D. Shirokoff, R.R. Rosales, An efficient method for the incompressible Navier-Stokes equations on irregular domains with no-slip boundary conditions, high order up to the boundary. J. Comput. Phys. **230**(23), 8619–8646 (2011)
30. R. Temam, Sur l'approximation de la solution des equations de Navier-Stokes par la methode des pas fractionnaries, II. Arch. Ration. Mech. Anal. **33**(3), 377–385 (1969)
31. R. Verfürth, Mixed finite element approximation of the vector potential. Numer. Math. **50**(6), 685–695 (1986)
32. D. Zhou, High-order numerical methods for pressure Poisson equation reformulations of the incompressible Navier-Stokes equations. Dissertation, Temple University, 2014

Numerical Integration of On-the-Fly-Computed Enrichment Functions in the PUM

Marc Alexander Schweitzer and Sa Wu

Abstract The approximation power of a Partition of Unity Method stems from the use of problem-dependent enrichment functions which are, in general, non-smooth functions. However, these enrichments are not always known analytically. Even in the case of analytically known enrichments, their integration is usually challenging. The direct use of standard quadrature rules is, in general, not appropriate. But, effective numerical integration is possible by using all a priori information of the analytically known enrichment, i.e. the locations of discontinuities and singularities and the orders of the latter.

In this study, we now consider the more involved case of numerical enrichments which are computed on-the-fly via an embedded particle method. Therefore, subdivision approaches using a priori information are not (directly) applicable. This study aims to investigate, whether the particularities of the underlying particle method and the construction of the enrichments might allow for better convergence than theory would suggest.

Keywords Numerical enrichment functions • Quadrature • Partition of unity methods

1 Introduction

Generalized Finite Element Methods (GFEM), also called Partition of Unity Methods (PUM), [1, 3, 8, 13, 14, 22, 23, 25, 26, 29, 32, 33] obtain their approximation power from the use of local, problem-dependent approximation spaces $\mathcal{V}_a, a \in A$.

M.A. Schweitzer
Institut für Numerische Simulation, Rheinische Friedrich-Wilhelms-Universität Bonn, Bonn, Germany
e-mail: schweitzer@ins.uni-bonn.de

S. Wu (✉)
Sonderforschungsbereich 1060, Rheinische Friedrich-Wilhelms-Universität Bonn, Bonn, Germany
e-mail: wu@ins.uni-bonn.de

M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations VII*, Lecture Notes in Computational Science and Engineering 100,
DOI 10.1007/978-3-319-06898-5_13

These are spliced together with some Partition of Unity (PU) $\{\varphi_a, a \in A\}$, see Eq. (2), to form a global approximation space

$$\mathcal{V}_{\text{global}}^{\text{PU}} = \sum_{a \in A} \varphi_a \mathcal{V}_a \ . \tag{1}$$

The local spaces $\mathcal{V}_a$ are in general comprised of smooth, discontinuous and singular functions called enrichments. Often, these singularities or discontinuities are known a priori. Hence, when actually using these enrichments, an appropriate (numerical) integration scheme can be selected or constructed a priori [27, 28, 41]. However, if the enrichment functions are not known analytically, the situation becomes far more challenging.

Classical results on numerical quadrature deal with sufficiently smooth functions. (High) Regularity of the integrand is used to derive the respective (high) order of convergence. However, even in the case of piecewise smooth functions, a finite jump leads to a deterioration of the convergence order. For the one-dimensional setting with quadrature nodes

$$x_e \in \mathbb{R} \ , \quad e \in \{1, \ldots, M\} \ , \quad e < f \Rightarrow x_e < x_f \ , \quad h := \max_{1<e\leq M} |x_e - x_{e-1}| \ ,$$

a modified Taylor expansion around the finite jump of some otherwise arbitrarily smooth function f can be used to show an $O(h)$ error estimate [9]. In particular, this estimate is shown to be applicable to both composite quadrature rules of finite polynomial order, e.g. composite trapezoidal rule, composite Simpson rule, and spectral quadrature rules, e.g. Gauss-Legendre, Gauss-Chebyshev.

Hence, with the full tensor product construction of $d-$dimensional quadrature rules, this theory would suggest a convergence order of only $O(M^{\frac{-1}{d}})$ in the number of function evaluations M needed for application of some quadrature rule. Similarly, for smooth functions f with jumps in their n-th derivatives $f^{(n)}$, in general, only a $O(M^{\frac{-n}{d}})$ error estimate holds.

However, in the one-dimensional setting, the situation improves, if the location of the jump is known. Using a domain decomposition at this location and the sum of the quadrature results on each of the computed subdomains, we can retrieve the original convergence order of whatever quadrature rule is used. The same holds in higher dimensional settings, if the domain can be split at the known "lines" of discontinuity. Moreover, even in the case of known boundary singularities with unknown exponent, spectral quadrature rules can be constructed [12]. But, this construction again depends on the explicit knowledge of the location of those singularities.

In the finite element (FE) setting, the common assumption is that the underlying mesh resolves all the features of the solution, coefficients and the domain. Hence, using sufficiently high order quadrature rules on each cell of the underlying mesh yields a stable and accurate assembly of the mass and stiffness matrices. Further studies on the relationship between quadrature error and the error of the Galerkin solution can be found in [40].

In the typical PUM setting, see (1) and Sect. 2, a priori knowledge of the explicit location of discontinuities and singularities, e.g. interfaces, jump points of coefficients, reentrant corners, is used for construction of quadrature rules for analytically known enrichments [4,6,27,28,31,32,39]. With respect to the employed PU, a domain decomposition can resolve the piecewise (rational) nature of the PU [8, 32].

In this paper, we focus on the influence of the enrichments only on numerical integration in system assembly. Thus, we employ a FE-based PU. Moreover, we study a model problem of crack nucleation and growth in a brittle material. We obtain the numerical enrichments for this problem from an embedded particle method based on Peridynamics (PD), see Sect. 3. To recover the developing cracks, the enrichments contain finite jump discontinuities. A priori knowledge on the exact location of these is not available. In principle, the cracks may be reconstructed a posteriori. However, this is rather expensive and too involved for on-the-fly enrichment.

Hence, this paper presents a numerical study on the use of standard, (piecewise-) polynomial-interpolation-based quadrature rules for a PUM discretization with embedded PD enrichment. The reasoning behind this is that the $O(h)$ estimate from theory might be too conservative and, for particular problems, convergence may be better. In particular, convergence might be good enough to balance errors from the remainder of the coupled global discretization.

The remainder of this paper is organized as follows. First, in Sect. 2, we briefly review the PUM. In Sect. 3, we given an overview of the considered model problem and its discretization. In particular, we give a short overview of PD and the construction of the enrichments from the PD solution. Then, we present the results of our numerical experiments in Sect. 4. The results show that, in a typical localized fracture example, the quadrature error is less pronounced than the conservative $O(M^{-\frac{1}{d}})$ estimate suggests. Finally, we conclude with some remarks in Sect. 5.

2 The Partition of Unity Method

In this section we shortly summarize the basic construction of a PUM, see [13, 25] for details. To this end let us consider a PU $\varphi_a, a \in A \subset \mathbb{N}$ with $D := |A|$, i.e. a set of D sufficiently smooth functions $0 \leq \varphi_a$ with

$$\sum_{a\in A}\varphi_a \equiv 1 \text{ on } \bar{\Omega} \ , \qquad \forall a \in A : \omega_a := \text{supp}\,(\varphi_a) \ , \qquad \bigcup_{a\in A}\omega_a \supseteq \bar{\Omega} \ . \tag{2}$$

With this PU, so called enrichments $\boldsymbol{\eta}_{a,b}, a \in A, b \in B_a \subset \mathbb{N}$, which span local approximation spaces

$$\forall a \in A \ : \qquad \mathcal{V}_a := \text{span}\left\{\boldsymbol{\eta}_{a,b} | a \in B_a\right\} \ni 1 \tag{3}$$

on ω_a, are spliced together to form a global approximation space

$$\mathcal{V}^{\mathrm{PU}}_{\mathrm{global}} = \sum_a \varphi_a \mathcal{V}_a = \{\sum_{a\in A} \varphi_a \sum_{b\in B_a} u_{a,b}\boldsymbol{\eta}_{a,b} | u_{a,b} \in \mathbb{R}\} \tag{4}$$

on $\bar{\Omega}$. This approach is an efficient framework for the solution of problems involving localized special behaviour, e.g. singularities, discontinuities, multiscale behaviour [1,3,11,33].

Usually, the local spaces $\mathcal{V}_a$ are composed of a smooth, typically polynomial, part $\mathcal{P}$ and a problem-dependent part $\mathcal{E}_a$

$$\mathcal{V}_a = \mathcal{P} + \mathcal{E}_a \ . \tag{5}$$

For a field problem $\boldsymbol{u} : \Omega \to \mathbb{R}^d$, (3) and (5) become

$$\boldsymbol{\eta}_{a,b} : \operatorname{supp}\varphi_a \to \mathbb{R}^d \ , \quad \mathcal{V}_a \supseteq \mathcal{P} \supseteq \operatorname{span}\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d\} \ , \quad \text{with } (\boldsymbol{e}_c)_d = \delta_{c,d} \ .$$

The local spaces $\mathcal{V}_a$ give the method its approximation power whereas the PU φ_a provides the required global continuity. Thus, whenever microscale effects, discontinuities or singularities can be found in the solutions, the enrichments $\boldsymbol{\eta}_{a,b} \in \mathcal{E}_a \subset \mathcal{V}_a$ have to be able to capture these.

However, such features make numerical integration very challenging. Typically, the PU is made up of piecewise smooth functions, e.g. they are often piecewise polynomial FE shape functions [6,26] or piecewise rational via a Shepard or MLS construction [8,13]. Hence, special attention is paid to the construction of quadrature cells ω, such that $\varphi_a|_\omega, \omega_a \cap \omega \neq \emptyset$ are smooth rational functions. Then, assuming the enrichments $\boldsymbol{\eta}_{a,b}|_\omega$ to be smooth enough on those cells, endowing the quadrature cells ω with Gauss-type quadrature of sufficiently high order enables one to achieve the necessary accuracy.

If the PU stems from the nodal shape functions of a FE method (FEM) [26], the PU does not require special attention with respect to numerical integration, if the underlying elements are used as quadrature cells. Moreover, explicit knowledge of the location of discontinuities and singularities of the enrichments $\boldsymbol{\eta}_{a,b}$ can be used to create additional (sub-)triangulations of the underlying mesh just for the purposes of integrating the weak forms. Yet, the challenge arising from the use of numerical enrichments computed on-the-fly (by a particle method) still remains.

In this study, we use a PUM based on such nodal shape functions, i.e. trilinear hexahedral finite elements in $\Omega \subset \mathbb{R}^3$. But, we do not have explicit knowledge of the location of discontinuities of the enrichment which is computed on-the-fly.

3 Model Problem and Discretization

The model problem considered throughout this paper is the (linearized) elastic wave equation $\boldsymbol{u} : \Omega \subset \mathbb{R}^3 \times [0, T] \to \mathbb{R}^3$

$$\rho(\boldsymbol{x})\ddot{\boldsymbol{u}}(\boldsymbol{x}, t) = \mu(\boldsymbol{x})\Delta\boldsymbol{u}(\boldsymbol{x}, t) + (\lambda(\boldsymbol{x}) + \mu(\boldsymbol{x}))\nabla \operatorname{div} \boldsymbol{u}(\boldsymbol{x}, t) + \boldsymbol{b}(\boldsymbol{x}, t) \tag{6}$$

with $\boldsymbol{x} = (x, y, z)$, displacement $\boldsymbol{u}(\boldsymbol{x}, t) = \boldsymbol{X}(\boldsymbol{x}, t) - \boldsymbol{x}$ and Lamé-parameters λ and μ, which can be derived from the Poisson-ratio ν and the bulk-modulus K. This model is applicable in fracture mechanics to study, i.a., the growth of already existing cracks using additional growth criteria. However, (more involved) additional modelling is needed to allow for crack initiation.

The local particle method employed throughout this paper is based on Peridynamics (PD) [2, 30, 36–38], a nonlocal formulation of the mechanics of solids which directly allows for the nucleation and growth of multiple interacting cracks. In the simplest, history-dependent, bond-based case with (pairwise) force density $\boldsymbol{f} : \left(\mathbb{R}^3\right)^{(-\infty,0]} \times \mathbb{R}^3 \to \mathbb{R}^3$ and a difference operator

$$\left(\tau_{\boldsymbol{x}',\boldsymbol{x},t}\boldsymbol{u}\right)(t') = \begin{cases} \boldsymbol{u}(\boldsymbol{x}', t' + t) - \boldsymbol{u}(\boldsymbol{x}, t' + t) & t' \in [-t, 0] \\ \boldsymbol{u}(\boldsymbol{x}', 0) - \boldsymbol{u}(\boldsymbol{x}, 0) & t' \in (-\infty, -t) \end{cases}$$

with a local area of influence $H = H(\boldsymbol{x}) \subseteq \Omega$ of some $\boldsymbol{x} \in \Omega$, the PD equation of motion is

$$\rho(\boldsymbol{x})\ddot{\boldsymbol{u}}(\boldsymbol{x}, t) = \int_{H(\boldsymbol{x})} \boldsymbol{f}\left(\tau_{\boldsymbol{x}',\boldsymbol{x},t}\boldsymbol{u}, \boldsymbol{x}' - \boldsymbol{x}\right) \mathrm{d}\boldsymbol{x}' + b(\boldsymbol{x}, t) \,. \tag{7}$$

Note that, in this particular (common) notation, possible interactions are only based on the reference configuration Ω (in contrast to the current configuration $\boldsymbol{X}(\Omega, t) = \{\boldsymbol{x} + u(\boldsymbol{x}, t) | \boldsymbol{x} \in \Omega\}$ at time t). In particular, only forces resulting from the initial configuration are considered, i.e. for $\boldsymbol{y} \notin H(\boldsymbol{x})$, $\boldsymbol{y}$ never influences $\boldsymbol{x}$, $H(\boldsymbol{x}) \neq H(\boldsymbol{X}(\boldsymbol{x}, t))$. Hence, contact forces arising, i.a., from large deformation are not considered. We also remark that (7) admits discontinuous solutions and is valid even on developing cracks. Thus, this generalized continuum model directly allows for crack initiation. Therefore, it is a suitable sub-model for our model problem.

3.1 Discretization

We employ a standard explicit temporal discretization using timesteps $t_n = (\Delta t)n$ and central differences (Verlet-Störmer)

$$\boldsymbol{u}(\boldsymbol{x}, t_{n+1}) \approx 2\boldsymbol{u}(\boldsymbol{x}, t_n) - \boldsymbol{u}(\boldsymbol{x}, t_{n-1}) + (\Delta t)^2 \ddot{\boldsymbol{u}}(\boldsymbol{x}, t_n) \tag{8}$$

for the treatment of time in (6) and (7).

The spatial Galerkin discretization of (6) with

$$\boldsymbol{u}_{\text{global}}(\cdot, t_n) \in \mathcal{V}_{\text{global}}^{\text{PU},n} = \sum_{a\in A} \varphi_a \mathcal{V}_a^n = \operatorname{span}\left\{\varphi_a \boldsymbol{\eta}_{a,b}^n | a \in A, b \in B_a^n\right\} ,$$

a space with basis $\mathcal{B}^n = \{\varphi_a \boldsymbol{\eta}_{a,b}^n\} = \{\boldsymbol{\phi}_d^n | 1 \le d \le D^n := \dim \mathcal{V}_{\text{global}}^{\text{PU},n}\}$,

$$\boldsymbol{u}_{\text{global}}(\boldsymbol{x}, t_n) = \sum_{a\in A} \varphi_a(\boldsymbol{x}) \sum_{b\in B_a^n} u_{a,b}^n \boldsymbol{\eta}_{a,b}^n(\boldsymbol{x}) \text{ with } \mathbf{u}^n = (u_{a,b}^n)_{a\in A, b\in B_a^n} \in \mathbb{R}^{D^n} ,$$

leads to the discrete linear system of equations

$$\mathbf{M}^{n+1,n+1}\mathbf{u}^{n+1} = 2\mathbf{M}^{n+1,n}\mathbf{u}^n - \mathbf{M}^{n+1,n-1}\mathbf{u}^{n-1} + (\Delta t)^2 \left(-\mathbf{K}^{n+1,n}\mathbf{u}^n + \mathbf{b}^{n+1}\right) \tag{9}$$

where, without boundary conditions,

$$\begin{aligned}
\mathbf{M}^{k,l} &:= \left(\int_\Omega \rho \boldsymbol{\phi}_1 \cdot \boldsymbol{\phi}_2\right)_{\boldsymbol{\phi}_1\in\mathcal{B}^k, \boldsymbol{\phi}_2\in\mathcal{B}^l} = \left(\mathbf{M}_{c,d}^{k,l}\right)_{1\le c\le D^k, 1\le d\le D^l} \in \mathbb{R}^{D^k\times D^l} \\
\mathbf{K}^{k,l} &:= \left(\int_\Omega \mu \nabla\boldsymbol{\phi}_1 \cdot \nabla\boldsymbol{\phi}_2 + (\lambda+\mu) \operatorname{div}\boldsymbol{\phi}_1 \operatorname{div}\boldsymbol{\phi}_2\right)_{\boldsymbol{\phi}_1\in\mathcal{B}^k, \boldsymbol{\phi}_2\in\mathcal{B}^l} \\
&= \left(\mathbf{K}_{c,d}^{k,l}\right)_{1\le k\le D^k, 1\le d\le D^l} \in \mathbb{R}^{D^k\times D^l} , \\
\boldsymbol{b}^k &:= \left(\int_\Omega \boldsymbol{\phi}(x)\boldsymbol{b}(\boldsymbol{x}, t_n)\,\mathrm{d}\boldsymbol{x}\right)_{\boldsymbol{\phi}\in\mathcal{B}^k} = \left(\mathbf{b}_d^k\right)_{1\le d\le D^k} \in \mathbb{R}^{D^k} .
\end{aligned} \tag{10}$$

For the PD (7), a choice of collocation points $\boldsymbol{x}_i$ together with (8) yields

$$\boldsymbol{u}_i^{n+1} = 2\boldsymbol{u}_i^n - \boldsymbol{u}_i^{n-1} + (\Delta t)_{\text{local}}^2 \left(\sum_{\boldsymbol{x}_j\in H(\boldsymbol{x}_i)} \boldsymbol{f}\left(\tau_{\boldsymbol{x}_j,\boldsymbol{x}_i,t_n}\boldsymbol{u}, \boldsymbol{x}_j - \boldsymbol{x}_i\right)\mathcal{V}_{i,j} + \boldsymbol{b}(\boldsymbol{x}_i, t_n)\right) . \tag{11}$$

with volume fragments $\mathcal{V}_{i,j}$, discretized force density $\boldsymbol{f}$ and discretized "history"-restriction $\tau_{\boldsymbol{x}_j,\boldsymbol{x}_i,t_n}$ to accommodate for $\boldsymbol{x} \in \{\boldsymbol{x}_i\}$, $t \in (\Delta t)_{\text{local}}\mathbb{Z}$. Moreover, from the force density $\boldsymbol{f}$ we obtain an incidence matrix

$$\mathbf{A}_{i,j}^n = \begin{cases} 1 & \boldsymbol{f}\left(\tau_{\boldsymbol{x}_j,\boldsymbol{x}_i,t_n}\boldsymbol{u}, \boldsymbol{x}_j - \boldsymbol{x}_i\right) \ne 0 \\ 0 & \boldsymbol{f}\left(\tau_{\boldsymbol{x}_j,\boldsymbol{x}_i,t_m}\boldsymbol{u}, \boldsymbol{x}_j - \boldsymbol{x}_i\right) = 0 \quad \forall m \ge n \end{cases} \tag{12}$$

as a natural damage measure to identify cracks.

The goal now is to incorporate the cracks developing in a local PD (11) into the global elasticity problem (9). To this end, we follow the so-called Global-Local approach.

3.2 Global-Local Enrichments

Often, special features of the solution are known a priori because they stem from particular features of the underlying domain, i.a. internal interfaces, reentrant corners. In these cases, the available a priori knowledge is used to obtain the enrichments, e.g. via an asymptotic expansion of the solution. However, in many applications, the special features of the solution are not known analytically. Yet, on a finer scale, these features might become apparent as solutions to some local problems. One approach at the bidirectional coupling of such two scales is the Global-Local approach [7, 16–18, 43, 44].

In the PUM setting, we first compute some coarse global solution $\boldsymbol{u}_{\text{global}}$ on Ω. If effects from a finer scale dominate the features of the solution on some subdomain $\Omega_{\text{local}} \subset \Omega$, $\boldsymbol{u}_{\text{global}}$ is used to obtain boundary conditions on $\partial\Omega_{\text{local}}$. With these, a local, fine-scale solution $\boldsymbol{u}_{\text{local}}$ on Ω_{local} can be computed. Then, $\boldsymbol{u}_{\text{local}}$ is used as an enrichment for the improvement of $\boldsymbol{u}_{\text{global}}$. In spirit, this is similar to a residual correction of the global solution [43] or, in other words, the application of some Newton-like iteration for a linear problem.

In the context of (9) and (11), this yields the following Algorithm.

Algorithm 1 Global-Local iteration of PUM with embedded PD

1. Discretize PDE model (6) using a particular $\mathcal{V}_{\text{global}}^{\text{PU},n}$ at time t_n and obtain (9) with an approximation $\boldsymbol{u}_{\text{global}}(\cdot, t_n) = \mathbf{u}^n \cdot (\boldsymbol{\phi})_{\boldsymbol{\phi} \in \mathcal{B}^n}$.
2. Extract subdomain $\Omega_{\text{local}}^n \subset \Omega$ to run PD on.
3. Run PD (11) with $\boldsymbol{x}_i \in \Omega_{\text{local}}^n, \boldsymbol{u}_i^n = \boldsymbol{u}_{\text{global}}(\boldsymbol{x}_i, t_n)$ (and (initially coarse) boundary conditions from $\boldsymbol{u}_{\text{global}}(\cdot, t_n)$) to obtain $\boldsymbol{u}_i^{n+1}$ and $\mathbf{A}_{i,j}^{n+1}$.
4. Use $\boldsymbol{x}_i, \boldsymbol{u}_i^{n+1}$ and $\mathbf{A}_{i,j}^{n+1}$ to construct vector field

$$\boldsymbol{u}_{\text{local}}^{n+1} : \Omega_{\text{local}}^n \to \mathbb{R}^3 \tag{13}$$

5. Obtain $\boldsymbol{u}_{\text{global}}(\cdot, t_{n+1}) = \mathbf{u}^{n+1} \cdot (\boldsymbol{\phi})_{\boldsymbol{\phi} \in \mathcal{B}^{n+1}}$ from (9) with updated local spaces (5)

$$\mathcal{E}_a^{n+1} = \begin{cases} \text{span}\{(\boldsymbol{u}_{\text{local}}^{n+1} \cdot e_1)e_1, \ldots, (\boldsymbol{u}_{\text{local}}^{n+1} \cdot e_3)e_3, \} & \omega_a \cap \Omega_{\text{local}}^n \neq \emptyset \\ \{0\} & \text{otherwise} \end{cases}.$$

If required for stability, e.g. because $(\boldsymbol{u}_{\text{local}}^{n+1} \cdot e_d) \in \text{span}\{1\}$, thin out $\mathcal{E}_a^{n+1}$, e.g. (19).

6. If necessary, e.g. $\boldsymbol{u}_{\text{global}}(\cdot, t_{n+1})$ does not align well with boundary conditions obtained from $\boldsymbol{u}_{\text{global}}(\cdot, t_n)$, improve boundary conditions and return to step 3.

We point out, that, depending on the method chosen for the solution of the local problem, a sufficiently accurate quadrature scheme might be at hand [17, 18]. For example, when using a FEM on a locally refined mesh for the computation $\boldsymbol{u}_{\text{local}}^{n+1}$, the elements of the refined mesh readily provide viable quadrature cells for the use of $\boldsymbol{u}_{\text{local}}^{n+1}$ as enrichment. Yet, this is a very expensive quadrature rule. Here, $\boldsymbol{u}_{\text{local}}^{n+1}$ is obtained through a particle method. Hence, beyond issues of interpolation and obtaining gradients, only nodal integration using the particles as nodes and their associated volumes as weights can be applied directly.

3.3 Construction of Enrichment Functions

In this work, $\boldsymbol{u}_{\text{local}}^{n+1}$ in (13) is constructed using a modification of the Moving Least Squares (MLS) Approximation [10, 20, 21, 24, 35, 42]. The details of our construction beyond the following brief overview can be found in [34].

Given some space $\mathcal{S} \subset (\mathbb{R}^3)^{\Omega}$ and weights $W_i^{n+1} : \Omega_{\text{local}}^n \to [0, \infty)$, each representing the importance of data at $\boldsymbol{x}_i$ for an evaluation of the approximand $\boldsymbol{u}_{\text{local}}^{n+1}$ at some $\boldsymbol{x}$, we introduce a family of error functionals

$$J_{\boldsymbol{x}}^{n+1} : \quad \mathcal{S} \to [0, \infty) , \quad J_{\boldsymbol{x}}^{n+1}(\boldsymbol{s}) = \sum_i W_i^{n+1}(\boldsymbol{x}) \left(\boldsymbol{u}_i^{n+1} - \boldsymbol{s}(\boldsymbol{x}_i - \boldsymbol{x})\right)^2 . \tag{14}$$

With this family of error functionals, we obtain the overall MLS approximation at arbitrary points $\boldsymbol{x}$ as

$$\boldsymbol{u}_{\text{local}}^{n+1}(\boldsymbol{x}) = \boldsymbol{s}_{\boldsymbol{x}}(0) \qquad \text{where} \qquad \boldsymbol{s}_{\boldsymbol{x}} = \operatorname*{argmin}_{s \in \mathcal{S}} J_{\boldsymbol{x}}^{n+1}(\boldsymbol{s}) . \tag{15}$$

The weights W_i^{n+1} are typically given as shifted copies of some radial, compactly supported, non-negative, smooth function. In [34] we introduced a modification of these weights W_i^{n+1} using $\mathbf{A}_{i,j}^{n+1}$. This modification allows us to nullify the influence of some data pairs $(\boldsymbol{x}_i, \boldsymbol{u}_i^{n+1})$ on the approximands value at $\boldsymbol{x}$ across a crack from $\boldsymbol{x}_i$. In spirit, this is similar to a visibility criterion [5] and allows us to obtain an enrichment $\boldsymbol{u}_{\text{local}}^{n+1}$ with finite jump discontinuities.

Throughout this paper we use the simplest choice of $\mathcal{S} = \text{span}\{(1, 1, 1)\}$ to obtain a Shepard-like [35] approximation of the PD displacement field as enrichment.

4 Numerical Examples

In the following, we present some results of our numerical experiments. Here, we first focus on the numerical integration of analytically known discontinuous functions to obtain a reference measure on the convergence behaviour and overall quality of the used numerical integration schemes. These functions cover some selected cases of finite jump discontinuities and consider the reduced problem of numerically computing the integral $\int_\Omega f_i$.

Then, we turn to an actual example of a PD simulation of a brittle thin plate $\Omega \subset \mathbb{R}^3$. With respect to the Algorithm 1 and (5), we use

$$n = 0\,, \quad \mathcal{P} = \text{span}\{e_1, e_2, e_3\}\,, \quad \forall a \in A: \quad \mathcal{E}_a^{-1} = \mathcal{E}_a^1 = \{0\}\,, \quad \Omega^0_{\text{local}} = \Omega$$

in this example. Thus, we need to compute the matrices $\mathbf{M}^{1,1}$, $\mathbf{M}^{1,0}$, $\mathbf{M}^{1,-1}$, $\mathbf{K}^{1,0}$ and load vector $\mathbf{b}^1$, see (9) and (10), for step 5 of the algorithm in the initial timestep $n = 0$. To avoid errors due to an inappropriate choice of Ω^0_{local}, we use $\Omega^0_{\text{local}} = \Omega$ throughout the remainder of this paper. Moreover, we assume an ordering of the shape functions $\{\boldsymbol{\phi}^1_d\}_{1 \le d \le D^1} = \mathcal{B}^1$ such that

$$\begin{aligned} \forall 1 \le d \le D^0 : \exists a \in A : \exists c \in \{1,2,3\} : \quad & \boldsymbol{\phi}^1_d = \varphi_a \boldsymbol{e}_c \\ \forall D^0 < d \le D^1 : \exists a \in A : \exists c \in \{1,2,3\} : \quad & \boldsymbol{\phi}^1_d = \varphi_a (\boldsymbol{u}^1_{\text{local}} \cdot \boldsymbol{e}_c) \boldsymbol{e}_c\,. \end{aligned}$$

Then, $\mathbf{M}^{1,0}$ and $\mathbf{M}^{1,-1}$ are sub-matrices of $\mathbf{M}^{1,1}$ and, $\mathbf{K}^{1,0}$ is a sub-matrix of $\mathbf{K}^{1,1}$ (10). Hence, we choose to only investigate $\mathbf{M}^{1,1}$ and $\mathbf{K}^{1,1}$. Also, this yields additional insights into the numerical integration of the derivatives of the computed discontinuous enrichments via $\mathbf{K}^{1,1}_{c,d}$ with $D^0 < c, d \le D^1$. The configuration and loads are chosen to produce, after around 22,000 timesteps, a nucleating crack, see Fig. 1, which branches after around 24,000 timesteps, see Fig. 2.

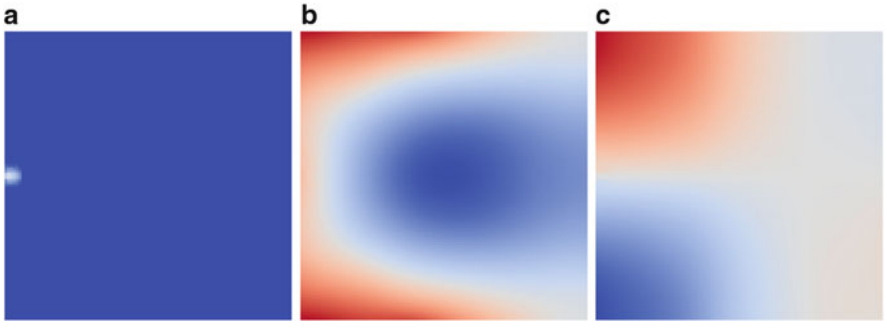


Fig. 1 Plots of the global PD solution after 22,500 timesteps. (**a**) Crack initiation. (**b**) x component of PD solution. (**c**) y component of PD solution

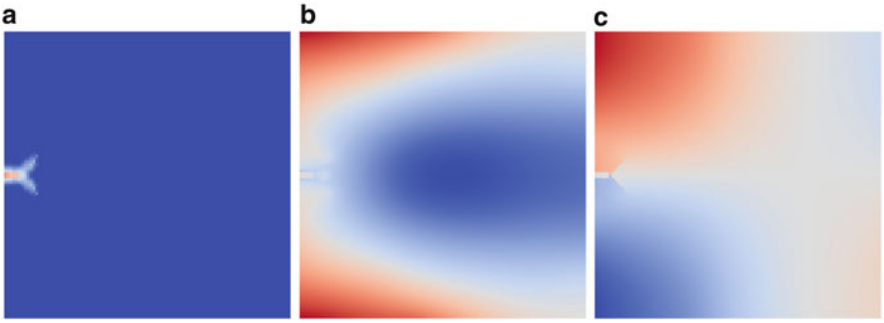


Fig. 2 Plots of the global PD solution after 25,000 timesteps. (**a**) Crack branching. (**b**) x component of PD solution. (**c**) y component of PD solution

After having run the global PD, a FE mesh is used to discretize the domain Ω in its initial configuration. Using the resulting FE nodal shape functions as PU, and $\boldsymbol{u}^1_{\text{local}}$ obtained from the final configuration of the PD simulation, we then assemble $\mathbf{M} := \mathbf{M}^{1,1}$, $\mathbf{K} := \mathbf{K}^{1,1}$, $\mathbf{b} := \mathbf{b}^1$ using various quadrature rules and compute $\mathbf{x} := \mathbf{M}^{-1}\mathbf{b}$, see (9). The boundary conditions for both the global PD and global PUM are vanishing Neumann conditions and, the same load $\boldsymbol{b}$ is applied in both models. However, we incorporate the global enrichment functions only locally on some patches, see (19), to mimic the effect of having chosen Ω^0_{local}, i.e. using useful problem-dependent enrichments. The results of this choice are highlighted in Figs. 6, 7, 8 and 9.

The chosen standard quadrature rules are

- Composite rectangle rule,
- Composite trapezoidal rule,
- Composite Simpson rule,
- Gauss-Legendre,

on a regular, full grid with M quadrature nodes, i.e. the full tensor product of the 1-dimensional quadrature rules.

Recall that by a "correct" choice of a domain decomposition, a higher order convergence can be observed. To avoid such outliers cluttering the plots, we remove results where the numerical integration yields the almost exact solution by "chance", i.e. we remove the configurations which gave relative errors $\varepsilon_{\text{rel}} < 10^{-10}$ in (16) and (17).

4.1 Reference Case

As mentioned above, we first focus on approximating the integral of a discontinuous function $f : [0, 1]^3 \to \mathbb{R}$

$$I(f) = \int_\Omega f(\boldsymbol{x})\,\mathrm{d}x \approx \sum_{i=0}^{M} f(\boldsymbol{x}_i) w_i = Q(f) = (Q((\boldsymbol{x}_i)_i, (\omega_i)_i))\,(f)$$

by a quadrature scheme Q using M point evaluations at $\boldsymbol{x}_i$ with weights w_i. We consider the error measure

$$\varepsilon^I_{\mathrm{rel}} = \varepsilon^I_{\mathrm{rel}}(f) = \frac{|Q(f) - I(f)|}{|I(f)|}\,. \tag{16}$$

As f, we use the characteristic function of the open ball of given radius r around y

$$\chi_{r,\|\cdot\|;y}(\boldsymbol{x}) = \begin{cases} 1 & \|\boldsymbol{y} - \boldsymbol{x}\| < r \\ 0 & \text{otherwise} \end{cases} \quad \text{with} \qquad \|\cdot\| \in \{\|\cdot\|_1, \|\cdot\|_2, \|\cdot\|_\infty\}$$

as functions on $\Omega = [0,1]^3$ with $r \in \{\frac{1}{2}, \frac{1}{\sqrt{2}}\}$. In the case of $r = \frac{1}{\sqrt{2}}$ and $\|\cdot\|_1, \|\cdot\|_\infty$ no quadrature node of the composite rules should coincide with the discontinuities. For $r = \frac{1}{2}$ and odd M, some quadrature nodes of the composite rules will coincide with the location of the discontinuities. This may yield additional insight on the influence of quadrature points coinciding with the location of the discontinuities. When $\boldsymbol{y} = 0$, we omit $\boldsymbol{y}$ and simplify notation to $\chi_{r,\|\cdot\|}$. These reference cases should model enrichments from simple crack patterns or material inclusions.

We also consider the function

$$\begin{aligned} f_3 = {} & 1\chi_{\frac{1}{2},1;(0,\frac{1}{2},\frac{1}{2})} + 2\chi_{\frac{1}{2},1;(1,\frac{1}{2},\frac{1}{2})} + 3\chi_{\frac{1}{2},1;(\frac{1}{2},0,\frac{1}{2})} + \\ & 4\chi_{\frac{1}{2},1;(\frac{1}{2},1,\frac{1}{2})} + 5\chi_{\frac{1}{2},1;(\frac{1}{2},\frac{1}{2},0)} \end{aligned}$$

modeling a 6-way branching point of some cracks.

The results of these numerical experiments are given in Fig. 3. We observe that, in general, we obtain at most the theoretical $O(M^{-\frac{1}{d}})$ convergence. In particular, this estimate may not only appear sharp in the case of the complex branched crack but even something simple as $\chi_{\infty,\cdot}$. For multiples of characteristic functions, the significant factor for the quadrature error appears to be the change of the volume where the quadrature error actually occurs with M. Moreover, independent of the exact choice of the discontinuous function, Gaussian quadrature rules do not seem to benefit at all from their usual spectral convergence.

4.2 PUM with Global-Local PD

Now, we present some results obtained from the system assembly of an elasticity problem with enrichments obtained through PD, see Sect. 3. We measure the errors

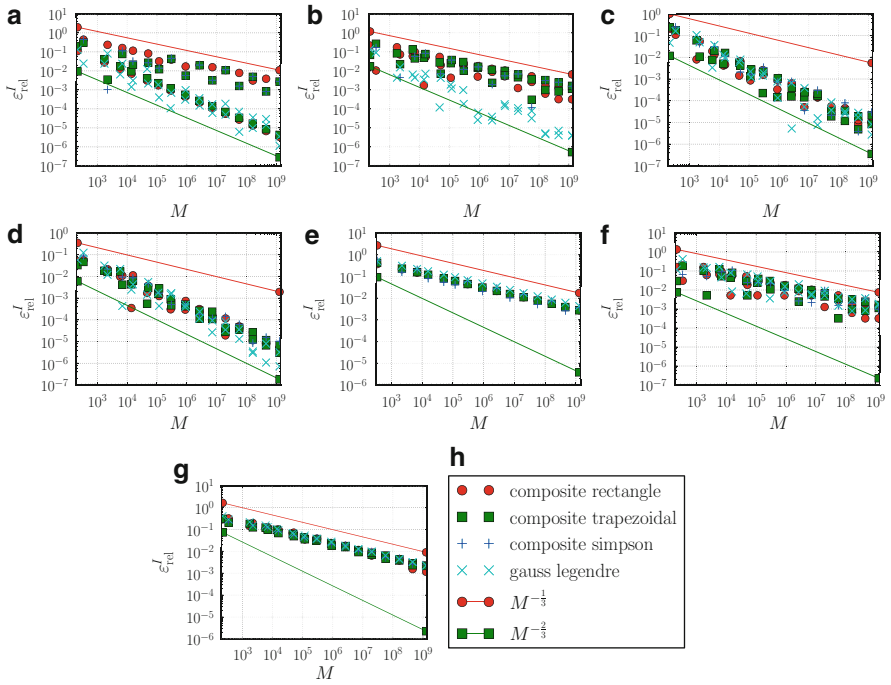


Fig. 3 Convergence plots for the approximation of the integral of the functions given in Sect. 4.1. **(a)** $\chi_{1,\frac{1}{2}}$. **(b)** $\chi_{1,\frac{1}{\sqrt{2}}}$. **(c)** $\chi_{2,\frac{1}{2}}$. **(d)** $\chi_{2,\frac{1}{\sqrt{2}}}$. **(e)** $\chi_{\infty,\frac{1}{2}}$. **(f)** $\chi_{\infty,\frac{1}{\sqrt{2}}}$. **(g)** f_3. **(h)** Legend

in the numerical assembly of $\mathbf{M}$, $\mathbf{K}$, $\mathbf{b}$, see (10), and the resulting $\mathbf{x}$. Since no reference solution is available, the reference solutions $\tilde{\mathbf{M}}$, $\tilde{\mathbf{K}}$, $\tilde{\mathbf{b}}$ and $\tilde{\mathbf{x}} = \tilde{\mathbf{M}}^{-1}\tilde{\mathbf{b}}$ are obtained through overkill with a composite rectangle rule using $2{,}049^2$ quadrature nodes per element in contrast to the maximum 513^2 used for the measurements. We employ the following error measures

$$\begin{aligned} \varepsilon^{\mathbf{M}}_{\text{rel}} &= \max_{1\le c,d\le D} \epsilon(\mathbf{M}_{c,d}, \tilde{\mathbf{M}}_{c,d}) \,, & \varepsilon^{\mathbf{K}}_{\text{rel}} &= \max_{1\le c,d\le D} \epsilon(\mathbf{K}_{c,d}, \tilde{\mathbf{K}}_{c,d}) \,, \\ \varepsilon^{\mathbf{b}}_{\text{rel}} &= \max_{1\le d\le D} \epsilon(\mathbf{b}_{d}, \tilde{\mathbf{b}}_{d}) \,, & \varepsilon^{\mathbf{x}}_{\text{rel}} &= \max_{1\le d\le D} \epsilon(\mathbf{x}_{d}, \tilde{\mathbf{x}}_{d}) \end{aligned} \tag{17}$$

with

$$\epsilon(x,y) := \begin{cases} \frac{|x-y|}{|y|} & |y| \gg 0 \\ |x| & |y| \approx 0 \end{cases}$$

in the respective plots.

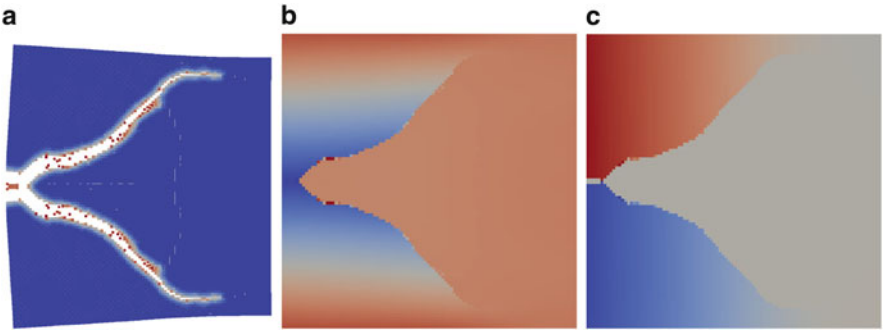


Fig. 4 Plots of the global PD solution $\{\boldsymbol{u}_i^1\}$. (**a**) Material failure in deformed configuration $\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}, t_n)$. (**b**) x component of PD solution $\{\boldsymbol{u}_i^1\}$. (**c**) y component of PD solution $\{\boldsymbol{u}_i^1\}$

The studied physical setting is that of a loaded brittle plate of dimensions

$$\Omega = [-0.1, 0.1] \times [-0.1, 0.1] \times [-0.001, 0.001]$$

with material parameters

$$\nu = 0.25\,, \qquad \rho = 500\,, \qquad K = 10^9\,.$$

Symmetric loads are applied on the top and bottom left corners, as a body force in lieu with PD, such that a crack develops in the center of the left edge of the plate. With time, this crack grows and branches into two separate directions, see Fig. 4.

The discretization parameters for PD are chosen as

$$h = 0.002 \qquad \delta = 3h\,,$$

a choice leading to a uniform $100 \times 100 \times 1$-particle discretization of Ω. The PD simulation uses a force density $\boldsymbol{f}$ from the so called PMB material model [36] with parameters c, obtained from δ and K, and s_{00}, chosen as 0.02. The first is based on theory [36], the latter is just chosen to produce a branching crack that yields a challenge in the numerical integration. The PD simulation was run for 100,000 timesteps with $\Delta t = 8 \cdot 10^{-9}$. Nonetheless, we shall refer to the final PD solution as $\boldsymbol{u}_i^1$. To avoid any additional error sources related to the global-local iteration, we limit ourselves to just a one-way coupling in this study. Thus, we run only one global PD simulation and use the PD approximation obtained at the final timestep as enrichment.

With this setup, the z-component of the global PD has no dynamics and, hence, no nonzero entries in the solution $\{\boldsymbol{u}_i^1\}$. Therefore, the numerical quadrature used in the assembly of $\mathbf{M}$, $\mathbf{K}$ and $\mathbf{b}$ is limited to the (x, y)-plane, i.e. using only 1 node in z direction in the centre of the quadrature cells. Moreover, the plots only show the x, y-components of the solutions and data in the $[-0.1, 0.1] \times [-0.1, 0.1] \times \{0.001\}$ plane and the (G)FEM-dofs located therein.

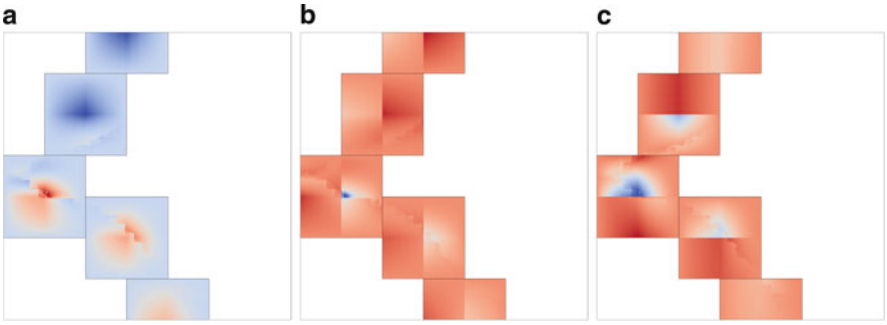


Fig. 5 Some shape functions $\varphi_a(\tilde{\boldsymbol{u}}^1_{\text{local}} \cdot \boldsymbol{e}_2)\boldsymbol{e}_2$ with φ_a obtained from $7 \times 7 \times 1$ mesh. (**a**) $\varphi_a(\tilde{\boldsymbol{u}}^1_{\text{local}} \cdot \boldsymbol{e}_2)$. (**b**) $\frac{\partial}{\partial x}\left(\varphi_a(\tilde{\boldsymbol{u}}^1_{\text{local}} \cdot \boldsymbol{e}_2)\right)$. (**c**) $\frac{\partial}{\partial y}\left(\varphi_a(\tilde{\boldsymbol{u}}^1_{\text{local}} \cdot \boldsymbol{e}_2)\right)$

The employed PU is obtained from trilinear hexahedral elements. The elements are directly used as quadrature cells and the number of quadrature nodes M is given per element. Since this particular partition of unity does not feature a flat top [25], care must be taken to avoid stability issues. Hence, we follow the Stabilized GFEM (SGFEM) [15] approach to achieve a near-orthogonalization through the subtraction of the FE interpolant of the enrichment

$$\tilde{\boldsymbol{u}}^1_{\text{local}}(\boldsymbol{x}) = \boldsymbol{u}^1_{\text{local}}(\boldsymbol{x}) - \sum_{a \in A} \varphi_a(\boldsymbol{x}) \boldsymbol{u}^1_{\text{local}}(\boldsymbol{x}_a) \,, \tag{18}$$

see Fig. 5, where $\boldsymbol{x}_a$ is the node φ_a is based in.

As mentioned above, we have $\Omega^0_{\text{local}} = \Omega$, but still want to mimic most of an actual global-local iteration. Still, not all shape functions $\varphi_a \tilde{\boldsymbol{u}}^1_{\text{local}}$ are equally useful, in particular with the displacement $\boldsymbol{u}^1_i$ having many zeroes. Hence, we are faced with the "dual" problem of making a selection of useful enrichment segments. Since all trilinear polynomials are already recovered with our FE-PU $(\varphi_a)_a$, the leading term g of any higher order function, i.e. the likely leading term of the enriched PUM shape functions $\varphi_a \boldsymbol{\eta}^1_{a,b}$, $\boldsymbol{\eta}^1_{a,b} \in \mathcal{E}^1_a$ after applying the SGFEM transformation $\tilde{\boldsymbol{\eta}}^1_{a,b}$ to the enrichments $\boldsymbol{\eta}^1_{a,b}$, on the hexahedral element supported on a reference cell $E = [0, h_x] \times [0, h_y], \times [0, h_z]$ can be written as

$$g(x, y, z) = C \left(\frac{x}{h_x} \frac{y}{h_y} \frac{z}{h_z} \right)^2 \qquad \text{with} \qquad C \in \mathbb{R} \,.$$

Since we assemble $\mathbf{M}^{1,1}$ and hence compute $\|\boldsymbol{\phi}\|^2_2, \boldsymbol{\phi} \in \mathcal{B}^1$ anyway, we consider

$$\|g\|^2_2 = C^2 \int_0^{h_x} \int_0^{h_y} \int_0^{h_z} \left(\frac{x}{h_x} \frac{y}{h_y} \frac{z}{h_z} \right)^4 \mathrm{d}x \, \mathrm{d}y \, \mathrm{d}z = \frac{C^2}{h_x^4 h_y^4 h_z^4} \frac{h_x^5 h_y^5 h_z^5}{125} = \frac{C^2 V}{125} \,.$$

Table 1 Color coding whether $\varphi_a(\boldsymbol{u}^1_{\text{local}} \cdot \boldsymbol{e}_c)\boldsymbol{e}_c \in \mathcal{B}^1$, i.e. if $(\boldsymbol{u}^1_{\text{local}} \cdot \boldsymbol{e}_c)\boldsymbol{e}_c \in \mathcal{E}^1_a$

•	no problem-dependent enrichment,	$\mathcal{E}^1_a = \{0\}$.
•	x-component enriched,	$\mathcal{E}^1_a = \text{span}\{(\tilde{\mathbf{u}}^1_{\text{local}} \cdot \mathbf{e}_1)\mathbf{e}_1\}$.
•	y-component enriched,	$\mathcal{E}^1_a = \text{span}\{(\tilde{\mathbf{u}}^1_{\text{local}} \cdot \mathbf{e}_2)\mathbf{e}_2\}$.
•	x, y-components enriched,	$\mathcal{E}^1_a = \text{span}\{(\tilde{\mathbf{u}}^1_{\text{local}} \cdot \mathbf{e}_1)\mathbf{e}_1, (\tilde{\mathbf{u}}^1_{\text{local}} \cdot \mathbf{e}_2)\mathbf{e}_2\}$.

where $V = h_x h_y h_z$ denotes the volume of E. With $|C|$ being the maximum of g on E, we use

$$C_{a,c} = \sqrt{\frac{125 \left\| \varphi_a(\tilde{\boldsymbol{u}}^1_{\text{local}} \cdot \boldsymbol{e}_c) \right\|_2^2}{|\operatorname{supp} \varphi_a|}} \tag{19}$$

as a lower cutoff for (useful) shape functions $\varphi_a(\tilde{\boldsymbol{u}}^1_{\text{local}} \cdot \boldsymbol{e}_c)\boldsymbol{e}_c \in \mathcal{V}^{\text{PU},1}_{\text{global}}$. By construction, this quantity is dimensionless and independent of the spacing of the chosen FE mesh. In the subsequent examples, see Figs. 6, 7, 8 and 9, we made a selection based on $C_{a,c} > 1$, where 1 was chosen empirically. Note that this 1 does not reflect the actual displacement obtained from PD, but is also influenced by the chosen scaling of φ_a for computational purposes (as shape functions, often such that $\|\varphi_a\|_2 = 1$).

The enrichment $\tilde{\boldsymbol{u}}^1_{\text{local}}$ decouples into shape functions $\boldsymbol{\phi}^1_d$, $D^0 < d \leq D^1$ for the separate dimensions $\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3$ of our field problem $\boldsymbol{u} : \Omega \subseteq \mathbb{R}^3 \to \mathbb{R}^3$ via the product

$$\boldsymbol{\phi}^1_d = \varphi_a \left(\tilde{\boldsymbol{u}}^1_{\text{local}} \cdot \boldsymbol{e}_c \right) \boldsymbol{e}_c$$

Hence, particular nodes $\boldsymbol{x}_a$ may feature enrichment for just one, or several dimensions $\boldsymbol{e}_c$. Note that, in this example, no enrichment for z, i.e. $(\tilde{\boldsymbol{u}}^1_{\text{local}} \cdot \boldsymbol{e}_3)\boldsymbol{e}_3$, will appear because of (19). Table 1 gives the color coding of the nodal enrichment status employed in the following plots.

The PU is obtained from a (thin) regular hexahedral mesh with $3 \times 3 \times 1$, $4 \times 4 \times 1$, $7 \times 7 \times 1$ and $8 \times 8 \times 1$ elements. These sizes are chosen to both visualize possible effects from part of the cracks aligning with element borders and varying mesh sizing. Because of the lack of dynamics in z direction, we limit ourselves to plotting only data from $[-0.1, 0.1] \times [-0.1, 0.1] \times \{0.001\}$

Figure 5 shows some of the shape function $\varphi_a(\tilde{\boldsymbol{u}}^1_{\text{local}} \cdot \boldsymbol{e}_2)\boldsymbol{e}_2$ computed for the example case using $7 \times 7 \times 1$ elements. Figures 6, 7, 8 and 9 depict the enrichment state according to Table 1 of the FE dofs. Figures 10, 11, 12 and 13 show the convergence histories of the relative errors of the assembled entries.

It appears, as if the special features of this particular problem allow for a higher order of convergence than suggested by the $O(M^{-\frac{1}{d}})$ estimate. Contrary to the example depicted in Fig. 3g of the previous subsection, the branching cracks does not enforce this estimate to hold sharply. But, neither the usual $O(M^{-\frac{4}{d}})$

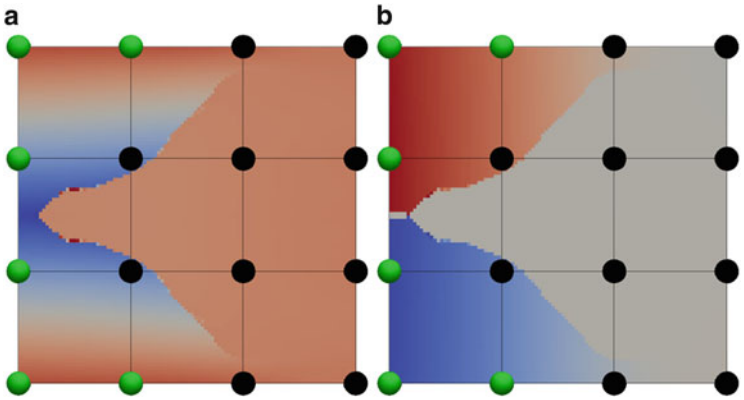


Fig. 6 x, y-Components from $\boldsymbol{u}^1_{\text{local}}$ with underlying FE-mesh and dofs from $3 \times 3 \times 1$ elements. Color coding shows enrichment status of dof as in Table 1

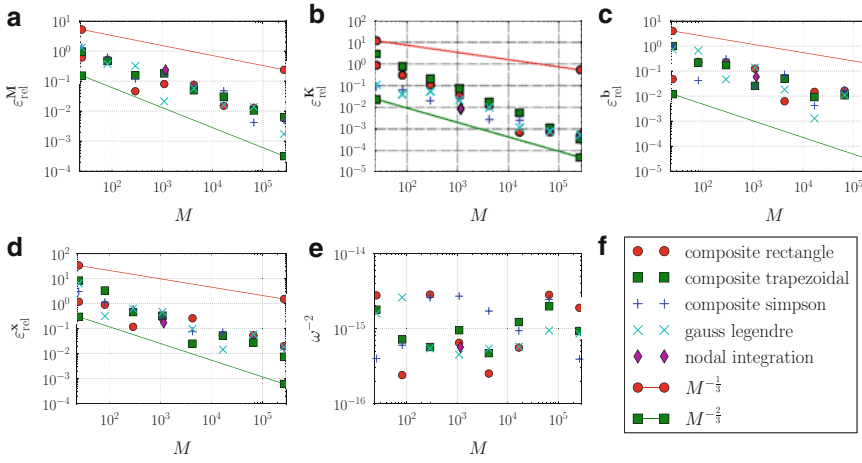


Fig. 7 x, y-Components from $\boldsymbol{u}^1_{\text{local}}$ with underlying FE-mesh and dofs from $4 \times 4 \times 1$ elements. Color coding shows enrichment status of dof as in Table 1

associated with the composite Simpson rule nor the spectral convergence of the Gauss-Legendre rule are attained.

The timestep Δt in (9) is limited by $\sqrt{\omega^{-2}}$ [19], with ω^2 being the largest eigenvalue of the generalized eigenvalue problem $\mathbf{Ky} = \omega^2 \mathbf{My}$. The plots of ω^{-2} suggest that, for the presented example, there is no strong correlation between the quadrature error in the system assembly and the attained critical timestep.

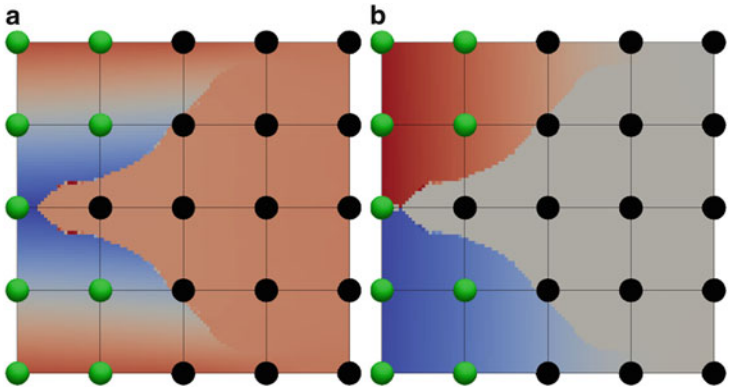


Fig. 8 x, y-Components from $\boldsymbol{u}^1_{\text{local}}$ with underlying FE-mesh and dofs from $7 \times 7 \times 1$ elements. Color coding shows enrichment status of dof as in Table 1

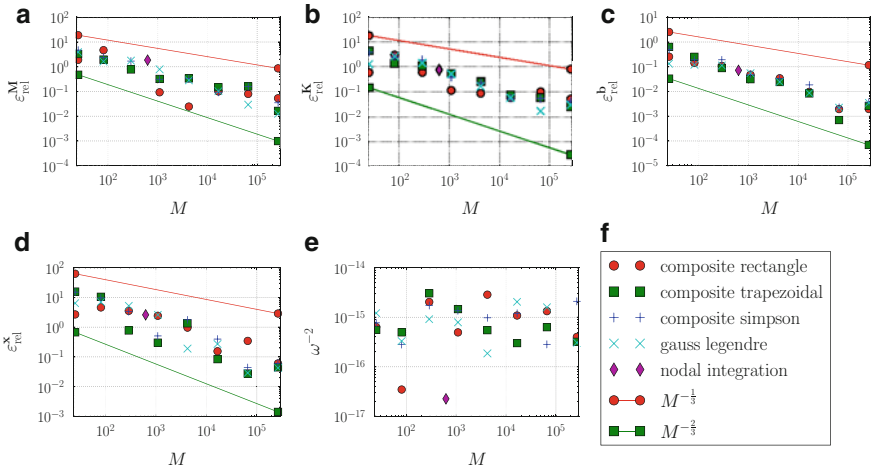


Fig. 9 x, y-Components from $\boldsymbol{u}^1_{\text{local}}$ with underlying FE-mesh and dofs from $8 \times 8 \times 1$ elements. Color coding shows enrichment status of dof as in Table 1

5 Summary and Outlook

We presented a PUM with embedded PD for the simulation of fracture nucleation and growth with on-the-fly computed enrichments. The results of our numerical experiments showed that, even though the theoretical bound $O(M^{-\frac{1}{d}})$ convergence for the integration error of discontinuous functions is sharp, the impact of numerical integration in the chosen application setting is less pronounced.

But, even with the suggested $O(M^{-\frac{2}{d}})$ accuracy, numerical integration comes with a high cost, i.e. necessitating a large number M of quadrature nodes when

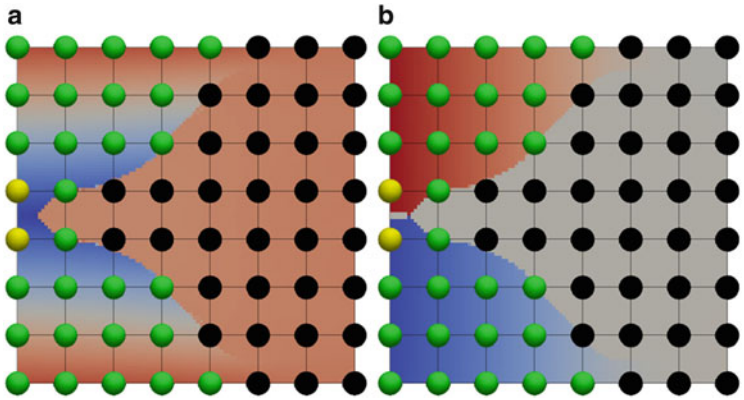


Fig. 10 Convergence histories for 3×3×1 elements. (**a**) Mass matrix **M** errors. (**b**) Stiffness matrix **K** errors. (**c**) Load vector **b** errors. (**d**) Errors of $\mathbf{M}^{-1}\mathbf{b}$. (**e**) Maximum eigenvalue of $\mathbf{Ky} = \omega^2\mathbf{My}$. (**f**) Legend

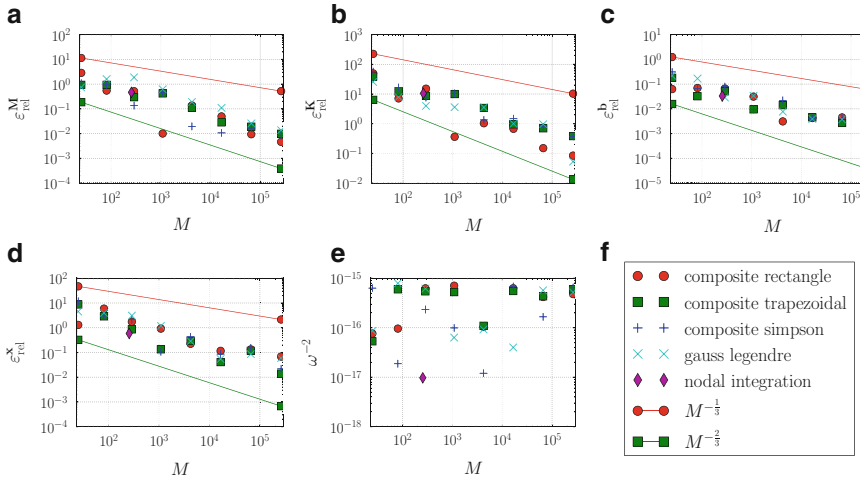


Fig. 11 Convergence histories for 4×4×1 elements. (**a**) Mass matrix **M** errors. (**b**) Stiffness matrix **K** errors. (**c**) Load vector **b** errors. (**d**) Errors of $\mathbf{M}^{-1}\mathbf{b}$. (**e**) Maximum eigenvalue of $\mathbf{Ky} = \omega^2\mathbf{My}$. (**f**) Legend

using standard quadrature rules. Yet, since these rules are usually much more easy to implement and efficiently parallelize, they might still be an cost-effective option on a massively-parallel-architecture, for instance, when the local PD is run a GPU. This, however, will have to be thoroughly checked in another study.

Otherwise, this study would suggest that we need either an efficient post-processing of the constructed enrichment which gives a good decomposition of the integration domain with respect to the piecewise character of the enrichment or

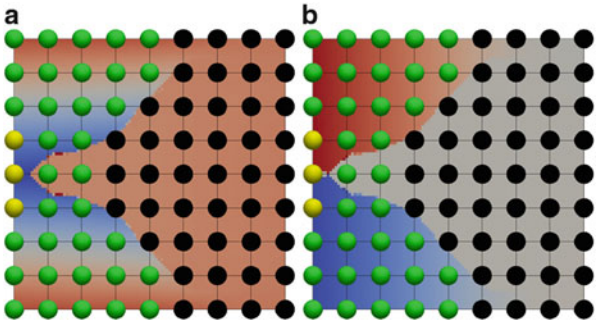


Fig. 12 Convergence histories for 7×7×1 elements. (**a**) Mass matrix **M** errors. (**b**) Stiffness matrix **K** errors. (**c**) Load vector **b** errors. (**d**) Errors of $\mathbf{M}^{-1}\mathbf{b}$. (**e**) Maximum eigenvalue of $\mathbf{Ky} = \omega^2\mathbf{My}$. (**f**) Legend

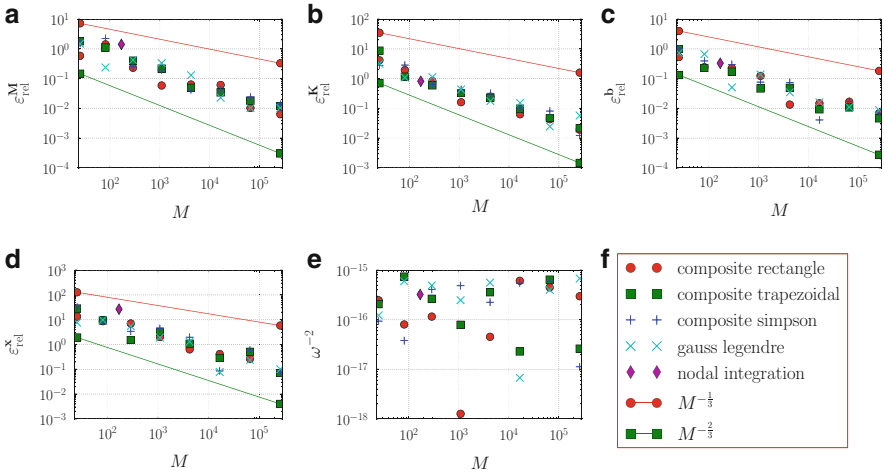


Fig. 13 Convergence histories for 8×8×1 elements. (**a**)Mass matrix **M** errors. (**b**) Stiffness matrix **K** errors. (**c**) Load vector **b** errors. (**d**) Errors of $\mathbf{M}^{-1}\mathbf{b}$. (**e**) Maximum eigenvalue of $\mathbf{Ky} = \omega^2\mathbf{My}$. (**f**) Legend

a pre-processing of the PD data prior to the construction of enrichments. Again, the actual possibilities and benefits of such an approach would have to be investigated.

References

1. A.M. Aragón, C.A. Duarte, P.H. Geubelle, Generalized finite element enrichment functions for discontinuous gradient fields. Int. J. Numer. Methods Eng. **82**(2), 242–268 (2010)
2. E. Askari, F. Bobaru, R.B. Lehoucq, M.L. Parks, S.A. Silling, O. Weckner, Peridynamics for multiscale materials modeling. J. Phys.: Conf. Ser. **125**(1), 012078 (2008)

3. I. Babuška, U. Banerjee, J.E. Osborn, Meshless and generalized finite element methods: a survey of some major results, in *Meshfree Methods for Partial Differential Equations*, ed. by M. Griebel, M.A. Schweitzer. Lecture Notes in Computational Science and Engineering, vol. 26 (Springer, Berlin/Heidelberg, 2003), pp. 1–20
4. I. Babuska, R. Lipton, Optimal local approximation spaces for generalized finite element methods with application to multiscale problems. Multiscale Model. Simul. **9**(1), 373–406 (2011)
5. T. Belytschko, Y.Y. Lu, L. Gu, Element-free Galerkin methods. Int. J. Numer. Methods Eng. **37**(2), 229–256 (1994)
6. J. Dolbow, T. Belytschko, Numerical integration of the galerkin weak form in meshfree methods. Comput. Mech. **23**(3), 219–230 (1999)
7. C.A. Duarte, D.J. Kim, Analysis and applications of a generalized finite element method with global–local enrichment functions. Comput. Methods Appl. Mech. Eng. **197**(6–8), 487–504 (2008)
8. C.A.M. Duarte, J.T. Oden, An h-p adaptive method using clouds. Comput. Methods Appl. Mech. Eng. **139**(1–4), 237–262 (1996)
9. C.C. Farrington, Numerical quadrature of discontinuous functions, in *Proceedings of the 1961 16th ACM National Meeting, ACM '61*, New York (ACM, 1961), pp. 21.401–21.404
10. R. Farwig, Multivariate interpolation of arbitrarily spaced data by moving least squares methods. J. Comput. Appl. Math. **16**(1), 79–93 (1986)
11. T.-P. Fries, T. Belytschko, The extended/generalized finite element method: an overview of the method and its applications. Int. J. Numer. Methods Eng. **84**(3), 253–304 (2010)
12. M. Griebel, J. Oettershagen, Dimension-adaptive sparse grid quadrature for integrals with boundary singularities, in *Sparse Grids and Applications*, Munich 2012, ed. by J. Garcke, D. Pflüger. Lecture Notes in Computational Science and Engineering, vol. 97 (Springer, Dordrecht, 2014). ISBN:978-3-319-04536-8. Also available as INS Preprint no 1310, pp. 109–136
13. M. Griebel, M.A. Schweitzer, A particle-partition of unity method for the solution of elliptic, parabolic and hyperbolic PDE. SIAM J. Sci. Comput. **22**(3), 853–890 (2000)
14. M. Griebel, M.A. Schweitzer, Particle-partition of unity methods in elasticity, in *Proceedings of the International Conference on Computational Methods* (Springer, Dordrecht, 2005), pp. 1483–1493. ISBN:978-1-4020-3952-2
15. V. Gupta, C.A. Duarte, I. Babuška, U. Banerjee, A stable and optimally convergent generalized FEM (SGFEM) for linear elastic fracture mechanics. Comput. Methods Appl. Mech. Eng. **266**(0), 23–39 (2013)
16. V. Gupta, D.-J. Kim, C.A. Duarte, Analysis and improvements of global–local enrichments for the generalized finite element method. Comput. Methods Appl. Mech. Eng. **245–246**, 47–62 (2012)
17. D.-J. Kim, C.A. Duarte, S.P. Proenca, Generalized finite element method with global-local enrichments for nonlinear fracture analysis, in *Mechanics of Solids in Brazil 2009*, ed. by M.H.S. da Costa, M. Alves (Brazilian Society of Mechanical Sciences and Engineering, 2009). ISBN:978-85-85769-43-7
18. D.-J. Kim, C.A. Duarte, S.P. Proenca, A generalized finite element method with global-local enrichment functions for confined plasticity problems. Comput. Mech. **50**(5), 563–578 (2012)
19. J.R. Koteras, R.B. Lehoucq, Estimating the critical time-step in explicit dynamics using the lanczos method. Int. J. Numer. Methods Eng. **69**(13), 2780–2788 (2007)
20. P. Lancaster, K. Salkauskas, Surfaces generated by moving least squares methods. Math. Comput. **37**(155), 141–158 (1981)
21. D. Levin, The approximation power of moving least-squares. Math. Comput. **67**(224), 1517–1531 (1998)
22. S. Li, W.K. Liu, *Meshfree Particle Methods* (Springer, Berlin/New York, 2004)
23. T.J. Liszka, C.A. Duarte, W.W. Tworzydlo, HP-meshless cloud method. Comput. Methods Appl. Mech. Eng. **139**(1–4), 263–288 (1996)
24. D.H. McLain, Drawing contours from arbitrary data points. Comput. J. **17**(4), 318–324 (1974)

25. J.M. Melenk, I. Babuška, The partition of unity finite element method: basic theory and applications. Comput. Methods Appl. Mech. Eng. **139**(1–4), 289–314 (1996)
26. N. Moës, J.E. Dolbow, T. Belytschko, A finite element method for crack growth without remeshing. Int. J. Numer. Methods Eng. **46**(1), 131–150 (1999)
27. S.E. Mousavi, N. Sukumar, Generalized Gaussian quadrature rules for discontinuities and crack singularities in the extended finite element method. Comput. Methods Appl. Mech. Eng. **199**(49–52), 3237–3249 (2010)
28. S.E. Mousavi, N. Sukumar, Numerical integration of polynomials and discontinuous functions on irregular convex polygons and polyhedrons. Comput. Mech. **47**(5), 535–554 (2011)
29. J.T. Oden, C.A.M. Duarte, O.C. Zienkiewicz, A new cloud-based HP finite element method. Comput. Methods Appl. Mech. Eng. **153**(1–2), 117–126 (1998)
30. M.L. Parks, R.B. Lehoucq, S.J. Plimpton, S.A. Silling, Implementing peridynamics within a molecular dynamics code. Comput. Phys. Commun. **179**(11), 777–783 (2008)
31. J.C.J. Schellekens, R. De Borst, On the numerical integration of interface elements. Int. J. Numer. Methods Eng. **36**(1), 43–66 (1993)
32. M.A. Schweitzer, *A Parallel Multilevel Partition of Unity Method for Elliptic Partial Differential Equations*. Lecture Notes in Computational Science and Engineering, vol. 29 (Springer, Berlin/New York, 2003)
33. M.A. Schweitzer, Generalizations of the finite element method. Cent. Eur. J. Math. **10**(1), 3–24 (2012)
34. M.A. Schweitzer, S. Wu, A moving least squares approach to the construction of discontinuous enrichment functions, in *Singular Phenomena and Scaling in Mathematical Models*, ed. by M.Griebel (Springer International Publishing, Cham, 2014), pp. 347–360
35. D. Shepard, *A Two-Dimensional Interpolation Function for Irregularly-Spaced Data. Proceedings of the 1968 23rd ACM national conference, ACM '68*, New York (ACM, 1968), pp. 517–524
36. S.A. Silling, Reformulation of elasticity theory for discontinuities and long-range forces. J. Mech. Phys. Solids **48**(1), 175–209 (2000)
37. S.A. Silling, M. Epton, O. Weckner, J. Xu, E. Askari, Peridynamic states and constitutive modeling. J. Elast. **88**(2), 151–184 (2007)
38. S.A. Silling, R.B. Lehoucq, Convergence of peridynamics to classical elasticity theory. J. Elast. **93**(1), 13–37 (2008)
39. A. Simone, Partition of unity-based discontinous elements for interface phenomena: computational issues. Commun. Numer. Meth. Eng. **20**(6), 465–478 (2004)
40. G. Strang, G.J. Fix, *An Analysis of the Finite Element Method*. Prentice-Hall Series in Automatic Computation (Prentice-Hall, Englewood Cliffs, 1973)
41. A.-K. Tornberg, Multi-dimensional quadrature of singular and discontinuous functions. BIT Numer. Math. **42**(3), 644–669 (2002)
42. H. Wendland, Local polynomial reproduction and moving least squares approximation. IMA J. Numer. Anal. **21**, 285–300 (2001)
43. J.D. Whitcomb, K. Woo, Application of iterative global/local finite-element analysis. Part 1: linear analysis. Commun. Numer. Methods Eng. **9**(9), 745–756 (1993)
44. J.D. Whitcomb, K. Woo, Application of iterative global/local finite-element analysis. Part 2: geometrically non-linear analysis. Commun. Numer. Methods Eng. **9**(9), 757–766 (1993)

Dispersion Properties of the Partition of Unity Method and Explicit Dynamics

Marc Alexander Schweitzer and Albert Ziegenhagel

Abstract In this paper, we focus on the dispersion properties of the partition of unity method. To this end, we consider the semi-discretized wave equation and determine the resulting numerical phase and group velocities when using a consistent mass matrix and a lumped mass matrix. Our results clearly show that the vartiational mass lumping approach due to Schweitzer (SIAM J Sci Comput 35(2):A1073–A1097, 2013) yields comparable dispersion properties than the consistent mass matrix approach. Thus, the lumped mass matrix approach attains very similar accuracy at substantially lower computational cost than the consistent mass matrix in simulations of dynamic behavior.

Keywords Partition of unity • Dispersion • Lumped mass

1 Introduction

Wave propagation problems arise in many applications and thus their efficient and accurate approximation is of great practical interest. The classical finite element method (FEM), however, is not very effective for such problems due to the so-called pollution effect [2, 5, 12, 13] which is a numerical error essentially due to a phase lag of the approximate solution. In the generalized finite element method (GFEM) and the partition of unity method (PUM) this error can be completely eliminated by appropriate enrichment, yet only in one dimensional problems [3,13]. In the multi-dimensional case, the number of particular solutions is infinite so that enrichment by a finite number of functions cannot remove all dispersion errors. However, an exponential convergence may be attained [3, 19–21].

Another issue of practical relevance in wave propagation problems is the appropriate discretization of time. Often, explicit time stepping schemes in conjunction with a lumped mass matrix are employed in practice since they usually

M.A. Schweitzer (✉) • A. Ziegenhagel
Fraunhofer SCAI, Schloss Birlinghoven, 53754, Sankt Augustin, Germany
e-mail: marc.alexander.schweitzer@scai.fraunhofer.de; aziegenhagel@web.de

M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations VII*, Lecture Notes in Computational Science and Engineering 100,
DOI 10.1007/978-3-319-06898-5_14

provide the shortest time-to-solution. Yet, the use of a lumped mass matrix may have a strong impact on the overall approximation and dispersion properties.

In this paper, we give a detailed study of the dispersion properties of the PUM when using a consistent mass matrix and a lumped mass matrix. As it turns out, the variational mass lumping approach due to [18] does not only yield very good conservation properties but also shows dispersion properties comparable to those of the consistent mass matrix. Moreover, we present some numerical results for elastodynamics in two and three dimensions which clearly demonstrate the overall benefits of using the lumped mass matrix.

The remainder of this paper is structured as follows. After a short summary of the main ingredients of the PUM in Sect. 2, we present the problem setup considered throughout this study in Sect. 2.1. The numerical dispersion analysis of our PUM and the employed extraction approach to determine the numerical phase and group velocities are the topic of Sect. 3. The results of our numerical experiments are presented in Sect. 4 before we conclude with some remarks in Sect. 5.

2 Partition of Unity Method

The partition of unity method (PUM) is a generalized finite element method (GFEM) typically employed for the spatial discretization of a partial differential equation (PDE), see e.g. [8, 17]. The notion of a PUM was coined in [3, 4] and is based on the special finite element methods developed in [1]. The abstract ingredients of a PUM are a partition of unity (PU) $\{\varphi_i : i = 1, \dots, N\}$ and a collection of local approximation spaces $V_i(\omega_i) := \operatorname{span}\langle \vartheta_i^m \rangle_{m=1}^{d_{V_i}}$ defined on the patches $\omega_i := \operatorname{supp}(\varphi_i)$ for $i = 1, \dots, N$ which overlap and whose union covers the computational domain $\Omega \subset \mathbb{R}^d$. With these two ingredients we define the PUM space

$$V^{\mathrm{PU}} := \sum_{i=1}^{N} \varphi_i V_i = \operatorname{span}\langle \varphi_i \vartheta_i^m \rangle; \tag{1}$$

i.e., the shape functions of a PUM space are simply defined as the products of the PU functions φ_i and the local approximation functions ϑ_i^m. The PU functions provide the locality and global regularity of the product functions whereas the functions ϑ_i^m equip V^{PU} with its approximation power.

Throughout this paper, we employ the following construction for the PU functions which is a simplified version of [9, 14]. First we define a cover $C_\Omega := \{\omega_i\}$ of the domain Ω with the help of a uniform regular mesh of mesh-width $2h$ by an isotropic scaling of the mesh cells

$$\mathcal{C}_i = \prod_{l=1}^{d} (o_i^l - h, o_i^l + h),$$

i.e., we define the patches ω_i as

$$\omega_i := \prod_{l=1}^{d} (o_i^l - \alpha h, o_i^l + \alpha h), \quad \text{with } \alpha > 1. \tag{2}$$

To obtain a PU on a cover C_Ω with $N := \text{card}(C_\Omega)$ we define a weight function $W_i : \Omega \to \mathbb{R}$ with $\text{supp}(W_i) = \omega_i$ for each cover patch ω_i by

$$W_i(x) = \begin{cases} \mathcal{W} \circ T_i(x) & x \in \omega_i \\ 0 & \text{else} \end{cases} \tag{3}$$

with the affine transforms $T_i : \overline{\omega}_i \to [-1, 1]^d$ and $\mathcal{W} : [-1, 1]^d \to \mathbb{R}$ the reference d-linear B-spline. By simple averaging of these weight functions we obtain the functions

$$\varphi_i(x) := \frac{W_i(x)}{S(x)}, \quad \text{with} \quad S(x) := \sum_{l=1}^{N} W_l(x) = \sum_{\{l : \omega_l \cap \omega_i \neq \emptyset\}} W_l(x). \tag{4}$$

Note that the PU (4) is non-negative (if the employed weight functions are non-negative) and that the φ_i satisfy the so-called flat-top property for $\alpha \in (1, 2)$, see [10,15]. Due to this construction we can easily control the overlap of the patches by the parameter $\alpha \in (1, 2)$.

For the ease of notation, we make the following conventions. For an arbitrary function $u \in V^{\text{PU}}$ with the basis representation

$$u(x) = \sum_{i=1}^{N} \sum_{m=1}^{d_{V_i}} u_i^m \varphi_i(x) \vartheta_i^m(x) \tag{5}$$

we denote the respective local contributions from $V_i(\omega_i)$ by

$$u_i(x) = \sum_{m=1}^{d_{V_i}} u_i^m \vartheta_i^m(x)$$

and the overall coefficient vector by

$$\tilde{u} = (u_{(i,m)}) \in \mathbb{R}^{d_{\text{PU}}} \text{ with } d_{\text{PU}} := \sum_{i=1}^{N} d_{V_i}. \tag{6}$$

With the help of the vector-valued function

$$\Phi : \Omega \to \mathbb{R}^{d_{\mathrm{PU}}}, \quad \Phi : x \mapsto \Phi(x) := (\varphi_i(x)\vartheta_i^m(x)) \tag{7}$$

we can rewrite (5) as

$$u(x) = \sum_{i=1}^{N} \sum_{m=1}^{d_{V_i}} u_i^m \varphi_i(x)\vartheta_i^m(x) = \langle \tilde{u}, \Phi(x) \rangle_{\mathbb{R}^{d_{\mathrm{PU}}}}. \tag{8}$$

2.1 Model Problem

The model problems considered in this paper, complemented with appropriate boundary and initial conditions, are the scalar wave equation

$$\ddot{u} = c^2 \Delta u \text{ in } \Omega \times (0, T) \tag{9}$$

and the equations of motion in a linearly elastic medium

$$\rho \ddot{u} = \mathbf{div}\, \boldsymbol{\sigma}(u) \text{ in } \Omega \times (0, T) \tag{10}$$

where the stress tensor is given by $\boldsymbol{\sigma}(\boldsymbol{u}) := \mathcal{C}\boldsymbol{\varepsilon}(\boldsymbol{u}) = 2\mu\boldsymbol{\varepsilon}(\boldsymbol{u}) + \lambda \operatorname{trace}(\boldsymbol{\varepsilon}(\boldsymbol{u}))\mathbb{I}$ and the strain tensor is defined as $\boldsymbol{\varepsilon}(u) := \frac{1}{2}(\boldsymbol{\nabla u} + (\boldsymbol{\nabla u})^T)$.

We discretize (9) and (10) by a method of lines approach; i.e., we discretize the PDEs in space with the help of our PUM to obtain the semi-discrete system of ordinary differential equations (ODE)

$$M\ddot{\tilde{u}}(t) = -K\tilde{u}(t) \tag{11}$$

for the coefficient vector $\tilde{u}$ which still is a function of time $t \in [0, T]$. Here M denotes the (consistent) mass matrix and K the usual stiffness matrix respectively. Due to the product structure of our basis functions $\varphi_i \vartheta_i^n$ these matrices are block-structured; i.e. the (consistent) mass matrix for instance is given as

$$M = (M_{(i,n),(j,m)}), \quad M_{(i,n),(j,m)} = \langle \varphi_j \vartheta_j^m, \varphi_i \vartheta_i^n \rangle_{L^2(\Omega)}. \tag{12}$$

Finally, we need to discretize (11) by some time-stepping scheme which essentially requires the solution of a particular system of linear equations

$$A\tilde{u}_{n+1} = \hat{b}_n \tag{13}$$

with some right-hand side $\hat{b}_n$ (involving information from time-step n only) for the coefficient vector $\tilde{u}_{n+1} := \tilde{u}(t_{n+1})$ at time t_{n+1}, i.e. the $n+1$ time-step. With an

implicit time-stepping scheme we have $A = A(M, \delta t, K)$ whereas explicit schemes yield $A = M$ only. Thus, the explicit time-stepping techniques, which usually require a smaller time-step size δt, may provide a more efficient approach if the solution of the respective linear system $M\tilde{u}_{n+1} = \hat{b}_n$ can be computed much more efficiently than for the implicit methods. To this end, the (consistent) mass matrix M is often approximated by a so-called lumped mass matrix $\bar{M}$ which allows for an extremely efficient solution of $\bar{M}\tilde{u}_{n+1} = \hat{b}_n$. For our PUM such an approximation $\bar{M}$ to the consistent mass matrix M was presented in [18] and is defined with the help of a localized inner product

$$\langle f, \varphi_i \vartheta_i^n \rangle_{L^2(\Omega)} = \int_\Omega f \varphi_i \vartheta_i^n \, dx = \int_{\Omega \cap \omega_i} f \varphi_i \vartheta_i^n \, dx =: \langle f | \varphi_i | \vartheta_i^n \rangle_{L^2(\Omega \cap \omega_i)},$$

by

$$\bar{M} := (\bar{M}_{(i,n),(j,m)}), \quad \bar{M}_{(i,n),(j,m)} := \begin{cases} 0 & i \neq j \\ \langle \vartheta_i^m | \varphi_i | \vartheta_i^n \rangle_{L^2(\Omega \cap \omega_i)} & i = j \end{cases} \tag{14}$$

which is a symmetric positive definite block-diagonal matrix. Thus, with the help of $A = \bar{M}$ the coefficient vector $\tilde{u}_{n+1}$ in (13) can be computed very efficiently and in parallel whereas the solution of (13) with $A = M$ in each time-step is in general prohibitively expensive. For a detailed study of the approximation and conservation properties of this lumped mass matrix as well as its impact on the critical time-step size see [18].

3 Dispersion Analysis

The numerical treatment of time-dependent problems such as (9) and (10) often gives rise to a so-called dispersion error. The characterization of this error is most natural in the context of the Helmholtz problem where it is also referred to as the pollution error [2, 5, 12, 13]. To this end, let us consider a Helmholtz problem with wave number $\|k\|$ and let the solution be $u \in V$. The discrete Galerkin solution $u_{\mathrm{PU}} \in V^{\mathrm{PU}} \subset V$ satisfies the asymptotic error bound

$$\|u - u_{\mathrm{PU}}\|_V \leq C \inf_{v_{\mathrm{PU}} \in V^{\mathrm{PU}}} \|u - v_{\mathrm{PU}}\|_V$$

where the generic *constant* C actually is a function of the wave number $\|k\|$ and usually of the form

$$C(\|k\|) = C_1 + C_2 \|k\|^\beta (h\|k\|)^\gamma$$

with $\beta > 0$, $\gamma \geq 0$, C_1 and C_2 independent of $\|k\|$ and h. Thus, the second term may dominate the overall error in the pre-asymptotic range when $h\|k\|$ is large, i.e. for large wave numbers $\|k\|$.

The common tool to study the dispersive properties of a specific numerical scheme is a von Neumann analysis [6, 7, 22] which is applied to the semi-discrete PDE (11). Since plane waves

$$q(x,t) = q_0 \exp(\imath(\langle x,k\rangle - \omega t + \phi)), \tag{15}$$

where q_0 refers to the amplitude, $\|k\|$ denotes the wave number, $k/\|k\|$ the direction and ω gives to the angular frequency of the wave, satisfy (9) for any wave number on an infinite span with the dispersion relation

$$\omega(\|k\|) = c\|k\|,$$

we can compare any plane wave to its semi-discrete counterpart, i.e. the solutions to the semi-discrete equation (11), (away from the boundary $\partial\Omega$ of the computational domain Ω) to obtain an error measure associated with the spatial discretization only.

For spatial discretization schemes with interpolatory basis functions the common approach is to set the entries of the coefficient vector $\tilde{d}$ in (11), which are functions of time $t \in [0,T]$, simply to

$$d_j(t) = q_0 \exp(\imath(\langle x_j,k\rangle - \breve{\omega} t + \phi)); \tag{16}$$

i.e. to the function values of a plane wave with the same wave number $\|k\|$ but unknown angular frequency $\breve{\omega}$. For such coefficient vectors $\tilde{d}$ we can easily compute the time-derivatives and (11) simply becomes

$$-\breve{\omega}^2 M\tilde{d}(t) + K\tilde{d}(t) = 0 \tag{17}$$

which must hold for any time t, arbitrary amplitudes q_0 and shifts ϕ in (15). Thus, we redefine the entries of the coefficient vector $\tilde{d}$ as

$$d_j := \exp(\imath\langle x_j,k\rangle) \tag{18}$$

by choosing $q_0 = 1$, and $t = \phi = 0$ and can then compute the numerical angular frequency $\breve{\omega}$ of the semi-discrete solution via

$$\breve{\omega}_j = \sqrt{\frac{(K\tilde{d})_j}{(M\tilde{d})_j}}, \tag{19}$$

where $(K\tilde{d})_j$ denotes the jth entry of the respective vector $K\tilde{d}$, for any interior node x_j. Observing now, that the rows of the matrices M and K are constant

for all interior nodes x_j when the nodes x_j come from a uniform grid, we can obtain a unique numerical angular frequency $\breve{\omega}$ for all interior nodes x_j. To this end, we consider an appropriately shifted plane wave $q_0 \exp(\imath \langle x - x_j, k \rangle)$ in (18) for each such j so that the values $(K\tilde{d})_j$ and $(M\tilde{d})_j$ and thus $\breve{\omega}_j$ in (19) become independent of j on a uniform grid.

With the help of the numerical angular frequency $\breve{\omega}$ we can then obtain values for the numerical phase velocity $\breve{c}$ and the normalized phase velocity ψ respectively,

$$\breve{c} := \frac{\breve{\omega}}{\|k\|}, \quad \psi := \frac{\breve{c}}{c}. \tag{20}$$

Obviously, ψ gives the relative deviation of the numerical phase velocity from the analytical phase velocity and thus defines a measure for the numerical error entirely due to the spatial semi-discretization. The definition of the numerical group velocity and normalized group velocity

$$\breve{c}_g := \partial_{\|k\|} \breve{\omega}, \quad \zeta := \frac{\breve{c}_g}{c} \tag{21}$$

result from an analogue computation (see e.g. [7]) where we make use of the fact that in a non-dispersive medium

$$c = c_g = \frac{\omega}{\|k\|}.$$

3.1 Extraction Methods

For a spatial discretization with the PUM, however, the assumptions employed above are not valid and we need to generalize the extraction of the normalized phase and group velocities ψ and ζ slightly. In essence, in the construction given above we choose a specific point of evaluation (any node x_j of the uniform grid), then we select a plane wave with prescribed wave number $\|k\|$ that is shifted to that specific point of evaluation and interpolate this function (on the uniform grid) at time $t = 0$ to attain a respective coefficient vector $\tilde{d}$ (18). Finally, we solve (17) with this $\tilde{d}$ for $\breve{\omega}$ (just for the particular single row j associated with x_j (19)).

In the PUM the basis functions are in general non-interpolatory and thus the entries of a coefficient vector $\tilde{d}$ cannot be chosen as function values of a particular plane wave. Thus, we need to compute the best-approximation to this plane wave from our discrete space V^{PU} with respect to the H^1-seminorm to obtain a valid coefficient vector $\tilde{d}$; i.e. in general

$$\langle \tilde{d}(t), \Phi(x) \rangle = d(t,x) = q_0 \Pi^{\mathrm{PU}}_{H^1(\Omega)} \exp(\imath(\langle x, k \rangle - \breve{\omega} t + \phi)) \tag{22}$$

where $\Pi^{\mathrm{PU}}_{H^1(\Omega)}$ denotes the respective projection onto V^{PU}. Moreover, an entry d_i^n of this coefficient vector is associated with a particular basis function $\varphi_i \vartheta_i^n$ only and not with a specific point of evaluation since we assign a multidimensional approximation space $V_i = \operatorname{span}\langle \vartheta_i^n \rangle$ to a patch ω_i. Thus, we rather need to solve (17) globally instead of the row wise approach pursued above. Thus, we consider

$$-\breve{\omega}^2 \tilde{d}(t) + M^{-1} K \tilde{d}(t) = 0 \tag{23}$$

and interpret this equation for the coefficient vector $\tilde{d} = \tilde{d}(0)$ in $t = 0$ as an equation for the respective discrete function $d \in V^{\mathrm{PU}}$; i.e. we make use of the equality (8). Hence, we obtain a value for the numerical angular frequency for each $x \in \Omega$ by

$$\breve{\omega}(x) = \sqrt{\frac{\langle M^{-1} K \tilde{d}, \Phi(x) \rangle}{\langle \tilde{d}, \Phi(x) \rangle}}, \tag{24}$$

assuming that $d(x) = \langle \tilde{d}, \Phi(x) \rangle \neq 0$.

Throughout this paper we consider PUM spaces (1) based on uniformly refined covers only, compare Fig. 1. Moreover, we assign identical local approximation spaces V_i to each patch ω_i. Thus, it suffices to consider a single interior patch $\omega_i \cap \partial\Omega = \emptyset$. Again, we can choose $q_0 = 1$ and $t = 0$ in (22) as in the construction above. For the phase shift ϕ, however, we consider two cases for a particular patch ω_i to account for the overlap of our cover patches and the fact that we use a flat-top PU. In complete analogy to the approach above we shift the considered wave to the center o_i of the patch ω_i, see (25). Note that $o_i \notin \omega_j$ for all $j \neq i$ and thus the approximation at o_i is entirely due to the single local approximation space V_i. Then, we consider an additional point o which lies in the overlap of neighboring patches,

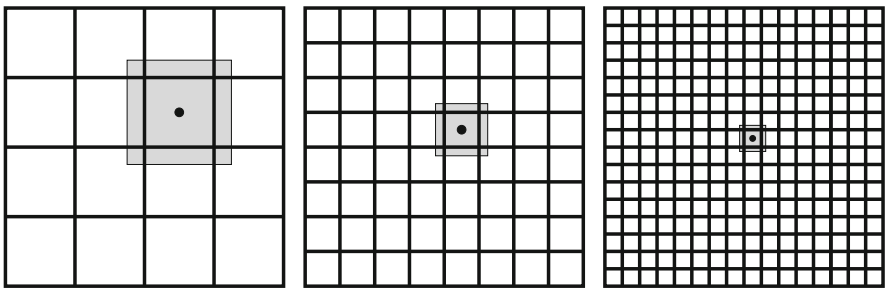


Fig. 1 Schematic of a sequence of uniformly refined covers which come from the scaling of uniform grid cells. Depicted is a single cover patch $\omega_i = \prod_{l=1}^{d} (o_i^l - \alpha h, o_i^l + \alpha h)$ (*gray*) with $2h = 1/4, 1/8, 1/16$ (*left* to *right*) and its center o_i

i.e. $o \in \omega_i \cap \omega_j \neq \emptyset$, so that the approximation at o is coming from multiple overlapping approximation spaces V_j with $o \in \omega_j$.

In summary, we obtain the numerical phase (31) and group velocities (32) associated with our PUM via the following approach where we in fact consider the real part of a plane wave only.

- Choose an interior patch $\omega_i = \prod_{l=1}^{d}(o_i^l - \alpha h, o_i^l + \alpha h)$ and $o \in \omega_i$ such that $o \in \omega_j$ for some $\omega_j \neq \omega_i$.
- Choose the wave number $\|k\|$ (i.e. the wave vector k) and define the (real) plane waves

$$w_{pc}(x) := \cos(\langle x - o_i, k\rangle), \quad w_{dc}(x) := \cos(\langle x - o, k\rangle) \tag{25}$$

and their derivatives with respect to the wave number $\|k\|$

$$\begin{aligned} s_{pc}(x) &:= -\langle x - o_i, {}^{k}/_{\|k\|}\rangle \sin(\langle x - o_i, k\rangle), \\ s_{dc}(x) &:= -\langle x - o, {}^{k}/_{\|k\|}\rangle \sin(\langle x - o, k\rangle). \end{aligned} \tag{26}$$

- Compute the respective best-approximation in V^{PU} with regard to the H^1-seminorm to obtain the associated coefficient vectors

$$\begin{aligned} \langle \tilde{d}_{pc}, \Phi(x)\rangle &:= d_{pc}(x) := \Pi^{\mathrm{PU}}_{H^1(\Omega)} w_{pc}(x), \\ \langle \tilde{d}_{dc}, \Phi(x)\rangle &:= d_{dc}(x) := \Pi^{\mathrm{PU}}_{H^1(\Omega)} w_{dc}(x) \end{aligned} \tag{27}$$

and

$$\begin{aligned} \langle \tilde{b}_{pc}, \Phi(x)\rangle &:= b_{pc}(x) := \Pi^{\mathrm{PU}}_{H^1(\Omega)} s_{pc}(x), \\ \langle \tilde{b}_{dc}, \Phi(x)\rangle &:= b_{dc}(x) := \Pi^{\mathrm{PU}}_{H^1(\Omega)} s_{dc}(x). \end{aligned} \tag{28}$$

- Define the discrete functions

$$\begin{aligned} q_{pc}(x) &:= \langle M^{-1} K \tilde{d}_{pc}, \Phi(x)\rangle, \quad q_{dc}(x) := \langle M^{-1} K \tilde{d}_{dc}, \Phi(x)\rangle, \\ l_{pc}(x) &:= \langle M^{-1} K \tilde{s}_{pc}, \Phi(x)\rangle, \quad l_{dc}(x) := \langle M^{-1} K \tilde{s}_{dc}, \Phi(x)\rangle. \end{aligned} \tag{29}$$

- Compute the numerical angular frequencies $\breve{\omega}$ for each $x \in \omega_i$ via

$$\breve{\omega}_{pc}(x) = \sqrt{\frac{q_{pc}(x)}{d_{pc}(x)}}, \quad \breve{\omega}_{dc}(x) = \sqrt{\frac{q_{dc}(x)}{d_{dc}(x)}}. \tag{30}$$

- Compute the normalized phase velocities ψ as

$$\psi_{pc}(x) = \frac{\breve{\omega}_{pc}(x)}{c\|k\|}, \quad \psi_{dc}(x) = \frac{\breve{\omega}_{dc}(x)}{c\|k\|} \tag{31}$$

and the normalized group velocities ζ as

$$\begin{aligned}\zeta_{pc}(x) &= \frac{1}{2c}\sqrt{\frac{d_{pc}(x)}{q_{pc}(x)}}\frac{d_{pc}(x)l_{pc}(x) - q_{pc}(x)s_{pc}(x)}{d_{pc}^2(x)},\\ \zeta_{dc}(x) &= \frac{1}{2c}\sqrt{\frac{d_{dc}(x)}{q_{dc}(x)}}\frac{d_{dc}(x)l_{dc}(x) - q_{dc}(x)s_{dc}(x)}{d_{dc}^2(x)}.\end{aligned} \tag{32}$$

Obviously, the extraction method for the group velocity (32) is arithmetically more involved than that for the phase velocity (31). Thus, we expect that the results for ζ will show more fluctuations and will be more sensitive to round-off errors.

4 Numerical Examples

In this section we present some numerical results obtained with our PUM for the approximation of our model problems (9) and (10).

At first, we study the dispersion properties of the PUM via the approach presented above. To this end, we consider the semi-discretization of (9) with our PUM on a uniform cover with 32×32 patches and an overlap of $\alpha = 1.3$. As local approximation spaces we choose linear ($p = 1$) or cubic ($p = 3$) Legendre polynomials on all patches only.[1] We consider wave numbers $\|k\|$ with $\|k\| \operatorname{diam}(\omega_i) \lesssim 1.5p$ and compute (31) and (32) for roughly one wavelength of the wave. For the shifts of the plane wave we choose one of the inner most patches ω_i with center $o_i = (0.515625, 0.515625)^T$ which also contains the center $o = (0.5, 0.5)^T$ of the domain $\Omega = (0, 1)^2$ which is in total covered by four patches ω_j. Moreover, we fix the direction $k/\|k\|$ of our plane waves (25) at $(1, 0)^T$ and can therefore limit the extraction of the normalized phase (31) and group velocities (32) to a cross section along the x-axis.

Then, we present some results in two and three space dimensions for the discretization of (10) using an explicit time-stepping by central differences and compare the overall performance and quality of the consistent and lumped mass approaches.

Example 1. In our first experiment, we focus on the extraction scheme where we initially consider the phase velocity (31) only. To this end, we employ our PUM with linear local approximation spaces for the approximation of plane waves with $\|k\| = 12, 24, 36, 48$ (i.e. approximately $17, 8, 6, 4$ patches cover one wavelength) and use a consistent mass matrix.

[1] This is in essence the worst case scenario since an explicit enrichment with particular waves, see e.g. [3, 11, 19–21], will only improve the approximation properties.

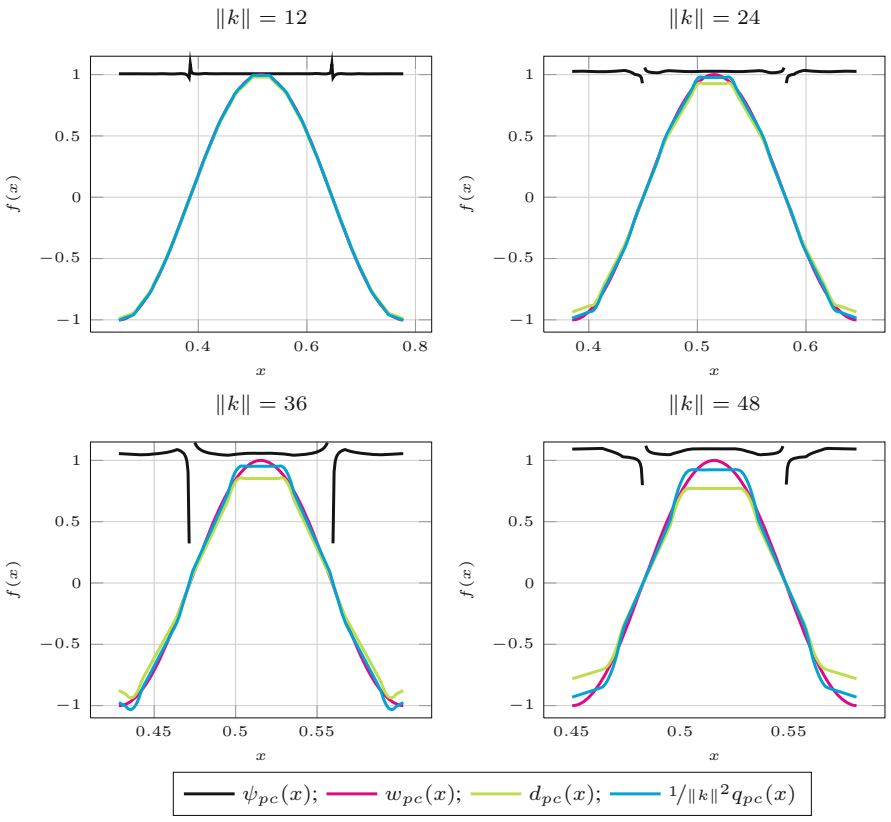


Fig. 2 Extraction of normalized phase velocity ψ_{pc} given in (31) using the consistent mass for a PUM space based on a uniform cover with 32×32 patches and linear local approximation spaces. Depicted are the normalized phase velocity $\psi_{pc}(x)$ (*black*) and the discrete functions $w_{pc}(x)$ (*magenta*), $d_{pc}(x)$ (*lime*), and $1/\|k\|^2 q_{pc}(x)$ (*cyan*) for reference, compare (31)

We give plots of the values $\psi_{pc}(x)$ and $\psi_{dc}(x)$ extracted along a cross section with $y = 0.515625$ and $y = 0.5$ in Figs. 2 and 3 respectively. Moreover, we depict the respective plane wave w (25) and the discrete functions d (27) and $1/\|k\|^2 q$ (29) which are essentially employed in the extraction to allow for a more detailed interpretation of the results.

From these plots we can clearly observe that the extracted values are essentially constant—at least when $|d(x)| > 0$. Comparing the plots for $\|k\| = 48$ in Figs. 2 and 3 we find a clear difference in the approximation when the plane wave is shifted to the overlap region (w_{dc}) or shifted to the center of the patch (w_{pc}). For large wave numbers $\|k\|$; i.e. when only a small number of patches cover one wavelength, the approximation d_{pc} must be flat near the maximum of the wave at the center o_i of the patch ω_i since no element from the local approximation space V_i allows for the necessary change of sign of its first derivative at o_i. The approximation d_{dc}

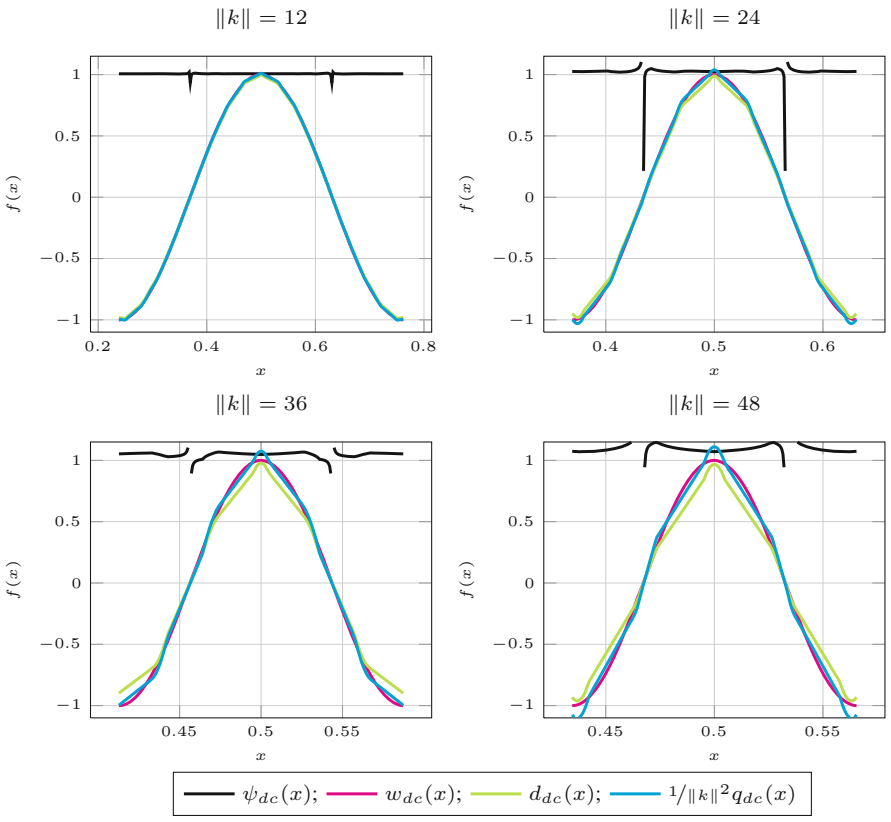


Fig. 3 Extraction of normalized phase velocity ψ_{dc} given in (31) using the consistent mass for a PUM space based on a uniform cover with 32×32 patches and linear local approximation spaces. Depicted are the normalized phase velocity $\psi_{dc}(x)$ (*black*) and the discrete functions $w_{dc}(x)$ (*magenta*), $d_{dc}(x)$ (*lime*), and $1/\|k\|^2 q_{dc}(x)$ (*cyan*) for reference, compare (31)

near the center of the domain o, however, employs not only the space V_i but all V_j with $o \in \omega_j$ so that the change of sign at o can be accomplished. With respect to the normalized phase velocities ψ_{pc} and ψ_{dc}, however, this clear difference in the approximation does not seem to have a substantial effect. To allow for a more detailed comparison we give zoomed plots of just the extracted values ψ_{pc} and ψ_{dc} in Fig. 4. From these plots we find somewhat more variation in ψ_{pc} than in ψ_{dc} which is also slightly closer to the optimal value of 1. Thus, the case where the plane wave is shifted to the overlap region (here the center of the domain o) is somewhat easier and yields slightly better phase velocity results (when using the consistent mass).

Let us now consider the extracted group velocities (32). Figure 5 shows the respective plots for ζ_{pc} and ζ_{dc} obtained for $\|k\| = 24, 36, 48$. Here, we can observe a very similar behavior with the expected increase in the fluctuations due to the more complex extraction expression (32).

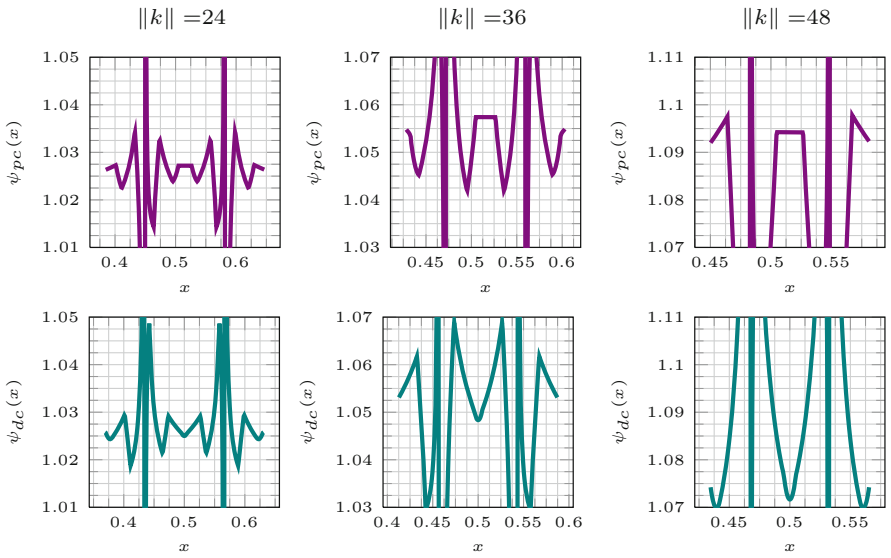


Fig. 4 Zoomed plots of extraction of normalized phase velocity ψ given in (31) using the consistent mass for a PUM space based on a uniform cover with 32×32 patches and linear local approximation spaces. Depicted are the normalized phase velocity $\psi_{pc}(x)$ (*top*) and the normalized phase velocity $\psi_{dc}(x)$ (*bottom*)

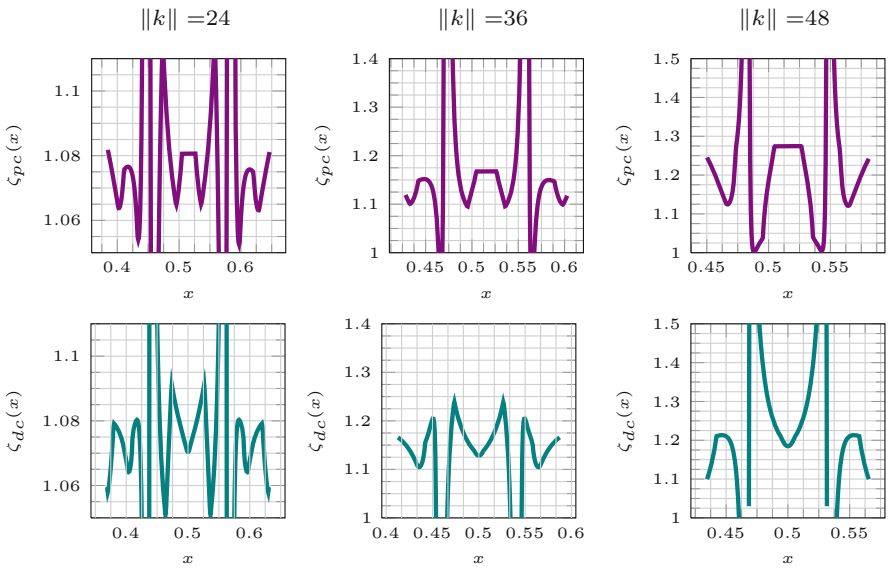


Fig. 5 Zoomed plots of extraction of normalized group velocity ζ given in (32) using the consistent mass for a PUM space based on a uniform cover with 32×32 patches and linear local approximation spaces. Depicted are the normalized group velocity $\zeta_{pc}(x)$ (*top*) and the normalized phase velocity $\zeta_{dc}(x)$ (*bottom*)

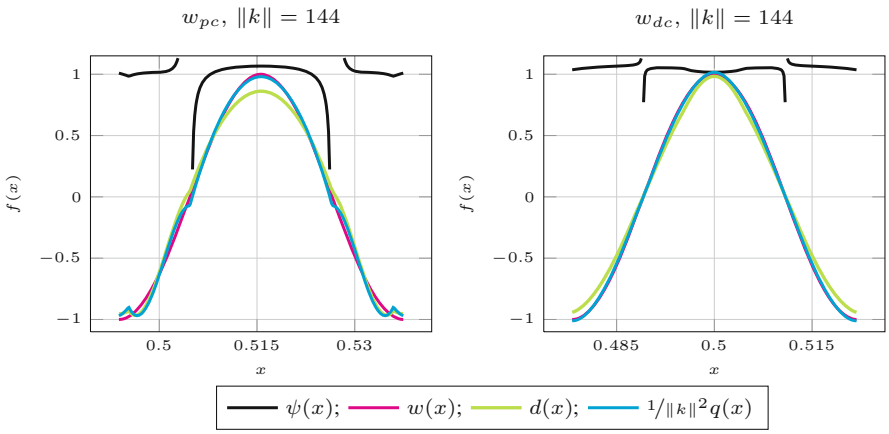


Fig. 6 Extraction of normalized phase velocity ψ given in (31) using the consistent mass for a PUM space based on a uniform cover with 32×32 patches and cubic local approximation spaces. Depicted are the normalized phase velocity $\psi(x)$ (*black*) and the discrete functions $w(x)$ (*magenta*), $d(x)$ (*lime*), and $1/\|k\|^2 q(x)$ (*cyan*) for reference, compare (31). Results for the plane wave w_{pc} shifted to patch center o_i are displayed on the left, the results for plane wave shifted to domain center o are given on the right

Before we discuss the overall dispersion properties of this configuration let us first turn to the case of higher order local approximation spaces.

Example 2. In this example we now employ cubic local approximation spaces and consider wave numbers $\|k\| \leq 144$ on our 32×32 cover. Thus, roughly a single patch covers the wavelength of a wave with $\|k\| = 144$. In Fig. 6 we give the extraction details for w_{pc} and w_{dc} with $\|k\| = 144$. Again, we find an almost constant behavior of the extracted values and d_{dc} as before provides a slightly better approximation to w_{dc} than d_{dc} to w_{pc} which is due to the use of multiple overlapping approximation spaces in the domain center o. With respect to the extracted phase and group velocities, however, there is no substantial difference between the two cases. Figures 7 and 8 show the comparison of the extracted values ψ_{pc}, ψ_{dc} and ζ_{pc}, ζ_{dc} respectively. Also in these plots, we can observe the difference in the approximations near the maximum of the considered waves due to the overlap of the patches. The function ψ_{pc} is concave in the vicinity of o_i whereas ψ_{dc} is convex near the center of the domain o. Moreover, we also find that the values ψ_{dc} are slightly better (near o) than those of ψ_{pc} (near o_i).

The differences in the behavior of the extracted group velocities ζ_{pc} and ζ_{dc} are even more characteristic. For $\|k\| = 72$ $\zeta_{dc}(o) < 1$ whereas $\zeta_{pc}(o_i) \geq 1$ for all wave numbers.

In summary, both experiments showed that the above extraction approach yields very detailed results since we construct functions $\psi(\cdot)$ and $\zeta(\cdot)$ instead of just single values. However, for an easy and compact presentation of the dispersion properties

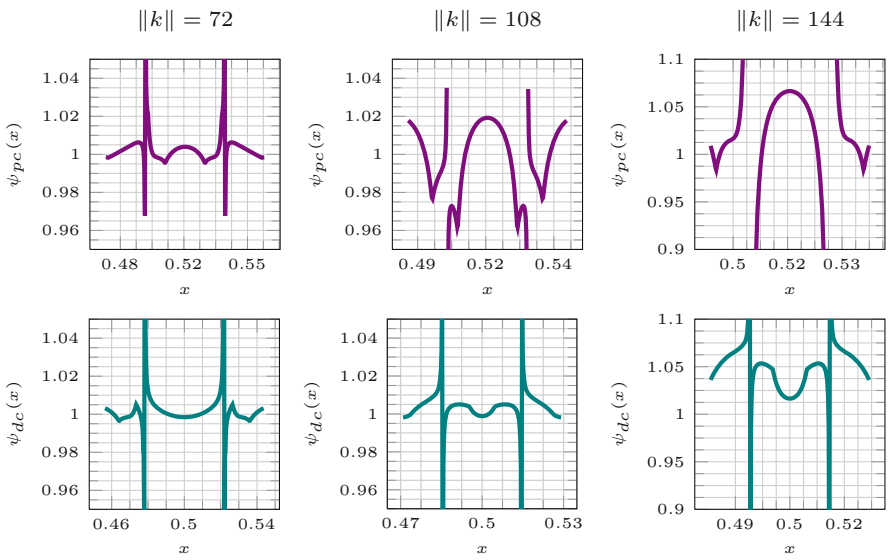


Fig. 7 Zoomed plots of extraction of normalized phase velocity ψ given in (31) using the consistent mass for a PUM space based on a uniform cover with 32×32 patches and cubic local approximation spaces. Depicted are the normalized phase velocity $\psi_{pc}(x)$ (*top*) and the normalized phase velocity $\psi_{dc}(x)$ (*bottom*)

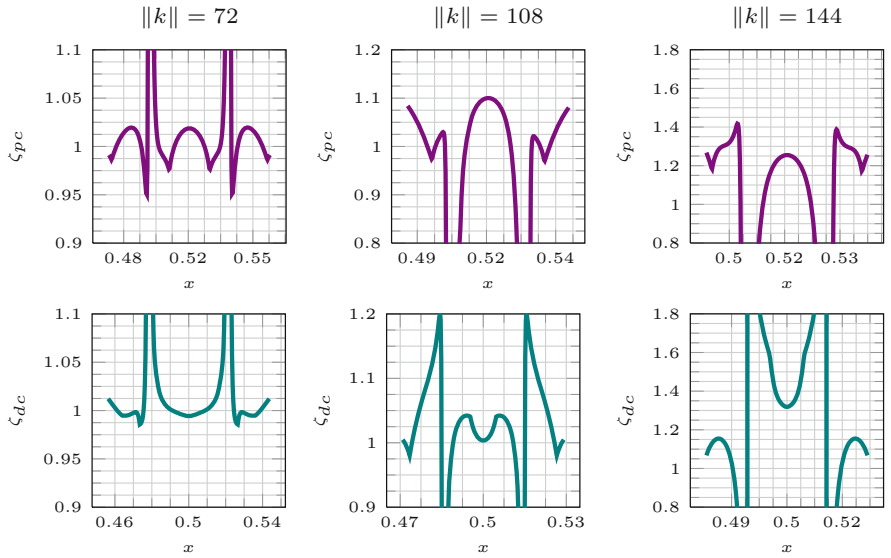


Fig. 8 Zoomed plots of extraction of normalized group velocity ζ given in (32) using the consistent mass for a PUM space based on a uniform cover with 32×32 patches and cubic local approximation spaces. Depicted are the normalized group velocity $\zeta_{pc}(x)$ (*top*) and the normalized phase velocity $\zeta_{dc}(x)$ (*bottom*)

we need to compress these functions into single values. To this end, we define (in analogy to the standard extraction approach)

$$\psi_{pc} := \psi_{pc}(o_i), \quad \psi_{dc} := \psi_{dc}(o), \quad \zeta_{pc} := \zeta_{pc}(o_i), \quad \zeta_{dc} := \zeta_{dc}(o) \tag{33}$$

and use these representative values only in the following.

Example 3. Let us now summarize the dispersion properties of our PUM. To this end, we again consider our uniform 32×32 cover with $\alpha = 1.3$ and choose at first linear polynomials as local approximation spaces. For these parameters we determine the (compressed) phase velocities ψ_{pc}, ψ_{dc} and group velocities ζ_{pc}, ζ_{dc} (33) when we use the consistent mass matrix (12) in (23) and compare these with the results when using the lumped mass matrix (14) in (23).

The only remarkable difference when comparing these plots is that for the consistent mass matrix the values ψ_{pc} and ζ_{pc} are larger than ψ_{dc} and ζ_{dc} whereas for the lumped mass matrix they are smaller. Moreover, the variation between the results ψ_{pc}, ζ_{pc} and ψ_{dc}, ζ_{dc} for different shifts are smaller for the lumped mass matrix than for the consistent mass matrix. This behavior is essentially due to the fact that the use of the lumped mass matrix (14) yields a more local approximation by construction so that the potential benefit of using multiple patches to define the approximations $\psi_{dc} = \psi_{dc}(o)$ and $\zeta_{dc} = \zeta_{dc}(o)$ is limited. Thus, these values come closer to the values $\psi_{pc} = \psi_{pc}(o_i)$ and $\zeta_{pc} = \zeta_{pc}(o_i)$ which employ only the approximation space from a single patch ω_i. Note, however, that the results ψ_{pc} and ζ_{pc} obtained with the consistent mass matrix are larger than ψ_{pc}, ψ_{dc} and ζ_{pc}, ζ_{dc} attained for the lumped mass matrix. Thus, the worst case dispersion error for the lumped mass matrix is in fact smaller than for the consistent mass matrix (Fig. 9).

Overall we find an accuracy of $\leq 5\,\%$ error, which is usually deemed to be sufficient for most engineering applications, in the phase velocity for $\|k\| \lesssim 32$, i.e.

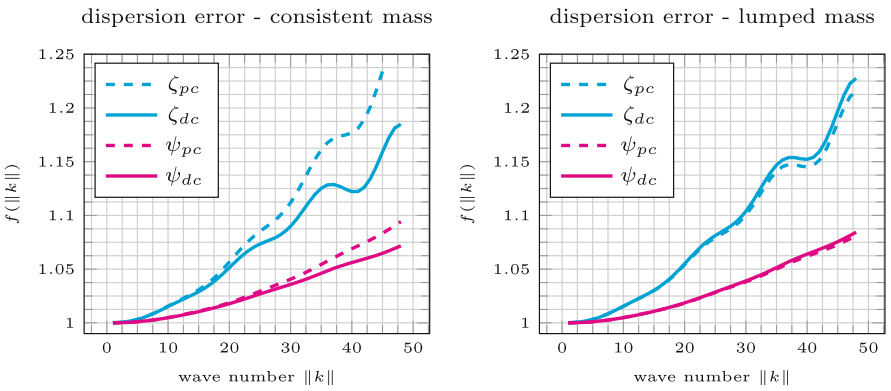


Fig. 9 Comparison of the dispersion properties for the consistent mass (*left*) and the lumped mass (*right*). Depicted are plots of the (*compressed*) normalized phase ψ and group velocities ζ given in (33) against the employed wave number $\|k\|$

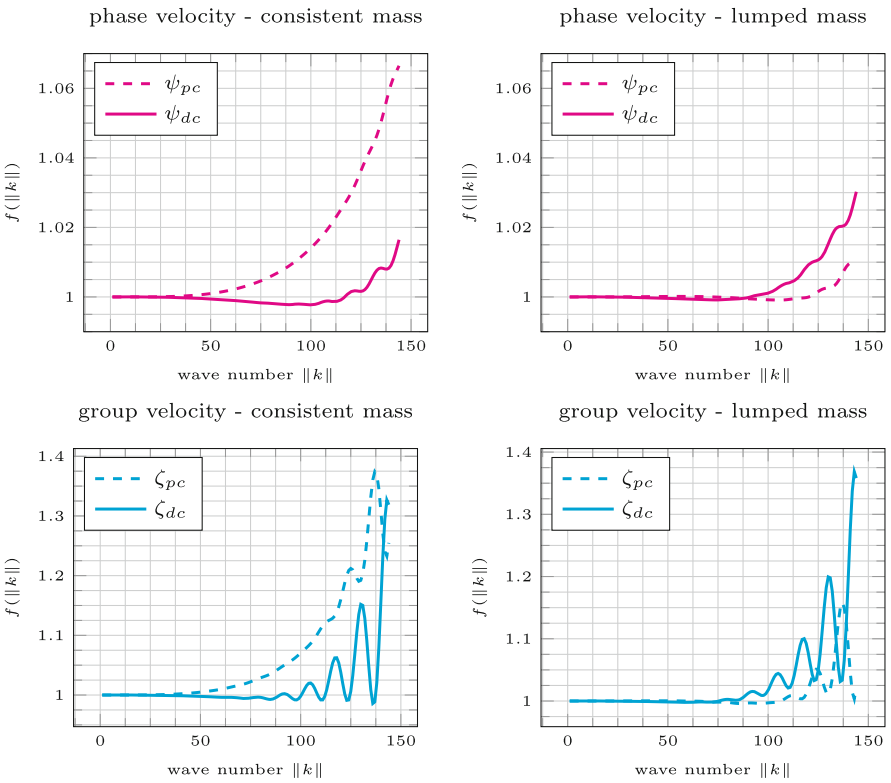


Fig. 10 Comparison of the dispersion properties for the consistent mass (*left*) and the lumped mass (*right*). Depicted are plots of the (*compressed*) normalized phase ψ (*top*) and group velocities ζ (*bottom*) given in (33) against the employed wave number $\|k\|$

when the wavelength is covered by at least 6 patches for linear local approximation spaces. With respect to the group velocity a 5 % error is already attained when less than 10 patches cover a wavelength.

The results for cubic polynomials are displayed in the plots of Fig. 10. Again, we see that the difference between the extracted values ψ_{pc} and ψ_{dc} as well as for ζ_{pc} and ζ_{dc} are smaller when using the lumped mass matrix and the absolute values are also smaller than the worst case results ψ_{pc} and ζ_{pc} for the consistent mass matrix. Hence, the lumped mass matrix approach also yields better dispersion properties than the consistent mass matrix for higher order approximations. The common 5 % engineering error bound in the phase velocity for cubic local approximation spaces is satisfied for all considered wave numbers $\|k\| \leq 144$ (i.e. when the patch size $\operatorname{diam}(\omega_i)$ is at least the wavelength) with the lumped mass matrix, whereas for the consistent mass matrix $\psi_{pc} \leq 1.05$ for $\|k\| \lesssim 125$ only. The group velocity shows an error less than 5 % for $\|k\| \lesssim 110$ when using the lumped mass matrix and $\|k\| \lesssim 80$ with the consistent mass matrix.

In summary we find that the dispersion properties of our PUM with the lumped mass matrix are very much comparable to those attained with the consistent mass matrix—if not slightly better. We find no significant reduction in the quality of the obtained phase or group velocity errors. Thus, an explicit time-stepping scheme with the lumped mass matrix yields a similar dispersion error yet allows for a larger critical time step size than the consistent mass matrix [18]. Moreover, the computational effort associated with single time step with the lumped mass matrix will be orders of magnitudes smaller than for the consistent mass matrix. In the final two examples we will therefore consider explicit time stepping schemes with our PUM in two and three dimensions for the equations of motion in a linearly elastic medium (10) and focus on the critical time step size, the computational cost per time step and compare the result obtained with the consistent and lumped mass matrix directly.

An explicit time-stepping scheme with central differences yields the following discrete two step scheme

$$N\tilde{a}_n = K\tilde{u}_n, \quad \tilde{v}_{n+1/2} = \delta t_{n+1/2}\tilde{a}_n + \tilde{v}_{n-1/2} \tag{34}$$

and

$$\tilde{u}_{n+1} = \delta t_n \tilde{v}_{n+1/2} + \tilde{u}_n \tag{35}$$

where N denotes the employed mass matrix, i.e. M or $\bar{M}$, K is the stiffness matrix, $\tilde{u}_{n+1}$ refers to the coefficient-vector of the discrete displacement at time t_{n+1}, $\tilde{v}_{n+1/2}$ is the velocity coefficient-vector at time $t_{n+1/2} = \frac{1}{2}(t_{n+1} + t_n)$ and $\tilde{a}_n$ encodes the acceleration at time t_n. For the treatment of essential boundary conditions we employ an algebraic splitting of the PUM space V^{PU} due to [16].

Example 4. In our first time-stepping example we consider

$$\begin{aligned} \rho\ddot{u} &= \mathbf{div}\,\boldsymbol{\sigma}(u) \text{ in } \Omega \times (0,T), \quad u(\cdot,0) = 0 \text{ in } \Omega, \\ \dot{u}(\cdot,0) &= \gamma \qquad\qquad \text{in } \Omega, \qquad\qquad u(x,t) = 0 \text{ for } x \in \partial\Omega, t > 0, \end{aligned} \tag{36}$$

with vanishing Dirichlet boundary conditions for the displacement on $\partial\Omega$ with $\Omega = (0,1)^2$. For the realization of essential boundary conditions we use the algebraic approach of [16] which provides (a basis transformation and) a splitting into interior and boundary degrees of freedom, i.e. there are maps Π° and Π^∂ such that $\tilde{u} = \Pi^\circ\tilde{u} + \Pi^\partial\tilde{u} = \tilde{u}^\circ + \tilde{u}^\partial$. Thus, we attain the scheme

$$N^\circ\tilde{a}_n^\circ = \Pi^\circ K\tilde{u}_n, \quad \tilde{v}_{n+1/2}^\circ = \delta t_{n+1/2}\tilde{a}_n^\circ + \tilde{v}_{n-1/2}^\circ \tag{37}$$

and

$$\tilde{u}_{n+1}^\circ = \delta t_n \tilde{v}_{n+1/2}^\circ + \tilde{u}_n^\circ, \quad \tilde{u}^\partial = 0 \tag{38}$$

with a critical time-step size

$$\delta t_{\text{critical}} \leq \frac{2}{\sqrt{\lambda_{\max}}}, \tag{39}$$

where $\lambda_{\max}$ denotes the maximal eigenvalue of $(N^{\circ})^{-1}(\Pi^{\circ} K)$. Here, we estimate $\lambda_{\max}$ via a simple power iteration to control the time-step size accordingly when we use the consistent mass matrix $N = M$ and the lumped mass matrix $N = \bar{M}$.

The employed initial conditions $u(\cdot, 0) = 0$ and $\dot{u}(\cdot, 0) = \gamma$ with

$$\gamma(x) = \begin{pmatrix} -\exp\left(\frac{-\|x-o\|^2}{\tau^2}\right) \\ 0 \end{pmatrix} \quad \text{with } \tau = 0.05, \tag{40}$$

model an elastic body at rest with an initial momentum (near the center $o = (0.5, 0.5)^T$ of the domain) in the x-direction.

We again employ a 32×32 uniform cover and cubic polynomials for the spatial discretization. Figure 11 gives snapshots of the evolution of the displacement field computed with the lumped mass matrix. From these plots we can clearly observe the faster pressure wave travelling in x-direction and the slower stress wave travelling in y-direction.

Let us know focus on the comparison of the results and the overall performance when using the consistent mass $N = M$ or the lumped mass $N = \bar{M}$. To this end, we estimate the critical time-step in both cases and obtain $\delta^c_{\text{critical}} = 0.002165$ and $\delta^l_{\text{critical}} = 0.005325$ respectively. Thus, the lumped mass matrix allows for a time-step more than twice the size of the time-step when using the consistent mass. Moreover, the use of the consistent mass matrix requires the application of a more involved and expensive linear solver.[2] Here, we used a conjugate gradient solver preconditioned by a symmetric block-Gauss–Seidel smoother for the consistent mass matrix whereas for the lumped mass matrix a single application of a simple block-Jacobi smoother already yields a direct solver. Thus, the compute time associated with a single time-step with the lumped mass matrix is in general multiple orders of magnitude smaller than that of a time-step with the consistent mass matrix.

In Fig. 12 we present snapshots of the computed solutions at $t = 4$, i.e. after roughly 2,000 time-steps with $\delta t = 0.9\delta t^c_{\text{critical}}$ and 800 time-steps with $\delta t = 0.9\delta t^l_{\text{critical}}$ respectively. From these plots we find that the dominant features are preserved in all approaches, however, the lumped mass matrix results (computed with the same time-step size) deviate in some details from those obtained with the consistent mass matrix. Yet, using a much larger time-step for the lumped mass matrix has no substantial impact on the quality of the solution. A more detailed

[2] In this example the use of a classical direct solver may be an option since the consistent mass matrix is independent of time. However, in general the discretization space, especially the enrichment functions in a PUM, may change over time so that a linear solver with an expensive setup phase is in general prohibitively expensive.

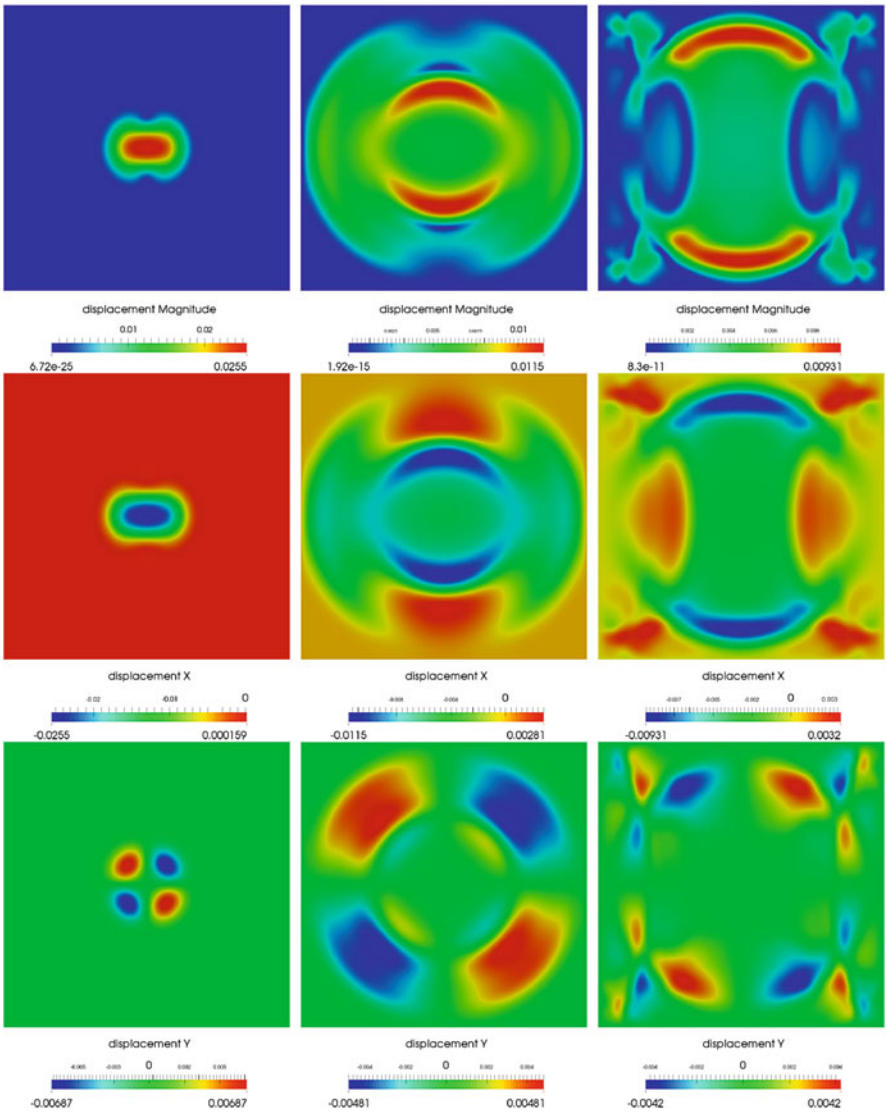


Fig. 11 Snapshots of the evolution of the displacement field satisfying (36) in two dimensions at $t = 0.1, 0.4, 0.68$ (*top* to *bottom*)

study with respect to convergence behavior in time, however, is of course necessary to quantify these errors and differences.

Example 5. Finally, we present some simulation results for (36) in three dimensions on a uniform cover of $\Omega = (0,1)^3$ with $32 \times 32 \times 32$ patches and quadratic

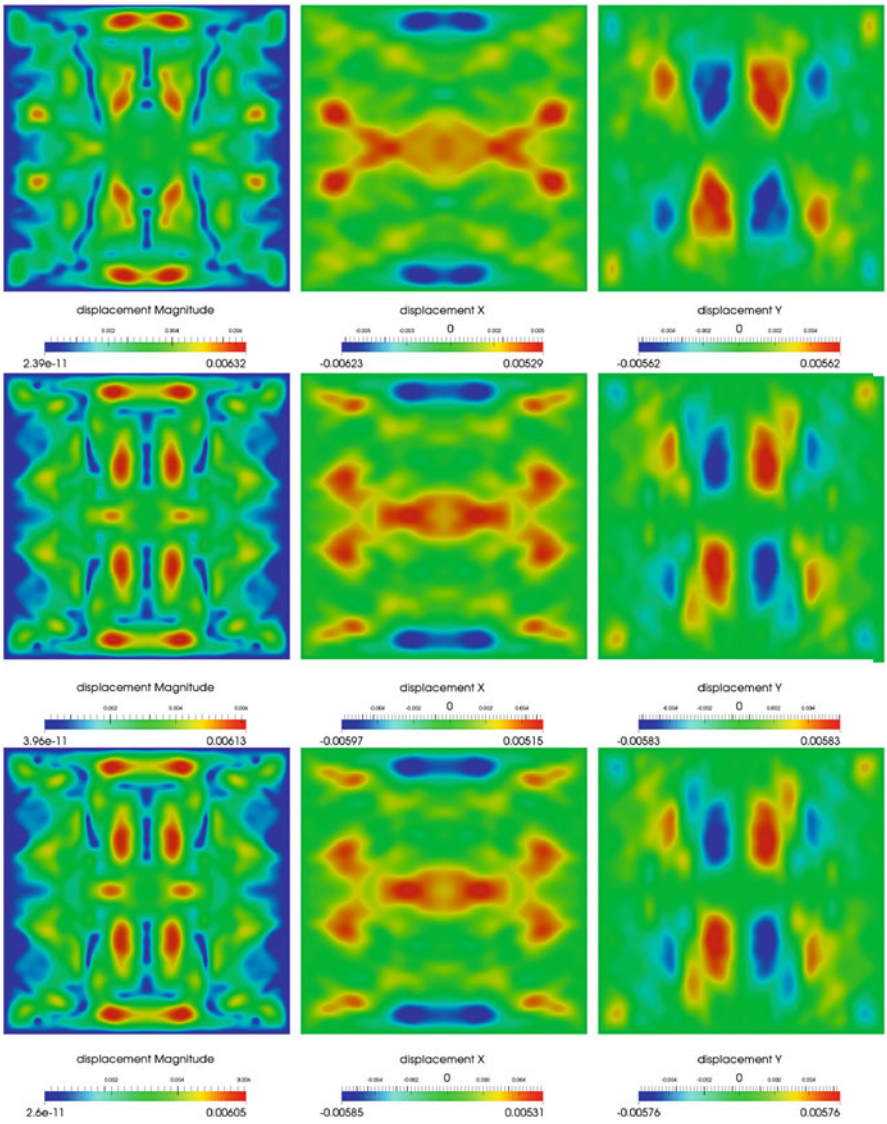


Fig. 12 Comparison of the computed displacement fields at $t = 4$ obtained with the consistent mass matrix (*top*) and the lumped mass matrix (*center*) with $\delta = 0.9\delta t^c_{\text{critical}}$ and the lumped mass matrix with $\delta = 0.9\delta t^l_{\text{critical}}$ (*bottom*)

polynomials as local approximation spaces. Here, we enforce vanishing Dirichlet conditions on the face $x = 0$ only (on the rest of the boundary we prescribe vanishing Neumann conditions) and employ the initial conditions $u(\cdot, 0) = 0$ and $\dot{u}(\cdot, 0) = \gamma$ with

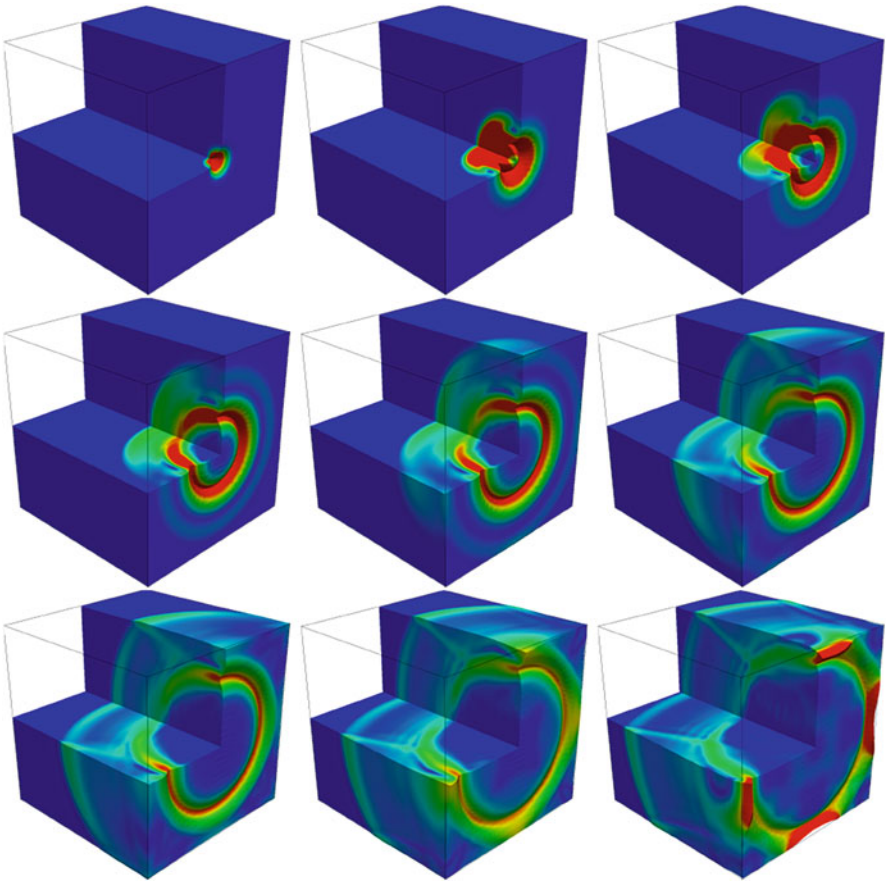


Fig. 13 Snapshots of the evolution of the displacement field satisfying (36) in three dimensions at $t = 0.08m$ with $m = 1, 3, 4, \ldots, 10$ (*top left* to *bottom right*). The magnitude of the displacement field is color-coded on $^3/_4$ of the deformed configuration (exaggerated by a factor of 10)

$$\gamma(x) = \begin{pmatrix} -\exp\left(\frac{-\|x-\breve{o}\|^2}{\tau^2}\right) \\ 0 \\ 0 \end{pmatrix} \quad \text{with } \tau = 0.05, \tag{41}$$

and $\breve{o} = (1, 0.5, 0.5)^T$ denoting the center of the face $x = 1$. Here, we employ only the lumped mass matrix approach since the adverse effect on the computational time when using the consistent mass matrix is substantially increased in three dimensions. The estimated critical time-step size in this configuration was $\delta t^l_{\text{critical}} = 0.0075$ and we employed a time-step $\delta t = 0.0066$. Figure 13 displays snapshots of the evolution of the displacement field. Again, we can observe the resulting pressure wave travelling faster through the domain in the x-direction than the stress waves travelling in the (y, z)-plane.

5 Concluding Remarks

We analyzed the dispersion properties of the PUM with polynomial enrichment functions using the consistent mass matrix as well as a lumped mass matrix according to [18]. Our results show that the dispersion errors attained with the lumped mass matrix are at least comparable to those associated with the consistent mass matrix. Thus, the use of the lumped mass matrix in a transient analysis with our PUM is well justified and yields a similar accuracy as the consistent mass matrix.

References

1. I. Babuška, G. Caloz, J.E. Osborn, Special finite element methods for a class of second order elliptic problems with rough coefficients. SIAM J. Numer. Anal. **31**, 945–981 (1994)
2. I. Babuška, F. Ihlenburg, E. Paik, S. Sauter, A generalized finite element method for solving the Helmholtz equation in two dimensions with minimal pollution. Comput. Methods Appl. Mech. Eng. **128**, 325–359 (1995)
3. I. Babuška, J.M. Melenk, The partition of unity finite element method: basic theory and applications. Comput. Methods Appl. Mech. Eng. **139**, 289–314 (1996). Special issue on Meshless Methods
4. I. Babuška, J.M. Melenk, The partition of unity method. Int. J. Numer. Methods Eng. **40**, 727–758 (1997)
5. I. Babuška, S. Sauter, Is the pollution effect of the FEM available for the Helmholtz equation considering high wave numbers. SIAM J. Numer. Anal. **34**(6), 2392–2423 (1997)
6. S.W. Chi, J.S. Chen, H. Luo, H.Y. Hu, L. Wang, Dispersion and stability properties of radial basis collocation method for elastodynamics. Numer. Methods Partial Differ. Equ. **29**(3), 818–842 (2013)
7. M.A. Christon, T.E. Voth, Results of von Neumann analyses for reproducing kernel semi-discretizations. Int. J. Numer. Methods Eng. **47**(7), 1285–1301 (2000)
8. T.-P. Fries, T. Belytschko, The extended/generalized finite element method: an overview of the method and its applications. Int. J. Numer. Methods Eng. **84**(3), 253–304 (2010)
9. M. Griebel, M.A. Schweitzer, A particle-partition of unity method—part II: efficient cover construction and reliable integration. SIAM J. Sci. Comput. **23**(5), 1655–1682 (2002)
10. M. Griebel, M.A. Schweitzer, A particle-partition of unity method—part VII: adaptivity, in *Meshfree Methods for Partial Differential Equations III*, ed. by M. Griebel, M.A. Schweitzer. Lecture Notes in Computational Science and Engineering, vol. 57 (Springer, Berlin, 2006), pp. 121–148
11. S. Ham, K.-J. Bathe, A finite element method enriched for wave propagation problems. Comput. Struct. **94–95**, 1–12 (2012)
12. I. Harari, T.J.R. Hughes, Finite element method for the Helmholtz equation in an exterior domain: model problems. Comput. Methods Appl. Mech. Eng. **87**, 59–96 (1991)
13. F. Ihlenburg, *Finite Element Analysis of Acoustic Scattering*. Applied Mathematical Sciences, vol. 132 (Springer, New York, 1998)
14. M.A. Schweitzer, *A Parallel Multilevel Partition of Unity Method for Elliptic Partial Differential Equations*. Lecture Notes in Computational Science and Engineering, vol. 29 (Springer, Berlin, 2003)
15. M.A. Schweitzer, An Adaptive hp-version of the multilevel particle–partition of unity method. Comput. Methods Appl. Mech. Eng. **198**, 1260–1272 (2009)
16. M.A. Schweitzer, An algebraic treatment of essential boundary conditions in the particle–partition of unity method. SIAM J. Sci. Comput. **31**(2), 1581–1602 (2009)

17. M.A. Schweitzer, Generalizations of the finite element method. Cent. Eur. J. Math. **10**(1), 3–24 (2012)
18. M. Schweitzer, Variational mass lumping in the partition of unity method. SIAM J. Sci. Comput. **35**(2), A1073–A1097 (2013)
19. T. Strouboulis, I. Babuska, R. Hidajat, The generalized finite element method for Helmholtz equation: theory, computation, and open problems. Comput. Methods Appl. Mech. Eng. **195**(37–40), 4711–4731 (2006). John H. Argyris Memorial Issue. Part I
20. T. Strouboulis, R. Hidajat, Partition of unity method for Helmholtz equation: q-convergence for plane-wave and wave-band local bases. Appl. Math. **51**(2), 181–204 (2006)
21. T. Strouboulis, R. Hidajat, I. Babuska, The generalized finite element method for Helmholtz equation. Part II: effect of choice of handbook functions, error due to absorbing boundary conditions and its assessment. Comput. Methods Appl. Mech. Eng. **197**(5), 364–380 (2008). Enriched Simulation Methods and Related Topics
22. Y. You, J.-S. Chen, T.E. Voth, Characteristics of semi- and full discretization of stabilized Galerkin meshfree method. Finite Elem. Anal. Des. **38**(10), 999–1012 (2002)

An Immersed Meshfree Galerkin Approach for Particle-Reinforced Composite Analysis

Cheng-Tang Wu

Abstract This paper introduces an immersed meshfree approach within a Galerkin framework for the elastic analysis of particle-reinforced composites. A new meshfree discretization is developed and applied to the composite solids with overlapping sub-domains. Since each sub-domain is discretized independently, the generation of a conforming mesh in the finite element method can be evaded. A meshfree convex approximation is employed to approximate the overlapping sub-domains and to enforce the global Dirichlet boundary conditions. In addition, a point-wise continuity is imposed on displacements of interfacial nodes across the interface. This leads to a nonconforming meshfree Galerkin formulation which can be shown to satisfy an optimal error estimate in the energy and L^2 norms. Two numerical benchmarks are analyzed to study the accuracy and the applicability of the method.

Keywords Interface problem • Meshfree method • Convex approximation • Nonconforming • Error estimate

1 Introduction

Numerical modeling of composite solids in elasticity problems has been extensively studied in last three decades. Mathematically, this type of boundary value problems can be modeled by the elliptic equations containing discontinuous coefficients [2]. The discontinuous coefficients introduce the jump conditions across the interface in displacements as well as in the flux. These jump conditions generally are determined by the relevant physics. The elasticity interface problems presented in this paper considers the homogeneous jump conditions which arise in a wide range of mathematical modeling in material science and bio-medical applications such as the elastic analysis of rubber compounds [40, 47], modeling of bone structures [21] and brain shifting [50]. The conventional finite element method

C.-T. Wu (✉)
Livermore Software Technology Corporation (LSTC), 7374 Las Positas Road, Livermore, CA 94551, USA
e-mail: ctwu@lstc.com

M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations VII*, Lecture Notes in Computational Science and Engineering 100,
DOI 10.1007/978-3-319-06898-5_15

in this context is to generate a matching (conforming) mesh across the material interface and use standard finite element shape functions to approximate the solution of the boundary value problems. However, generating matching meshes in multi-dimensional composite solids suitable for the finite element analysis is difficult in particular having interfaces in irregular geometries.

The mortar finite element method [10] provides a flexible way to couple mismatching meshes across the interface. The mortar finite element method is a domain decomposition technique that enforces the jump conditions across the interface by Lagrange multipliers and results in a saddle point problem which requires appropriate solvers. Nevertheless, an arbitrary choice of the approximation space for Lagrange multipliers may violate the inf-sup condition [3, 10, 12] and can lead to instabilities that eventually cause artificial oscillations in the traction fields. Alternative variational approach for the discretization of interface problem is offered by Nitsche's method [24]. The Nitsche's method has a close relationship [39] with Barbosa and Hughes' least-squares stabilized Lagrange multiplier formulation [7] in circumventing the inf-sup condition. Despite the fact that it is non-trivial in the implementation and there is a need to determine the penalty parameters, Nitsche's method does not suffer from ill-conditioning and has been shown to preserve optimal convergence in L^2 and energy norms for elasticity interface problems [23].

On the other hand, the partition of unity method (PUM) [5, 6] employs a priori knowledge about the behavior of the solution at the interface to obtain special finite element spaces. Later in [41], the method was referred to as generalized finite element method (GFEM), since the classical FEM is a special case of this method [4]. The ability of choosing a wide variety of enrichment functions in GFEM allows it to approximate non-smooth solutions of BVPs on domains having internal boundaries, corners or multiple cracks [20,40]. The enrichment function is extrinsic to the finite element basis function and is not limited to polynomials. Nevertheless, this advantage is achieved at a high computational cost due to expensive numerical integration [4]. The extended finite element method (XFEM) [8, 19, 33] is an application of PUM for problems of interface tracking and crack growth. Since the solution spaces of XFEM using discontinuous enrichment function generally do not satisfy the Dirichlet jump condition on the interface, linear or tied constraint methods such as Lagrangian multipliers or penalty methods [18, 37] are often utilized to enforce the Dirichlet jump condition as the one in the mortar finite element method. Inspired by the variational multiscale method [25], the bubble-stabilized method [34, 37] can be viewed as another improvement of Lagrangian multiplier method in solving the interface problem. The corresponding pair of spaces in displacement and Lagrangian multiplier fields can be proven to meet the inf-sup stable condition [34].

Similar implicit boundary representation techniques have been developed based on a uniform Cartesian mesh under different versions of immersed or fictitious domain approaches. Among them are finite cell method for geometrical modeling of immersed problems [35], immersed finite element method for fluid-structure interaction problems [36] and immersed finite element method for elasticity interface problems [27, 28, 30]. The recently developed non-conforming immersed

finite element method (IFEM) [28] is also rooted in the GFEM. A sophisticated conforming local basis functions [29] has also been introduced to non-conforming IFEM and shown to be optimally convergent in L^2 and energy norms. However, the extension of IFEM to the three-dimensional case is not clear. A different way of enlarging the local ansatz space in modeling interface elasticity problem was presented in the composite finite element method (CFE) [38]. The elements in CFE are constructed hierarchically in a way that the shape functions of the coarse grid elements are combinations of the ones of fine grid elements. The coarse grid shape functions are used to approximate the interface problem whereas the linear system can be solved efficiently on the fine scale by the multi-grid algorithm.

The idea of solving elasticity interface problems using meshfree method is not new. Cordes and Moran [17] impose the jump condition by adding an interface constraint in the variational formulation and solve the equation by Lagrange multiplier method. Since the material interface serves as a visibility criterion for the construction of meshfree approximation in bi-material model, a set of interface nodes has to be manually added along the interface together with properly adjusted integration cells for the domain integration. A parallel research [26,32] is devoted to the development of particular meshfree approximation that contains discontinuities in the derivatives across the material interface. This approach bears a need to define the interface nodes and may not be easily performed when the interface involves complex three-dimensional geometries object.

The aim of this study is to present an alternative approach for elasticity interface problem using a meshfree method. A significant feature of the present approach is its flexibility to adopt the overlapping meshes in immersed structures that can be easily discretized using the finite element model. As a consequence, there is no need to insert the interface nodes and their corresponding integration cells in the proposed method. The reminder of the paper is outlined as follows: In the next section we define the elliptic boundary-value problem containing discontinuous coefficients and formulates the weak form of the equilibrium equation. In Sect. 3, we modify the variational formulation for the meshfree method to be used in the elasticity interface problem. The details of numerical discretization, approximation and domain integration for the modified variational method are described. An a priori estimate is provided for the error measures in the energy and L^2 norms. Two numerical examples are presented in Sect. 4 to illustrate the accuracy and applicability of the method. Final remarks are drawn in Sect. 5.

2 Preliminary

We consider an elastic solid occupying a bounded and open domain $\Omega \subset R^2$ with Lipschitz boundary. Let $\partial\Omega_D$ and $\partial\Omega_N$ be two open subsets of boundary $\partial\Omega$ such that $\partial\Omega = \partial\Omega_D \bigcup \partial\Omega_N$ and $\partial\Omega_D \bigcap \partial\Omega_N = 0$. $\boldsymbol{g(x)}$ is the prescribed displacement applied on the Dirichlet boundary $\partial\Omega_D$, and $t(x) \in L^2(\partial\Omega_N)$ is the prescribed traction applied on the Neumann boundary $\partial\Omega_N$ with $\boldsymbol{n}_0$ denoting the outward unit

normal to the boundary $\partial\Omega_N$. The elastic body is composed of two perfectly bonded materials with a zero-thickness interface Γ. The equilibrium configuration of the elastic body is characterized by the continuity of displacement and continuity of normal stress across the material interface Γ. The elasticity interface problem can be described by the following second-order elliptic boundary value problem with the associated homogeneous Dirichlet and Neumann jump conditions on the interface Γ

$$\begin{gathered} -\nabla \cdot (C\ (x)\ \cdot \nabla_s u\,(x)) = f\ (x) \quad x \in\ \Omega \backslash \Gamma \\ u = g\ (x) \quad x \in\ \partial\Omega_D \\ C\ (x)\ \cdot \nabla_s u\,(x) \cdot n_0 = t\ (x)\ \ x \in\ \partial\Omega_N \\ [[\ \mathrm{u}\ ]] = 0\ \ x \in \Gamma \\ [[\ \mathrm{C}\,(x) \cdot \nabla_s u\,(x) \cdot n\ ]] = 0\ \ x \in \Gamma \end{gathered} \tag{1}$$

where the function $u\colon \Omega \rightarrow R^2$ is the displacement and $f\colon \Omega \rightarrow R^2$ is the prescribed body force over the domain Ω. The notation ∇su denotes the symmetric gradient of the displacement. Without loss of generality, we assume the interface Γ is a smoothed and closed curve that divides the global domain Ω into two regions: Ω^+ representing the matrix and Ω^- denoting the immersed media or inclusion such that their union gives the global domain Ω, $\bar{\Omega} = \overline{\Omega^+} \bigcup \overline{\Omega^-}$ and $\Gamma = \partial\Omega^+ \bigcap \partial\Omega^-$ as depicted in Fig. 1. The symbol $\boldsymbol{n}$ in Fig. 1 denotes the outward unit normal vector on Γ. We also define the jump operator $[[\ \cdot\ ]]$ by

$$[[\ q\ ]]\,(x) = q^+\,(x) - q^-\,(x) \tag{2}$$

in which $+$ and $-$ denote the two sides of the interface Γ with the jump of quantity q across the interface. The body force and material constants can exhibit discontinuities across the interface Γ, but have smooth restrictions f^+, C^+ to Ω^+ and f^-, C^- to Ω^-. They are given by

$$f = \begin{cases} f^+ \ in\ \ \Omega^+ \\ f^- \ in\ \ \Omega^- \end{cases}, \text{ and } C = \begin{cases} C^+ \ in\ \ \Omega^+ \\ C^- \ in\ \ \Omega^- \end{cases}. \tag{3}$$

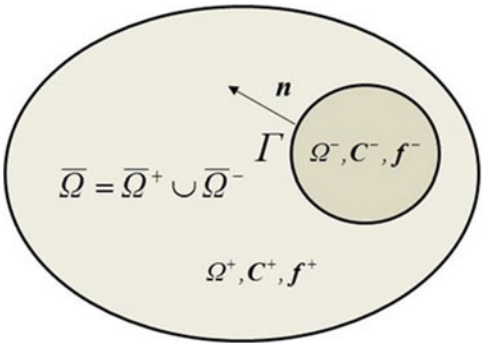


Fig. 1 Graphical depiction for interface elasticity model

The infinitesimal strain tensor $\varepsilon(u)$ is defined by

$$\varepsilon(u) = \nabla_s u = \frac{1}{2}\left(\nabla u + (\nabla u)^T\right) \tag{4}$$

C^+ and $C^- \in L^\infty(\Omega)$ are elasticity tensors with major and minor symmetries and are corresponding to different materials in Ω^+ and Ω^- respectively. In the case of linear isotropic elasticity, we take C^+ and C^- to be constants. The Cauchy stress tensor σ and strain tensor ε have the following relationship

$$\begin{cases} \sigma^+ = C^+ \cdot \varepsilon = 2\mu^+\varepsilon + \lambda^+ tr(\varepsilon) I & \text{in } \Omega^+ \\ \sigma^- = \mathrm{C}^- \cdot \varepsilon = 2\mu^-\varepsilon + \lambda^- \mathrm{tr}(\varepsilon)\mathrm{I} & \text{in } \Omega^- \end{cases} \tag{5}$$

where the positive constants μ^+, μ^- and λ^+, λ^- are the Lamé constants. The Lamé constants are related to the Young's modulus E and Poisson ratio v by $\mu = E/(2(1+v))$, $\lambda = vE((1+v)(1-2v))$.

The variational form of this problem is to find the displacement $u \in V^g = \left\{v \in H^1(\Omega) : v = g \text{ on } \partial\Omega_D\right\}$ such that for all $v \in V$

$$a(u, v) = l(v) \tag{6}$$

where the functional space $V = H_0^1(\Omega)$ consists of functions in Sobolev space $H^1(\Omega)$ which vanish on the boundary $\partial\Omega$ and is defined by

$$V(\Omega) = \left\{v : v \in H^1, v = 0 \text{ on } \partial\Omega_D\right\}. \tag{7}$$

The bilinear form $a(\cdot,\cdot)$ and linear functional $l(\cdot)$ are obtained by multiplying the test function $v \in V$ to both sides of Eq. (6) and integrating over the regions Ω^+ and Ω^- separately using Green's formula.

$$\begin{aligned} &\int_{\Omega^+} f^+(x) \cdot v d\Omega + \int_{\Omega^-} f^-(x) \cdot v d\Omega = \\ &-\int_{\Omega^+} \nabla \cdot \left(C^+(x) \cdot \nabla_s u(x)\right) \cdot v d\Omega - \int_{\Omega^-} \nabla \cdot (C^-(x) \cdot \nabla_s u(x)) \cdot v d\Omega = \\ &\int_{\Omega^+} \varepsilon(u) \cdot C^+ \cdot \varepsilon(v) d\Omega - \int_\Gamma C^+(x) \cdot \nabla_s u(x) \cdot n^+ \cdot v d\Gamma \\ &+ \int_{\Omega^-} \varepsilon(u) \cdot C^- \cdot \varepsilon(v) d\Omega - \int_\Gamma C^-(x) \cdot \nabla_s u(x) \cdot n^- \cdot v d\Gamma \\ &- \int_{\partial\Omega_N} (t \cdot v) d\partial\Omega. \end{aligned} \tag{8}$$

Using the fact that $n^+ = -n^-$, we can rewrite the above equation to become

$$\int_{\Omega^+} f^+(x)\cdot v d\Omega + \int_{\Omega^-} f^-(x)\cdot v d\Omega = \int_{\Omega^+} \varepsilon(u)\cdot C^+\cdot \varepsilon(v)\, d\Omega + \int_{\Omega^-} \varepsilon(u)\cdot C^-\cdot \varepsilon(v)\, d\Omega - \int_{\partial\Omega_N} (\mathrm{t}\cdot \mathrm{v})\, d\partial\Omega - \int_{\Gamma} [[\mathrm{C}(\mathrm{x})\cdot \nabla_s \mathrm{u}(\mathrm{x})\cdot \mathrm{n}]]\cdot \mathrm{v}\, d\Gamma. \tag{9}$$

Applying the homogeneous Neumann jump condition to Eq. (9) yields

$$\int_{\Omega^+} \varepsilon(u)\cdot C^+\cdot \varepsilon(v)\, d\Omega + \int_{\Omega^-} \varepsilon(u)\cdot C^-\cdot \varepsilon(v)\, d\Omega - \int_{\Omega^+} f^+\cdot v d\Omega - \int_{\Omega^-} f^-\cdot v d\Omega - \int_{\partial\Omega_N} t\cdot v d\partial\Omega = a(u,v) - l(v) = 0 \tag{10}$$

where

$$a(u,v) = \int_{\Omega^+} \varepsilon(u)\cdot C^+\cdot \varepsilon(v)\, d\Omega + \int_{\Omega^-} \varepsilon(u)\cdot C^-\cdot \varepsilon(v)\, d\Omega \tag{11}$$

$$l(v) = \int_{\Omega^+} f^+\cdot v d\Omega + \int_{\Omega^-} f^-\cdot v d\Omega + \int_{\partial\Omega_N} t\cdot v d\partial\Omega \tag{12}$$

It is noted that the elasticity tensors C^+ and C^- are symmetric, and homogeneous Neumann jump condition is enforced in the variational level. Obviously, the bilinear form $a(\cdot,\cdot)$ in Eq. (11) is symmetric, bounded and coercive by Friedrich's inequality. The existence and uniqueness of the problem is ensured by the Lax-Milgram theorem [2].

3 Immersed Meshfree Formulation

3.1 Immersed Meshfree Discretization and Integration

For simplicity, we only consider the case of a pure displacement problem under a homogeneous boundary condition ($\partial\Omega_D = 0$). The standard meshfree Galerkin method is formulated on a finite dimensional space $V_h \subset V$ employing the variational formulation of Eq. (10) to find $u_h \in V_h$ such that

$$a(u_h, v_h) = l(v_h) \quad \forall v_h \in V_h \tag{13}$$

where $V_h = \mathrm{span}\{\Psi_I : I \in Z_I\}$ and Z_I is an index set. $\{\Psi_I(x)\}_{I\in Z_I}$ are shape functions constructed using meshfree convex approximation [49]. Using the superposition principle for the above linear system, the discrete bilinear form in

Eq. (13) can be re-expressed by

$$\begin{aligned} &a\left(u_h, v_h\right) = \\ &\int_{\Omega^+} \varepsilon\left(u_h\right) \cdot C^+ \cdot \varepsilon\left(v_h\right) d\Omega + \int_{\Omega^-} \varepsilon\left(u_h\right) \cdot C^- \cdot \varepsilon\left(v_h\right) d\Omega = \\ &\int_{\Omega^+ \bigcup \Omega^-} \varepsilon\left(u_h\right) \cdot C^+ \cdot \varepsilon\left(v_h\right) d\Omega + \int_{\Omega^-} \varepsilon\left(u_h\right) \cdot \left(C^- - C^+\right) \cdot \varepsilon\left(v_h\right) d\Omega \end{aligned} \tag{14}$$

Similarly, we have the discrete linear functional to be rewritten as

$$\begin{aligned} l\left(v_h\right) &= \int_{\Omega^+} f^+ \cdot v_h d\Omega + \int_{\Omega^-} f^- \cdot v_h d\Omega \\ &= \int_{\Omega^+ \bigcup \Omega^-} f^+ \cdot v_h d\Omega + \int_{\Omega^-} {}^- \cdot v_h d\Omega - \int_{\Omega^-} f^+ \cdot v_h d\Omega \\ &= \int_{\Omega^+ \bigcup \Omega^-} f^+ \cdot v_h d\Omega + \int_{\Omega^-} \left(f^- - f^+\right) \cdot v_h d\Omega \end{aligned} \tag{15}$$

The expression of discrete forms in Eqs. (14) and (15) allows us to reset the meshfree computation domain by two overlapping sub-regions; namely $\Omega^+ \bigcup \Omega^-$ and Ω^-. In another words, the computation domain Ω^- can be considered "immersed" in computation domain $\Omega^+ \bigcup \Omega^-$. Consequently, we can define the computational sub-regions as follows:

Given a bounded domain $\Omega \subset R^2$, we consider sub-regions Ω_1 and Ω_2 to be the overlapping sub-regions satisfying

$$\Omega = \bigcup_{i=1}^{2} \Omega_i \tag{16}$$

where $\Omega_1 = \Omega^+ \bigcup \Omega^-$ is the computation domain containing the base matrix, and $\Omega_2 = \Omega^-$ is the computation domain containing the inclusion.

In the sub-region Ω_1, the total number of meshfree nodes consists of a set of component nodes that overlap and cover the domain Ω. The overlapping nodes include a set of structured nodes from sub-region Ω_1 and a set of interface-fitted nodes and interior nodes from sub-region Ω_2. Let $Z_I = \{x_l, l = 1, \ldots NP\}$ be the set of distinct nodes in Ω_1. NP indicates the total number of overlapping nodes in sub-region Ω_1. For each $x_l \in Z_1$, $\Psi_l\left(x\right)$ denotes the corresponding meshfree shape function. We define the meshfree interpolant of $\boldsymbol{u(x)}$ by the formula

$$u^I\left(x\right) = \sum_{I=1}^{NP} \Psi_I\left(x\right) u\left(x_I\right) = \sum_{I=1}^{NP} \Psi_I\left(x\right) u_I \quad \forall x \in \Omega_I \tag{17}$$

where $u_I = u\left(x_I\right)$ is called the 'generalized' displacement of node I. In general, conventional meshfree approximations are not interpolants, i.e., $u_I \neq u^I\left(x_I\right)$.

For this reason, special treatment [13, 44] is required to impose the essential boundary conditions on the global boundary $\partial\Omega$ of the model problem. In this study an alternative meshfree approximation that restores a weak Kronecker-delta property at the boundary, the convex meshfree approximation [1, 42], is utilized to allow the direct treatment of essential boundary conditions on $\partial\Omega$ for elasticity interface problem. We employ the generalized meshfree approximation method (GMF) [46, 47, 49] to obtain the first-order meshfree convex approximation.

The convex approximation space constructed by GMF method is a subspace of $H_0^1(\Omega)$ and conforms to the boundary conditions if the approximating domain is convex. Ideally this conforming meshfree approximation secures H^1-compatibility and the homogeneous Dirichlet jump condition across the interface is verified automatically. On the other hand, the meshfree approximation also introduces the non-locality [15] across the interface. This gives rise to a solution that exhibits a smearing near the interface. To remove the smearing, we invoke a second meshfree approximation in sub-region Ω_2, e.g. by zero extension in the sub-region Ω_1.

To be more precise, we let $\tilde{Z}_2 = \{\tilde{x}_l, l = 1, \ldots MP\} \subset Z_1$ be the subset nodes that contain the overlapping nodes in the sub-region Ω_2. MP is the total number of overlapping nodes in sub-region Ω_2. We also define $\tilde{\tilde{Z}}_2 = \left\{\tilde{\tilde{x}}_l, l = 1, \ldots IP\right\} \subset \tilde{Z}_1 \subset Z_1$ to be the subset nodes that collect the interface-fitted nodes along the boundary of sub-region Ω_2. IP is the total number of interface-fitted nodes. Analogously, every function $\tilde{u}_h(x) \in \tilde{V}_{h2}(\Omega_2)$ has a unique representation of the form

$$\tilde{u}_h(x) = \sum_{I=1}^{MP} \tilde{\Psi}_I(x)\, \tilde{u}_I \quad \forall x \in \Omega_2 \tag{18}$$

Since the sub-region Ω_2 is assumed to be convex, the subspace $\tilde{V}_{h2}(\Omega_2)$ is defined by

$$\tilde{V}_{h2}(\Omega_2) = \left\{v : v|_{\Omega_2} \in H^1(\Omega_2), v = 0 \text{ on } \partial\Omega \bigcup \partial\Omega_2\right\} \tag{19}$$

Apparently, the subspace $\tilde{V}_{h2}(\Omega_2)$ is not a subset of subspace $V_h(\Omega_1)$, i.e., $\tilde{V}_{h2}(\Omega_2) \not\subset V_h(\Omega_1)$. Since the approximations in sub-region Ω_1 and sub-region Ω_2 are constructed independently, the meshfree shape functions $\tilde{\Psi}_l(x)$ and $\Psi_1(x)$ of the same node $x_l \in \tilde{Z}_2 \subset Z_1$ are not necessarily the same. This is true in particular when the support of node x_l covers the interface Γ, i.e.,

$$\tilde{\Psi}_l(x) \neq \Psi_l(x) \quad \text{for } x_l \in \tilde{Z}_2 \subset Z_1 \text{ and } supp(x_l) \bigcap \Gamma \neq 0 \tag{20}$$

The support of shape function in sub-region Ω_2 is defined by

$$supp(x_l) = supp\left(\tilde{\Psi}_l(x)\right) = \left\{x| \in \Omega_2 \text{ and } \tilde{\Psi}_1(x) \neq 0\right\} \tag{21}$$

As a result, it leads to a non-conforming meshfree approximation and the continuity of displacement across the interface is not ensured.

4 Construction of Meshfree Approximation and Modified Variational Formulation

In standard mortar finite element method [10], the constraint equation is imposed weakly across the interface and is sufficient to guarantee an approximation with a consistency error of order h if the weak solution $\boldsymbol{u}$ is smooth enough. However in immersed meshfree method, the sharing nodes are well-defined in the subset of interface-fitted nodes $\tilde{\tilde{Z}}_2 = \left\{\tilde{\tilde{x}}_l, l = 1, \ldots IP\right\}$. Indeed, the non-conformity of approximation in immersed meshfree method is due to the non-matching meshfree shape functions in the overlapping domain. Since the sub-region Ω_2 is immersed in the sub-region Ω_1 ($\Omega_2 \subset \Omega_1$), we can redefine the approximation in sub-region Ω_1 such that $\tilde{V}_{h2}(\Omega_2) \subset V_h(\Omega_1)$ and $\tilde{\Psi}_l(x) = \Psi_l(x)$ *for* $x_l \in \tilde{Z}_2 \subset Z_1$. This can be achieved by decomposing the approximation in sub-region Ω_1 into two approximations in non-overlapping sub-domains ($\Omega_1 \backslash \Omega_2$ and Ω_2) and enforcing the nodal-wise continuity in displacement by introducing the Kronecker-delta property to the interface-fitted nodes set $\tilde{\tilde{Z}}_2$. Namely we define a new constrained discrete meshfree approximation space by

$$\hat{V}_h(\Omega) = \prod_{i=1}^{2} \tilde{V}_{hi}(\Omega_i) \tag{22}$$

where the definition of subspace $\tilde{V}_{h2}(\Omega_2) \subset \hat{V}_h(\Omega)$ is given in Eq. (19) and additional subspace $\tilde{V}_{h1}(\Omega_1)$ is defined by

$$\tilde{V}_{h1}(\Omega_1) = \left\{v : v\big|_{\Omega_1 \backslash \Omega_2} \in H^1(\Omega_1 \backslash \Omega_2), v = 0 \text{ on } \partial\Omega\right\} \subset \hat{\mathrm{V}}_{\mathrm{h}}(\Omega) \tag{23}$$

We note that $\tilde{V}_{hi}(\Omega_i)$, i = 1, 2 stand for the spaces of linear conforming meshfree approximations that satisfy homogeneous Dirichlet boundary conditions on $\partial\Omega \bigcap \partial\Omega_i$, i = 1, 2 in the sub-regions $\Omega_1 \backslash \Omega_2$ and Ω_2 respectively. Let $\tilde{Z}_1 = \{\tilde{x}_l, l = 1, \ldots LP\} \subset Z_1$ be the subset nodes such that

$$\tilde{Z}_1 = \left(Z_1 \backslash \tilde{Z}_2\right) \bigcup \tilde{\tilde{Z}}_2 \text{ and } LP\text{=}NP\text{-}MP\text{+}IP. \tag{24}$$

In the general case, it is difficult to construct a conforming approximation across the interface. We will look into the possibility of constructing a conforming meshfree approximation for the interface problem in the future. In this study, we consider to impose the displacement continuity weakly and point-wisely across the interface by introducing the Kronecker-delta property to the approximation

of the interface-fitted nodes. This can be done by either employing a singular kernel function [14, 47] to the interface-fitted nodes or applying a transformation method [13, 14] to those nodes whose supports cover the interface nodes. We have adopted the transformation method in this study. With the introduction of Kronecker-delta property to the shape functions at the interface-fitted nodes set $\tilde{\tilde{Z}}_2 = \left\{\tilde{\tilde{x}}_l, l = 1, \ldots IP\right\}$, the reconstructed meshfree shape functions in sub-region Ω_1 have the following form:

$$\Omega_1: \begin{cases} \hat{\Psi}_I^1(x) & \text{if } \mathrm{x}_I \in \tilde{Z}_1, x \in \Omega_1 \backslash \Omega_2 \\ \hat{\Psi}_I^2(x) = \tilde{\Psi}_I(x) & \text{if } \mathrm{x}_I \in \tilde{Z}_2,\ x \in \Omega_2 \\ \hat{\Psi}_I^1(x_J) = \hat{\Psi}_I^2(x_J) = \delta_{IJ} & \text{if } \mathrm{x}_I, \mathrm{x}_J \in \tilde{\tilde{Z}}_2 \end{cases} \tag{25}$$

satisfying

$$u_h(x) = \begin{cases} \sum_{I=1}^{LP} \hat{\Psi}_I^1(x)\, u_I & \text{if } \mathrm{x}_I \in \tilde{Z}_l,\ x \in \Omega_1 \backslash \Omega_2 \\ \sum_{I=1}^{MP} \hat{\Psi}_I^2(x)\, u_I = \sum_{\mathrm{I}=1}^{MP} \tilde{\Psi}_I(x)\, u_I & \text{if } \mathrm{x}_I \in \tilde{Z}_2\ \mathrm{x} \in \Omega_2 \\ \sum_{I=1}^{LP} \hat{\Psi}_I^1(x_J)\, u_I = \sum_{\mathrm{I}=1}^{MP} \hat{\Psi}_I^2(x_J)\, u_I = u_J & \text{if } \mathrm{x}_J \in \tilde{\tilde{Z}}_2 \end{cases}. \tag{26}$$

In general $\hat{\Psi}_I^1(x) \neq \hat{\Psi}_I^2(x)$ on interface Γ except at the interface nodes. Since the displacement continuity across the interface is only imposed point-wisely at the interface-fitted nodes, the weak form of Eq. (13) is reformulated based on the nonconforming meshfree approximation. We can now define the immersed meshfree solution of the elasticity interface problem as a function $u_h \in \hat{V}_h$ satisfying

$$\hat{a}(u_h, v_h) = \langle f,\ v_h, \rangle,\ \forall v_h \in \hat{V}_h \tag{27}$$

where

$$\hat{a}(u_h, v_h) = \underbrace{\int_{\Omega_1 \backslash \Omega_2} \varepsilon(u_h) \cdot C^+ \cdot \varepsilon(v_h)\, d\Omega + \int_{\Omega_2} \varepsilon(u_h) \cdot C^+ \cdot \varepsilon(v_h)\, d\Omega}_{\text{integrating using integration cells from base matrix}} + \underbrace{\int_{\Omega_2} \varepsilon(\mathrm{u_h}) \cdot \left(\mathrm{C}^- - \mathrm{C}^+\right) \cdot \varepsilon(\mathrm{v_h})\, \mathrm{d}\Omega}_{\text{integrating using integration cells from inclusion}} \tag{28}$$

$$\langle f,\ v_h \rangle = \underbrace{\int_{\Omega_1 \backslash \Omega_2} f^+ \cdot v_h d\Omega + \int_{\Omega_2} f^+ \cdot v_h d\Omega}_{\text{integrating using integration cells from base matrix}} + \underbrace{\int_{\Omega_2} \left(\mathrm{f}^- - \mathrm{f}^+\right) \cdot \mathrm{v_h} \mathrm{d}\Omega}_{\text{integrating using integration cells from inclusion}} \tag{29}$$

and the associated discrete (broken) energy norm is defined by

$$\|v_h\|_{1,h} = \hat{a}\,(v_h, v_h)^{\frac{1}{2}}\,,\ \mathrm{v_h} \in \hat{V}_h. \tag{30}$$

Using the Cauchy-Schwarz inequality and triangle inequality, it can be shown that the modified bilinear form $\hat{a}\,(\cdot,\cdot)$ is bounded on $\hat{V}_h \times \hat{V}_h$ with respect to the broken energy norm on Ω.

Lemma 1. *There exists a positive constant c_b such that for any $u_h,\ v_h \in \hat{V}_h$, we have*

$$|\hat{a}\,(u_h, v_h)| \leq c_b\,\|u_h\|_{1,h}\,\|v_h\|_{1,h}\quad . \tag{31}$$

Proof. Wu et al., [45, Lemma 3.1].

It is also not difficult to show that the modified bilinear form $\hat{a}\,(\cdot,\cdot)$ is positive-definite on $\hat{V}_h$.

Lemma 2. *There exists a positive constant c_c such that for any $v_h \in \hat{V}_h$, we have*

$$\hat{a}\,(v_h, v_h) \geq c_c\,\|v_h\|_{1,h}^2 \tag{32}$$

Proof. Observing that $\hat{a}\,(v_h, v_h) = 0$ implies v_h is constant. Since v_h vanishes on global boundary $\partial\Omega$ and satisfies continuity at the interface-fitted nodes, we have $v_h = 0$ in Ω and, thus, we ensure the coercivity of the modified bilinear form.

The uniqueness of the solution of the discretization problem (27) then follows by the Lax-Milgram theorem.

Theorem 1. *Let $u \in V$ be the solution of the variational problem (7). Then the discretized problem (27) in immersed meshfree method admits a uniqueness solution $u_h \in \hat{V}_h$.*

5 A Degenerated Mortar Method and Error Estimate

Since $\hat{a}\,(\cdot,\cdot)$ is coercive, we can apply the well-known second Strang's lemma [11] for the energy-norm error estimate.

$$\|u - u_h\|_{1,h} \leq C\left\{\inf_{\mathrm{v}\in\hat{\mathrm{V}}_h}\|\mathrm{u}-\mathrm{v}\|_{1,\mathrm{h}} + \sup_{\mathrm{w_h}\in\hat{\mathrm{V}}_h\setminus\{0\}}\frac{|\hat{a}\,(\mathrm{u}-\mathrm{u_h}, \mathrm{w_h})|}{\|\mathrm{w_h}\|_{1,\mathrm{h}}}\right\} \tag{33}$$

where the first term on the R.H.S. of inequality (33) is the best approximation error which can be obtained by the meshfree approximation error estimate and, the second term is the consistency error which comes from the nonconforming of $\hat{V}_h$.

Assume the regularity on the exact solution $u \in H^2(\Omega)$, we have the following error estimate by Céa's inequality [22]

$$\inf_{v \in \hat{V}_h} \|u - v\|_{1,h} \leq \|u - I_h u\|_{1,h} \leq ch\,|u|_{2,\Omega} \tag{34}$$

where h is the largest nodal support radius, i.e., $h = \sup_{I \in Z_I} \{\text{diam}\,(r_I)\}$. Note that, it is sufficient to choose the inverse tangent basis function and C^2 window function in GMF method to have $\boldsymbol{H}^2$ meshfree shape functions.

Because the interface Γ is associated with a one dimensional triangulation, we call this one dimensional triangulation on Γ, Ξ_h. Each integration segment $e_i \in \Xi_h$ is a boundary edge of integration cell (finite element triangulation) T_{h_2} in Ω_2. For the consistency error, we use the definition of $\hat{a}(\cdot,\cdot)$ in Eq. (28) and Galerkin orthogonality together with Green's theorem to yield

$$\begin{aligned}
\hat{a}(u - u_h, w_h) &= \hat{a}(u, w_h) - \langle f, w_h \rangle \\
&= \int_{\Omega_1 \setminus \Omega_2} \nabla \cdot \left(C^+ \cdot \nabla_s u\right) \cdot w_h \, d\Omega + \int_{\Omega_2} \nabla \cdot (C^- \cdot \nabla_s u) \cdot w_h \, d\Omega \\
&\quad - \int_{\Omega} f \cdot w_h \, d\Omega \\
&= \int_{\Omega_1 \setminus \Omega_2} \nabla \cdot \left(C^+ \cdot \nabla_s u\right) \cdot w_h \, d\Omega + \int_{\Omega_2} \nabla \cdot (C^- \cdot \nabla_s u) \cdot w_h \, d\Omega \\
&\quad - \left(\int_{\Omega_1 \setminus \Omega_2} \nabla \cdot \left(C^+ \cdot \nabla_s u\right) \cdot w_h \, d\Omega + \int_{\Omega_2} \nabla \cdot (C^- \cdot \nabla_s u) \cdot w_h \, d\Omega \right. \\
&\quad \left. - \int_{\Gamma} C \cdot \varepsilon(u) \cdot n \cdot w_h \, d\Gamma \right) \\
&= \sum\nolimits_{e_i \in \Xi_h} \int_{e_i} (C \cdot \varepsilon(u) \cdot n) \cdot [[w_h]] \, ds
\end{aligned} \tag{35}$$

where the term $\int_{e_i} (C \cdot \varepsilon(u) \cdot n) \cdot [[w_h]] \, ds$ can be realized as the weak constraint equation appearing in the Lagrange multiplier-type mortar method in which $\lambda_h = C \cdot \varepsilon(u) \cdot n \in \left(L^2(\Gamma)\right)^2$ are Lagrange multipliers. The symbol $[[w]]$ denotes the restriction of jump as defined in Eq. (2) for $w \in \Gamma$. A natural choice for the construction of Lagrange multiplier spaces in nonconforming formulation of the mortar method is to define the Lagrange multiplier basis function locally associated with the discrete nodes. From this point of view, we regard the proposed immersed meshfree method as a degenerated Lagrange multiplier-type mortar method due to the fact that the displacement jump vanishes at the interface-fitted nodes, i.e.,

$$[[w_h]]_I = \left(w_I^+ - w_I^-\right) = 0 \; \forall I \in \tilde{\tilde{Z}}_2. \tag{36}$$

Applying the point collocation method to the assembly of discrete constraint equation in mortar method and using Eq. (35) lead to the primal problem presented in Eq. (27).

$$\begin{aligned}\hat{a}\left(u, w_h\right)-\langle f, w_h\rangle &= \sum_{e_i \in \Xi_h} \int_{e_i} C : \varepsilon(u) \cdot n \cdot[[w_h]] \, ds \\ &= \sum_{e_i \subset \Xi_h} \sum_{I=1} C \cdot \varepsilon\left(u^I\right) \cdot n_I \cdot\left(w_I^+ - w_I^-\right)=0 \quad \forall w_h \in \hat{V}_h .\end{aligned} \tag{37}$$

Despite the vanishing constraint equation in the primal problem, the proposed immersed meshfree method still presents certain boundary quadrature error by different quadrature rules and requires a consistency error estimate to ensure a stable and convergent meshfree discretization. This consistency error estimate resembles the consistency error estimate of meshfree solution in standard Galerkin meshfree method using moving least-squares approximation [9] or reproducing kernel approximation [31] when Dirichlet boundary conditions are imposed point-wisely.

Lemma 3. *Assume that $u \in H^2(\Omega)$ be the solution of elasticity interface problem in Eq. (7), there exists a constant c_c independent of h and function $\boldsymbol{u}$ such that*

$$\left|\hat{a}\left(u, w_h\right)-\langle f, \ w_h\rangle\right| \leq c_c h\,|u|_2\,\left\|w_h\right\|_{I,h} \quad \forall w_h \in \hat{V}_h . \tag{38}$$

Proof. Wu et al. [45, Lemma 3.3].

Combining inequalities (33), (34) and (38), we obtain the following result for the energy error.

Proposition 1. *Assume $u \in H^2(\Omega)$ and $u_h \in \hat{V}_h$ be respectively the solutions of the weak problem (6) and of the discretized problem (27). Then it holds*

$$\left\|u-u_h\right\|_{I,h} \leq C_e h\,|u|_2 . \tag{39}$$

By Proposition 1, we expect the optimal rate of convergence for immersed meshfree method to be one for $\boldsymbol{u}$ in $H^2(\Omega)$. We also can obtain an optimal error estimate in L^2-norm for the problem (27). Assume that the conditions of Proposition 1 are met. According to the Aubin-Nitsche lemma [16], the immersed meshfree Galerkin solution $\boldsymbol{u}_h$ satisfies

$$\begin{aligned}\left\|u-u_h\right\|_0 \leq & \sup_{g \in L^2} \frac{C_2}{\|g\|_0}\left\{\left\|u-u_h\right\|_{1,h} \cdot\left\|\phi_g-\phi_h\right\|_{1,h}+\left|\hat{a}\left(u-u_h, \phi_g\right)-\left\langle u-u_h, g\right\rangle\right|\right. \\ & \left.+\left|\hat{a}\left(u, \phi_g-\phi_h\right)-\langle f, \phi-\phi_h\rangle\right|\right\} \\ = & \sup_{g \in L^2} \frac{C_2}{\|g\|_0}\left\{\left\|u-u_h\right\|_{1,h} \cdot\left\|\phi_g-\phi_h\right\|_{1,h}+\left|\hat{a}\left(\phi_g, u-u_h\right)-\left\langle g, u-u_h\right\rangle\right|\right. \\ & \left.+\left|\hat{a}\left(u, \phi_g-\phi_h\right)-\langle f, \phi-\phi_h\rangle\right|\right\}\end{aligned} \tag{40}$$

where for any $g \in L^2, \phi_g \in H^1$ is the unique solution of the variational problem

$$\hat{a}\left(\phi_g, v\right) = \langle g, v \rangle \text{ for any } \mathrm{v} \in \mathrm{H}^1 \tag{41}$$

We can also consider $\phi_g - \phi_h$is the discretization error for the problem (41). Using Proposition 1, we have

$$\left\|\phi_g - \phi_h\right\|_{1,h} \leq C_e h \left|\phi_g\right|_2 \leq C_e h \left\|g\right\|_0 \tag{42}$$

Therefore the first term on the R.H.S. of inequality (40) can be bounded by

$$\left\|u - u_h\right\|_{1,h} \cdot \left\|\phi_g - \phi_h\right\|_{1,h} \leq C_{e1} h^2 \left(|u|_2 \left\|g\right\|_0\right) \tag{43}$$

Using Lemma 3, the second term on the R.H.S. of inequality (40) can also be bounded by

$$\left|\hat{a}\left(\phi_g, u - u_h\right) - \langle g, u - u_h \rangle\right| \leq C_{e2} h \left|\phi_g\right|_2 \left\|u - u_h\right\|_{1,h} \leq C_{e2} h^2 \left\|g\right\|_0 |u|_2 \tag{44}$$

Using similar argument as above we obtain the third term on the R.H.S. of inequality (40) to be bounded by

$$\left|\hat{a}\left(u, \phi_g - \phi_h\right) - \langle f, \phi - \phi_h \rangle\right| \leq C_{e3} h^2 |u|_2 \left\|g\right\|_0 \tag{45}$$

Applying inequalities (43)–(45) to inequality (40) leads to the following error in the L^2 norm:

Proposition 2. *The L^2 discrete error of the immersed meshfree method is of order h^2*

$$\left\|u - u_h\right\|_{0,h} \leq C_l h^2 |u|_2 \tag{46}$$

6 Numerical Examples

In this section, we analyze two linear benchmark examples to study the performance of the proposed method in the interface elasticity problems. Unless otherwise specified, the following conditions are considered: (1) The weight function is chosen to be the cubic B-spline kernel function with normalized support size equal to 1.6 for the construction of meshfree shape functions. (2) A six-point Gauss quadrature rule is used in each integration cell for all examples. (3) The materials are considered to be compressible with Poisson ratio $v = 0.3$ and plane strain condition is assumed for the two-dimensional case. (4) For convergence, dimensionless unit system is adopted in this paper. (5) All the contour plots are reported at Gauss points for the stress field.

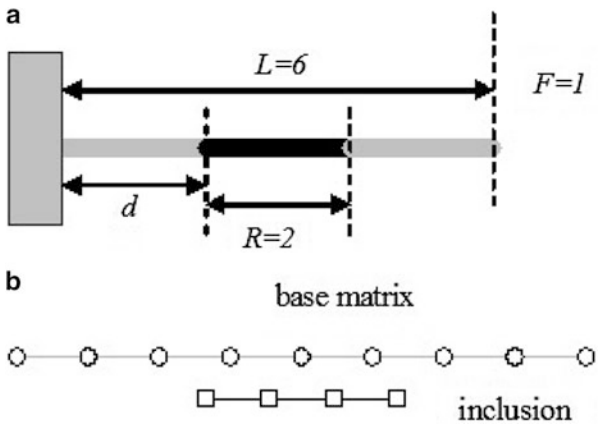


Fig. 2 One dimensional interface problem. (**a**) Graphical presentation of 1D model. (**b**) Discretization

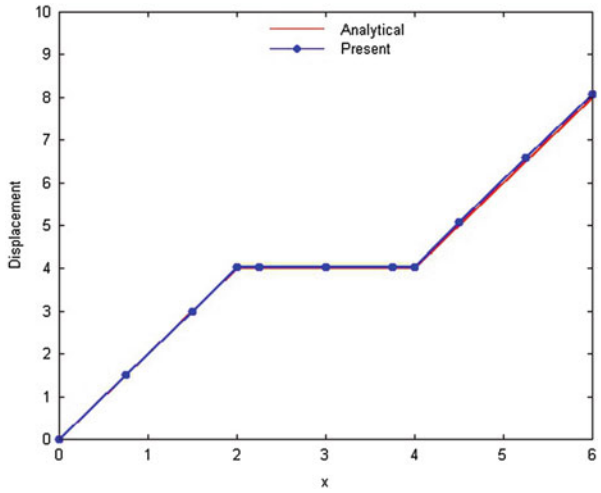


Fig. 3 Displacement distribution in d = 2.0 case

6.1 One-Dimensional Bi-Material Rod

A one-dimensional bi-material rod is subjected to a unit end force on the free end of the rod as described in Fig. 2a. The bi-material is made of elastic material with Young's modulus $E^+ = 1.0$ for base material and $E^- = 1{,}000.0$ for inclusion. The computational domain is discretized uniformly and separately for base matrix and inclusion as shown in Fig. 2b.

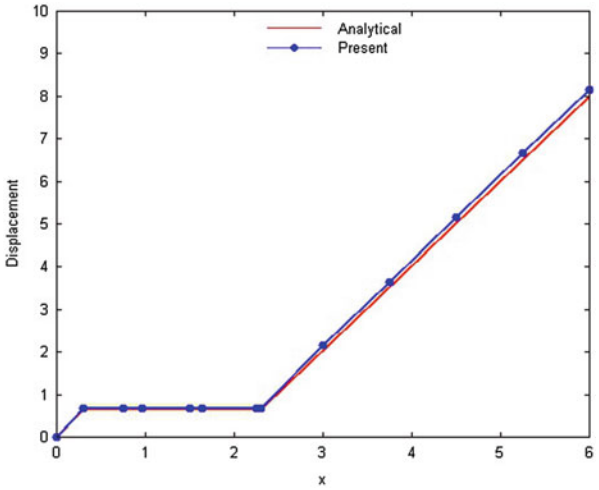


Fig. 4 Displacement distribution in d = 0.5 case

The problem is first analyzed with the inclusion located in the mid-way of the rod. Figure 3 depicts the nodal displacement of presented result where the numerical solution agrees with analytic solution very well.

In the second case, the inclusion is placed near the physical boundary. This is done by simply moving the whole inclusion mesh close to the fixed end and no repartition of the domain is required in this case. Compared to the exact solution, the presented immersed meshfree method remains to produce a very good result in displacement field using only 13 nodes as shown in Fig. 4. In both cases, the displacements are well-captured and no noticeable oscillations are observed near the interface.

The effect of particle refinement in the base material is also studied. The result of tip displacement against the total number of base matrix nodes is plotted in Fig. 5a which indicates a convergence in the displacement field as mesh is refined. In addition, the effect of nodal support size is investigated and presented in Fig. 5b. The non-sensitivity of the tip displacement to the nodal support size is an outcome of convex approximation. Similar observation has been reported elsewhere [49] in the structural analysis when meshfree convex approximation is adopted.

7 Cantilever Beam

In this example, accuracy and convergence of a beam problem for both single and composite solid models are studied. The problem statement and boundary conditions of the beam problem are given in Fig. 6a. For a single material model, it is assumed that the material contains only base matrix with Young's modulus

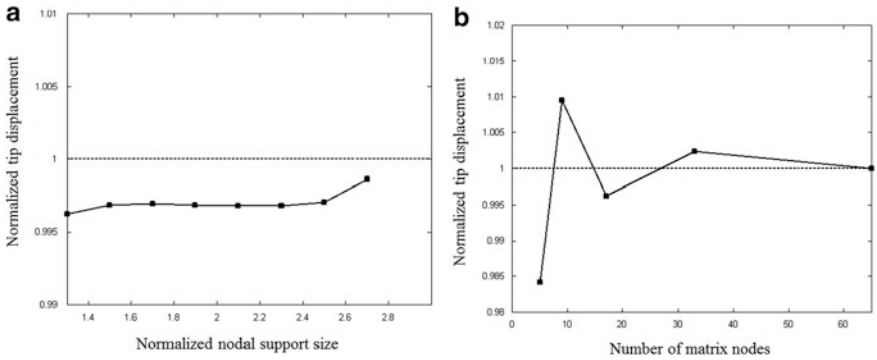


Fig. 5 Refinement and support size effects on normalized tip displacement. (**a**) Refinement effect. (**b**) Support size effect

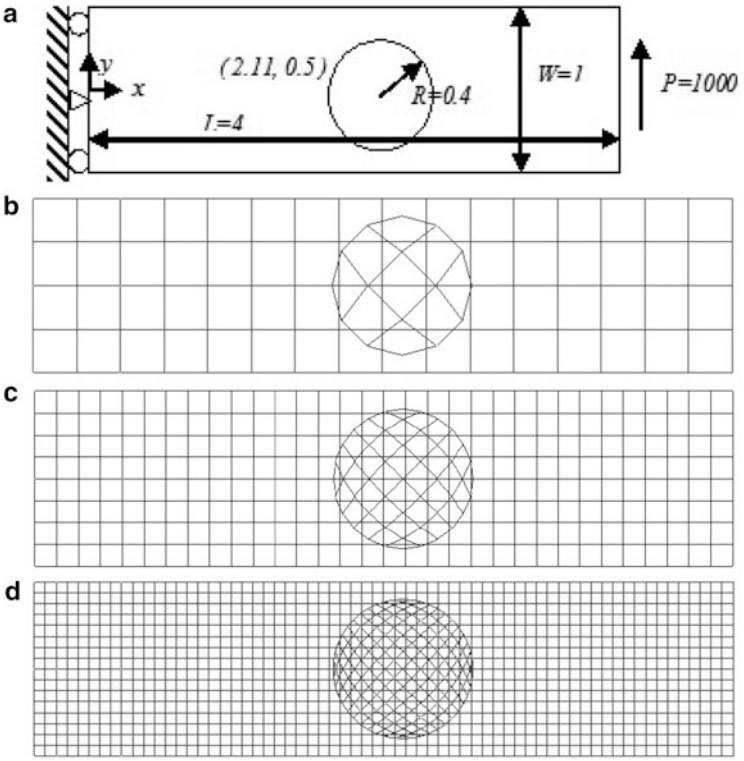


Fig. 6 Cantilever beam problem. (**a**) Geometry and boundary conditions. (**b**) 91 nodes. (**c**) 346 nodes. (**d**) 1,274 nodes

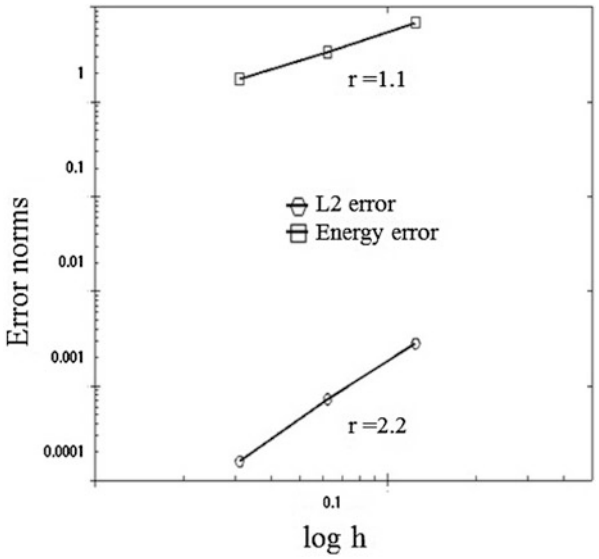


Fig. 7 L^2 and energy error norms in single material model

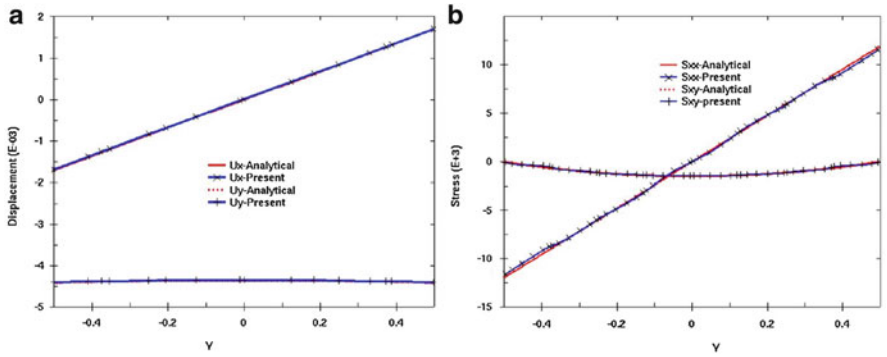


Fig. 8 Distribution of displacements along the cross-section x = 2.11 and stresses along the cross-section x = 2.01 in single material model (346 nodes). (**a**) Displacements. (**b**) Stresses

$E^+ = 2.0\text{e} + 07$. Three regularly refined meshfree discretizations for the beam model are shown in Fig. 6b.

The single material model is first analyzed using the overlapping mesh to verify the theoretical rate of convergence derived in the previous section. Since the beam is composed of one material, the analytical solution of the beam problem is available [43]. The results of L^2-norm and energy-norm errors against the element size are shown in Fig. 7. The proposed method achieves an optimal rate of convergence in both L^2-norm and energy-norm errors. Figure 8a displays the distribution of displacement along the cross-section x = 2.11 and compares with the

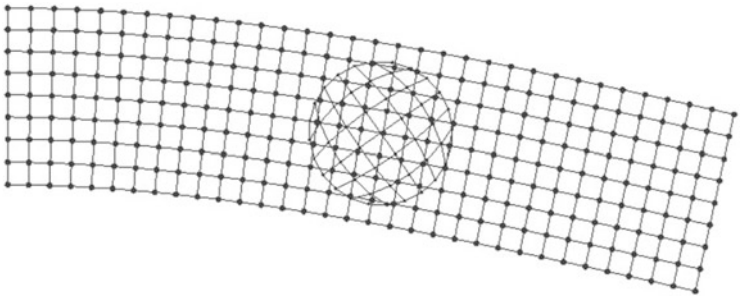


Fig. 9 Deformation plot in single material model (displacement scaled by 50 times): analytical (*mesh*), numerical (*black dot*)

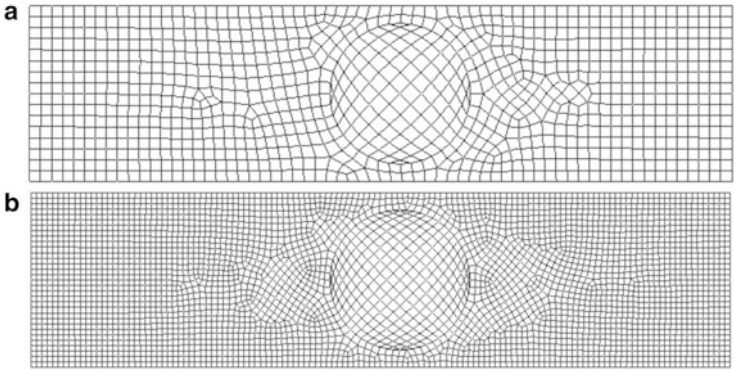


Fig. 10 Two conforming finite element meshes. (**a**) 1,074 nodes. (**b**) 4,055 nodes

analytical solution. Both displacement components are in good agreement with the analytical solution. Figure 8b compares the stresses distribution with the analytical solution along the cross-section x = 2.01 and good agreements are obtained. A more detailed comparison of displacement solution is shown in Fig. 9. Black dots in this figure denote the nodal locations obtained from the immersed meshfree method, while the mesh represents the analytical solution. In both solutions, the displacements are scaled by a factor of 50. Superior performance of the proposed method is apparent in this deformation plot.

In the composite solid model, we have chosen the Young's modulus of inclusion to be $E^- = 2.0\text{e} + 10$ which is 1,000 times higher than that of base matrix. Since the exact solution is not available, two reference solutions are generated from the conforming finite element method using the standard displacement-based bilinear element formulation. Two finite element meshes shown in Fig. 10a, b represent two level of mesh refinement that conform to the bi-material interface with total number of 1,074 and 4,055 nodes respectively.

A comparison of energy norm, L^2 norm and tip displacement using the immersed meshfree method and conforming finite element method are listed in Table 1.

Table 1 The convergence study in composite cantilever beam

	Energy norm	L^2 norm	Tip displacement
		Present	
91 nodes	3.3941	1.1480E−02	−1.1450E−02
346 nodes	3.3801	1.1352E−02	−1.1386E−02
1,274 nodes	3.3708	1.1338E−02	−1.1299E−02
		FEM	
4,055 nodes	3.3700	1.1336E−02	−1.1290E−02
1,074 nodes	3.3669	1.1320E−02	−1.1272E−02

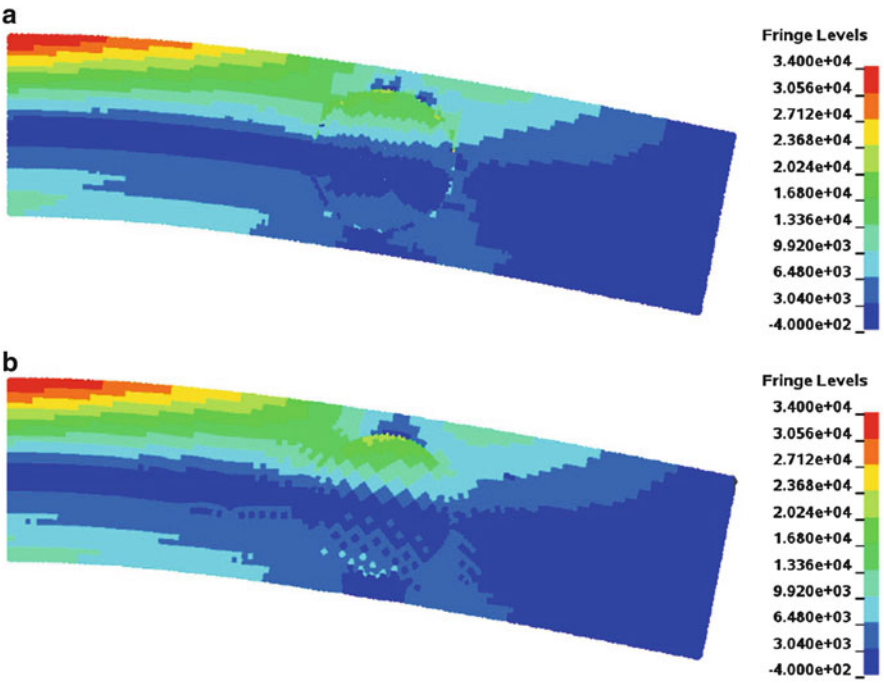


Fig. 11 Maximum principle stress contour on deformed plot (scaled by 50 times) in composite solid model. (**a**) Present solution (346 nodes). (**b**) Reference solution from conforming finite element mesh (1,074 nodes)

As shown in Table 1, the immersed meshfree solution converges to the finite element solution of the most refined mesh. Furthermore, the prediction of immersed meshfree method using a coarser discretization is comparable to the finite element solution with a finer mesh. Figure 11a, b show the comparison of maximum principle stress distribution in immersed meshfree method using 346 nodes and conforming finite element method using 1,074 nodes respectively. The stress contours are plotted on a deformed configuration with the scale factor of 50 in the displacement field. As shown in the comparison, the immersed meshfree method predicts similar results for the maximum principle stress.

8 Conclusion

We analyzed the elasticity interface problems using an immersed meshfree method. This method introduces a new discretization to approximate the overlapping sub-domains in the immersed structure. The method can be regarded as a non-conforming method which is related to the Lagrangian-type mortar method in treating the material discontinuity across the interface. In contrast to the existing Lagrangian-type mortar finite element and mortar meshfree methods, the proposed approach eliminates the interface constraint equation explicitly from the variational formulation by introducing the new meshfree discretization and a point-wise continuity across the material interface, thus involves no control or stabilization parameters. This unique property offers a great flexibility over the other methods with respect to the adoption of overlapping mesh in the immersed structure analyses. Theoretical results such as uniform ellipticity and consistency error are given in separate lemmas. Optimal energy and L^2 norm error estimates are provided and illustrated by numerical experiments. Our numerical results indicate that the solution is comparable with the solution obtained from the conforming finite element method.

An extension of the present study to the large-scale nonlinear cases is trivial but computationally expensive due to the high-order integration rule. The employment of conjugate gradient solvers in conjunction with appropriate preconditioning methods and parallelization will be considered in the future to improve the computation efficiency in the large-scale analysis. Attempts such as nodal integration, a coupling of meshfree approximation with finite element shape function [44] and the introduction of meshfree-enriched finite element approximation [46, 48] to the interface-fitted nodes are under investigation.

References

1. M. Arroyo, M. Ortiz, Local maximum-entropy approximation schemes: a seamless bridge between finite elements and meshfree methods. Int. J. Numer. Methods Eng. **65**, 2167–2202 (2006)
2. I. Babuška, The finite element method for elliptic equations with discontinuous coefficients. Computing **5**, 207–213 (1970)
3. I. Babuška, The finite element method with Lagrangian multipliers. Numer. Math. **20**, 179–192 (1973)
4. I. Babuška, U. Banerjee, J. Osborn, Generalized finite element methods – main ideas, results and perspective. Int. J. Comput. Methods **1**, 67–103 (2004)
5. I. Babuška, G. Caloz, J. Osborn, Special finite element methods for a class of second order elliptic problems with rough coefficients. SIAM J. Numer. Anal. **31**, 945–981 (1994)
6. I. Babuška, J.M. Melenk, The partition of unity finite element method: basic theory and applications. Comput. Methods Appl. Mech. Eng. **4**, 289–314 (1996)
7. H.J.C. Barbosa, T.J.R. Hughes, The finite element method with Lagrangian multipliers on the boundary: circumventing the Babuška-Brezzi condition. Comput. Methods Appl. Mech. Eng. **85**, 109–128 (1991)

8. T. Belytschko, T. Black, Elastic crack growth in finite elements with minimal remeshing. Int. J. Numer. Methods Eng. **45**, 610–620 (1999)
9. T. Belytschko, Y.Y. Lu, L. Gu, Element-free Galerkin methods. Int. J. Numer. Methods Eng. **37**, 229–256 (1994)
10. C. Bernardi, Y. Maday, A. Patera, Domain decomposition by the mortar element method, in *Asymptotic and Numerical Methods for Partial Differential Equations with Critical Parameters*, ed. by H.G. Kaper, M. Garbey. NATO ASI (Kluwer Academic, Dordrecht, 1993), pp. 269–286
11. S.C. Brenner, L.R. Scott, *The Mathematical Theory of Finite Element Methods*, 3rd edn. (Springer, New York, 2008)
12. F. Brezzi, M. Fortin, *Mixed and Hybrid Finite Element Methods* (Springer, New York, 1991)
13. J.S. Chen, C. Pan, C.T. Wu, W.K. Liu, Reproducing kernel particle methods for large deformation analysis of non-linear structures. Comput. Methods Appl. Mech. Eng. **139**, 195–227 (1996)
14. J.S. Chen, H.P. Wang, New boundary condition treatments in meshfree computation of contact problems. Comput. Methods Appl. Mech. Eng. **187**, 441–468 (2000)
15. J.S. Chen, C.T. Wu, T. Belytschko, Regularization of material instabilities by meshfree approximations with intrinsic length scales. Int. J. Numer. Methods Eng. **47**, 1303–1322 (2000)
16. P.G. Ciarlet, *The Finite Element Method for Elliptic Problem* (North-Holland, Amsterdam 1978)
17. L.W. Cordes, B. Moran, Treatment of material discontinuity in the element-free Galerkin method. Comput. Methods Appl. Mech. Eng. **139**, 75–89 (1996)
18. J. Dolbow, I. Harari, An efficient finite element method for embedded interface problem. Int. J. Numer. Methods Eng. **78**, 229–252 (2009)
19. J. Dolbow, N. Moës, T. Belytschko, An extended finite element method for modeling crack growth with frictional contact. Comput. Methods Appl. Mech. Eng. **190**, 6825–6846 (2009)
20. C.A. Duarte, I. Babuška, J.T. Oden, Generalized finite element methods for three-dimensional structural mechanics problems. Comput. Struct. **77**(2), 215–232 (2000)
21. A. Düster, J. Parvizian, Z. Yang, E. Rank, The finite cell method for three-dimensional problems for solid mechanics. Comput. Methods Appl. Mech. Eng. **197**, 3768–3782 (2008)
22. W. Han, X. Meng, Error analysis for the reproducing kernel particle method. Comput. Methods Appl. Mech. Eng. **190**, 157–466 (2001)
23. P. Hansbo, Nitsche's method for interface problems in computational mechanics. GAMM-Mitt. **28**, 183–206 (2005)
24. A. Hansbo, P. Hansbo, An unfitted finite element method, based on Nitsche's method, for elliptic interface problems. Comput. Methods Appl. Mech. Eng. **191**, 5537–5552 (2002)
25. T.J.R. Hughes, G.R. Feijóo, L. Mazzei, J-B. Quincy, The variational multiscale method – a paradigm for computational mechanics. Comput. Methods Appl. Mech. Eng. **166**, 3–24 (1998)
26. Y. Krongauz, T. Belytschko, EFG approximation with discontinuous derivative. Int. J. Numer. Methods Eng. **41**, 1215–1233 (1995)
27. R.J. LeVeque, Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. SIAM J. Numer. Anal. **31**, 1019–1044 (1994)
28. Z. Li, The immersed interface method using a finite element formulation. Appl. Numer. Math. **27**, 253–267 (1998)
29. Z. Li, T. Lin, X. Wu, New Cartesian grid methods for interface problems using the finite element formulation. Numer. Math. **96**, 61–98 (2003)
30. Z. Li, X. Yang, An immersed finite element method for elasticity equations with interfaces. AMS Contemp. Math. **383**, 285–298 (2005)
31. W.K. Liu, S. Jun, S. Li, J. Adee, T. Belytschko, Reproducing kernel particle methods for structural dynamics. Int. J. Numer. Methods Eng. **38**, 1655–1679 (1995)
32. H. Lu, D. W. Kim, W. K. Liu, Treatment of discontinuity in the reproducing kernel element method. Int. J. Numer. Methods Eng. **63**, 241–253 (2005)
33. N. Moës, M. Cloirec, P. Cartraud, J.F. Remacle, A computational approach to handle complex microstructure geometries. Comput. Methods Appl. Mech. Eng. **192**, 3163–3177 (2003)

34. H.M. Mourad, J. Dolbow, I. Harari, A bubble-stabilized finite element method for Dirichlet constraints on embedded interfaces. Int. J. Numer. Methods Eng. **69**, 1–21 (2007)
35. J. Parvizian, A. Düster, E. Rank, Finite cell method. Comput. Mech. **41**, 121–133 (2007)
36. T. Rüberg, F. Cirak, An immersed finite element method with integral equation correction. Int. J. Numer. Methods Eng. **86**, 93–114 (2011)
37. J.D. Sanders, J.E. Dolbow, T.A. Laursen, On methods for stabilizing constraints over enriched interfaces in elasticity. Int. J. Numer. Methods Eng. **78**, 1009–1036 (2009)
38. S.A. Sauter, R. Warnke, Composite finite elements for elliptic boundary value problems with discontinuous coefficients. Computing **77**, 29–55 (2006)
39. R. Stenberg, On some techniques for approximating boundary conditions in the finite element method. J. Comput. Appl. Math. **63**, 139–148 (1995)
40. K.R. Srinivasan, K. Matouš, P.H. Geubelle, Generalized finite element method for modeling nearly incompressible biomaterial hyperelastic solids. Comput. Methods Appl. Mech. Eng. **197**, 4882–4893 (2008)
41. T. Stroubolis, K. Copps, I. Babuška, The generalized finite element method. Comput. Methods Appl. Mech. Eng. **190**, 4081–4193 (2001)
42. N. Sukumar, Construction of polygonal interpolants: a maximum entropy approach. Int. J. Numer. Methods Eng. **61**, 2159–2181 (2004)
43. S.P. Timoshenko, J.N. Goodier, *Theory of Elasticity* (McGraw-Hill, New York, 1970)
44. H.P. Wang, C.T. Wu, Y. Guo, M.E. Botkin, A coupled meshfree/finite element method for automotive crashworthiness simulations. Int. J. Impact Eng. **36**, 1210–1222 (2009)
45. C.T. Wu, Y. Guo, E. Askari, Numerical modeling of composite solids using an immersed meshfree Galerkin method. Composites B **45**, 1397–1413 (2003)
46. C.T. Wu, W. Hu, Meshfree-enriched simplex elements with strain smoothing for the finite element analysis of compressible and near-incompressible solids. Comput. Methods Appl. Mech. Eng. **200**, 2991–3010 (2011)
47. C.T. Wu, M. Koishi, A meshfree procedure for the microscopic analysis of particle-reinforced rubber compounds. Interact. Multiscale Mech. **2**, 147–169 (2009)
48. C.T. Wu, M. Koishi, Three-dimensional meshfree-enriched finite element formulation for micromechanical hyperelastic modeling of particulate rubber composites. Int. J. Numer. Methods Eng. **91**, 1137–1157 (2012)
49. C.T. Wu, C.K. Park, J.S. Chen, A generalized meshfree approximation for the meshfree analysis of solids. Int. J. Numer. Methods Eng. **85**, 693–722 (2011)
50. K.H. Yang, A.I. King, A limited review of finite element models developed for brain injury biomechanics research. Int. J. Veh. Des. **32**, 116–129 (2003)

Editorial Policy

1. Volumes in the following three categories will be published in LNCSE:

i) Research monographs
ii) Tutorials
iii) Conference proceedings

Those considering a book which might be suitable for the series are strongly advised to contact the publisher or the series editors at an early stage.

2. Categories i) and ii). Tutorials are lecture notes typically arising via summer schools or similar events, which are used to teach graduate students. These categories will be emphasized by Lecture Notes in Computational Science and Engineering. **Submissions by interdisciplinary teams of authors are encouraged.** The goal is to report new developments – quickly, informally, and in a way that will make them accessible to non-specialists. In the evaluation of submissions timeliness of the work is an important criterion. Texts should be well-rounded, well-written and reasonably self-contained. In most cases the work will contain results of others as well as those of the author(s). In each case the author(s) should provide sufficient motivation, examples, and applications. In this respect, Ph.D. theses will usually be deemed unsuitable for the Lecture Notes series. Proposals for volumes in these categories should be submitted either to one of the series editors or to Springer-Verlag, Heidelberg, and will be refereed. A provisional judgement on the acceptability of a project can be based on partial information about the work: a detailed outline describing the contents of each chapter, the estimated length, a bibliography, and one or two sample chapters – or a first draft. A final decision whether to accept will rest on an evaluation of the completed work which should include

– at least 100 pages of text;
– a table of contents;
– an informative introduction perhaps with some historical remarks which should be accessible to readers unfamiliar with the topic treated;
– a subject index.

3. Category iii). Conference proceedings will be considered for publication provided that they are both of exceptional interest and devoted to a single topic. One (or more) expert participants will act as the scientific editor(s) of the volume. They select the papers which are suitable for inclusion and have them individually refereed as for a journal. Papers not closely related to the central topic are to be excluded. Organizers should contact the Editor for CSE at Springer at the planning stage, see *Addresses* below.

In exceptional cases some other multi-author-volumes may be considered in this category.

4. Only works in English will be considered. For evaluation purposes, manuscripts may be submitted in print or electronic form, in the latter case, preferably as pdf- or zipped ps-files. Authors are requested to use the LaTeX style files available from Springer at http://www.springer.com/gp/authors-editors/book-authors-editors/manuscript-preparation/5636 (Click on LaTeX Template → mongraphs or contributed books).

For categories ii) and iii) we strongly recommend that all contributions in a volume be written in the same LaTeX version, preferably LaTeX2e. Electronic material can be included if appropriate. Please contact the publisher.

Careful preparation of the manuscripts will help keep production time short besides ensuring satisfactory appearance of the finished book in print and online.

5. The following terms and conditions hold. Categories i), ii) and iii):

Authors receive 50 free copies of their book. No royalty is paid.
Volume editors receive a total of 50 free copies of their volume to be shared with authors, but no royalties.

Authors and volume editors are entitled to a discount of 33.3 % on the price of Springer books purchased for their personal use, if ordering directly from Springer.

6. Springer secures the copyright for each volume.

Addresses:

Timothy J. Barth
NASA Ames Research Center
NAS Division
Moffett Field, CA 94035, USA
barth@nas.nasa.gov

Michael Griebel
Institut für Numerische Simulation
der Universität Bonn
Wegelerstr. 6
53115 Bonn, Germany
griebel@ins.uni-bonn.de

David E. Keyes
Mathematical and Computer Sciences
and Engineering
King Abdullah University of Science
and Technology
P.O. Box 55455
Jeddah 21534, Saudi Arabia
david.keyes@kaust.edu.sa

and

Department of Applied Physics
and Applied Mathematics
Columbia University
500 W. 120 th Street
New York, NY 10027, USA
kd2112@columbia.edu

Risto M. Nieminen
Department of Applied Physics
Aalto University School of Science
and Technology
00076 Aalto, Finland
risto.nieminen@aalto.fi

Dirk Roose
Department of Computer Science
Katholieke Universiteit Leuven
Celestijnenlaan 200A
3001 Leuven-Heverlee, Belgium
dirk.roose@cs.kuleuven.be

Tamar Schlick
Department of Chemistry
and Courant Institute
of Mathematical Sciences
New York University
251 Mercer Street
New York, NY 10012, USA
schlick@nyu.edu

Editor for Computational Science
and Engineering at Springer:
Martin Peters
Springer-Verlag
Mathematics Editorial IV
Tiergartenstrasse 17
69121 Heidelberg, Germany
martin.peters@springer.com

Lecture Notes in Computational Science and Engineering

1. D. Funaro, *Spectral Elements for Transport-Dominated Equations.*
2. H.P. Langtangen, *Computational Partial Differential Equations.* Numerical Methods and Diffpack Programming.
3. W. Hackbusch, G. Wittum (eds.), *Multigrid Methods V.*
4. P. Deuflhard, J. Hermans, B. Leimkuhler, A.E. Mark, S. Reich, R.D. Skeel (eds.), *Computational Molecular Dynamics: Challenges, Methods, Ideas.*
5. D. Kröner, M. Ohlberger, C. Rohde (eds.), *An Introduction to Recent Developments in Theory and Numerics for Conservation Laws.*
6. S. Turek, *Efficient Solvers for Incompressible Flow Problems.* An Algorithmic and Computational Approach.
7. R. von Schwerin, ***M**ulti **B**ody **S**ystem **SIM**ulation.* Numerical Methods, Algorithms, and Software.
8. H.-J. Bungartz, F. Durst, C. Zenger (eds.), *High Performance Scientific and Engineering Computing.*
9. T.J. Barth, H. Deconinck (eds.), *High-Order Methods for Computational Physics.*
10. H.P. Langtangen, A.M. Bruaset, E. Quak (eds.), *Advances in Software Tools for Scientific Computing.*
11. B. Cockburn, G.E. Karniadakis, C.-W. Shu (eds.), *Discontinuous Galerkin Methods.* Theory, Computation and Applications.
12. U. van Rienen, *Numerical Methods in Computational Electrodynamics.* Linear Systems in Practical Applications.
13. B. Engquist, L. Johnsson, M. Hammill, F. Short (eds.), *Simulation and Visualization on the Grid.*
14. E. Dick, K. Riemslagh, J. Vierendeels (eds.), *Multigrid Methods VI.*
15. A. Frommer, T. Lippert, B. Medeke, K. Schilling (eds.), *Numerical Challenges in Lattice Quantum Chromodynamics.*
16. J. Lang, *Adaptive Multilevel Solution of Nonlinear Parabolic PDE Systems.* Theory, Algorithm, and Applications.
17. B.I. Wohlmuth, *Discretization Methods and Iterative Solvers Based on Domain Decomposition.*
18. U. van Rienen, M. Günther, D. Hecht (eds.), *Scientific Computing in Electrical Engineering.*
19. I. Babuška, P.G. Ciarlet, T. Miyoshi (eds.), *Mathematical Modeling and Numerical Simulation in Continuum Mechanics.*
20. T.J. Barth, T. Chan, R. Haimes (eds.), *Multiscale and Multiresolution Methods.* Theory and Applications.
21. M. Breuer, F. Durst, C. Zenger (eds.), *High Performance Scientific and Engineering Computing.*
22. K. Urban, *Wavelets in Numerical Simulation.* Problem Adapted Construction and Applications.
23. L.F. Pavarino, A. Toselli (eds.), *Recent Developments in Domain Decomposition Methods.*

24. T. Schlick, H.H. Gan (eds.), *Computational Methods for Macromolecules: Challenges and Applications.*

25. T.J. Barth, H. Deconinck (eds.), *Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics*.

26. M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations*.

27. S. Müller, *Adaptive Multiscale Schemes for Conservation Laws*.

28. C. Carstensen, S. Funken, W. Hackbusch, R.H.W. Hoppe, P. Monk (eds.), *Computational Electromagnetics*.

29. M.A. Schweitzer, *A Parallel Multilevel Partition of Unity Method for Elliptic Partial Differential Equations*.

30. T. Biegler, O. Ghattas, M. Heinkenschloss, B. van Bloemen Waanders (eds.), *Large-Scale PDE-Constrained Optimization*.

31. M. Ainsworth, P. Davies, D. Duncan, P. Martin, B. Rynne (eds.), *Topics in Computational Wave Propagation*. Direct and Inverse Problems.

32. H. Emmerich, B. Nestler, M. Schreckenberg (eds.), *Interface and Transport Dynamics.* Computational Modelling.

33. H.P. Langtangen, A. Tveito (eds.), *Advanced Topics in Computational Partial Differential Equations.* Numerical Methods and Diffpack Programming.

34. V. John, *Large Eddy Simulation of Turbulent Incompressible Flows.* Analytical and Numerical Results for a Class of LES Models.

35. E. Bänsch (ed.), *Challenges in Scientific Computing - CISC 2002.*

36. B.N. Khoromskij, G. Wittum, *Numerical Solution of Elliptic Differential Equations by Reduction to the Interface.*

37. A. Iske, *Multiresolution Methods in Scattered Data Modelling.*

38. S.-I. Niculescu, K. Gu (eds.), *Advances in Time-Delay Systems.*

39. S. Attinger, P. Koumoutsakos (eds.), *Multiscale Modelling and Simulation.*

40. R. Kornhuber, R. Hoppe, J. Périaux, O. Pironneau, O. Wildlund, J. Xu (eds.), *Domain Decomposition Methods in Science and Engineering.*

41. T. Plewa, T. Linde, V.G. Weirs (eds.), *Adaptive Mesh Refinement – Theory and Applications.*

42. A. Schmidt, K.G. Siebert, *Design of Adaptive Finite Element Software.* The Finite Element Toolbox ALBERTA.

43. M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations II.*

44. B. Engquist, P. Lötstedt, O. Runborg (eds.), *Multiscale Methods in Science and Engineering.*

45. P. Benner, V. Mehrmann, D.C. Sorensen (eds.), *Dimension Reduction of Large-Scale Systems.*

46. D. Kressner, *Numerical Methods for General and Structured Eigenvalue Problems.*

47. A. Boriçi, A. Frommer, B. Joó, A. Kennedy, B. Pendleton (eds.), *QCD and Numerical Analysis III*.

48. F. Graziani (ed.), *Computational Methods in Transport.*

49. B. Leimkuhler, C. Chipot, R. Elber, A. Laaksonen, A. Mark, T. Schlick, C. Schütte, R. Skeel (eds.), *New Algorithms for Macromolecular Simulation.*

50. M. Bücker, G. Corliss, P. Hovland, U. Naumann, B. Norris (eds.), *Automatic Differentiation: Applications, Theory, and Implementations.*

51. A.M. Bruaset, A. Tveito (eds.), *Numerical Solution of Partial Differential Equations on Parallel Computers.*

52. K.H. Hoffmann, A. Meyer (eds.), *Parallel Algorithms and Cluster Computing.*

53. H.-J. Bungartz, M. Schäfer (eds.), *Fluid-Structure Interaction.*

54. J. Behrens, *Adaptive Atmospheric Modeling.*

55. O. Widlund, D. Keyes (eds.), *Domain Decomposition Methods in Science and Engineering XVI.*

56. S. Kassinos, C. Langer, G. Iaccarino, P. Moin (eds.), *Complex Effects in Large Eddy Simulations.*

57. M. Griebel, M.A Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations III.*

58. A.N. Gorban, B. Kégl, D.C. Wunsch, A. Zinovyev (eds.), *Principal Manifolds for Data Visualization and Dimension Reduction.*

59. H. Ammari (ed.), *Modeling and Computations in Electromagnetics: A Volume Dedicated to Jean-Claude Nédélec.*

60. U. Langer, M. Discacciati, D. Keyes, O. Widlund, W. Zulehner (eds.), *Domain Decomposition Methods in Science and Engineering XVII.*

61. T. Mathew, *Domain Decomposition Methods for the Numerical Solution of Partial Differential Equations.*

62. F. Graziani (ed.), *Computational Methods in Transport: Verification and Validation.*

63. M. Bebendorf, *Hierarchical Matrices.* A Means to Efficiently Solve Elliptic Boundary Value Problems.

64. C.H. Bischof, H.M. Bücker, P. Hovland, U. Naumann, J. Utke (eds.), *Advances in Automatic Differentiation.*

65. M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations IV*.

66. B. Engquist, P. Lötstedt, O. Runborg (eds.), *Multiscale Modeling and Simulation in Science.*

67. I.H. Tuncer, Ü. Gülcat, D.R. Emerson, K. Matsuno (eds.), *Parallel Computational Fluid Dynamics 2007.*

68. S. Yip, T. Diaz de la Rubia (eds.), *Scientific Modeling and Simulations*.

69. A. Hegarty, N. Kopteva, E. O'Riordan, M. Stynes (eds.), *BAIL* 2008 – *Boundary and Interior Layers.*

70. M. Bercovier, M.J. Gander, R. Kornhuber, O. Widlund (eds.), *Domain Decomposition Methods in Science and Engineering XVIII.*

71. B. Koren, C. Vuik (eds.), *Advanced Computational Methods in Science and Engineering.*

72. M. Peters (ed.), *Computational Fluid Dynamics for Sport Simulation.*

73. H.-J. Bungartz, M. Mehl, M. Schäfer (eds.), *Fluid Structure Interaction II - Modelling, Simulation, Optimization.*

74. D. Tromeur-Dervout, G. Brenner, D.R. Emerson, J. Erhel (eds.), *Parallel Computational Fluid Dynamics 2008.*

75. A.N. Gorban, D. Roose (eds.), *Coping with Complexity: Model Reduction and Data Analysis.*

76. J.S. Hesthaven, E.M. Rønquist (eds.), *Spectral and High Order Methods for Partial Differential Equations.*

77. M. Holtz, *Sparse Grid Quadrature in High Dimensions with Applications in Finance and Insurance*.

78. Y. Huang, R. Kornhuber, O.Widlund, J. Xu (eds.), *Domain Decomposition Methods in Science and Engineering XIX*.

79. M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations V*.

80. P.H. Lauritzen, C. Jablonowski, M.A. Taylor, R.D. Nair (eds.), *Numerical Techniques for Global Atmospheric Models*.

81. C. Clavero, J.L. Gracia, F.J. Lisbona (eds.), *BAIL 2010 – Boundary and Interior Layers, Computational and Asymptotic Methods*.

82. B. Engquist, O. Runborg, Y.R. Tsai (eds.), *Numerical Analysis and Multiscale Computations*.

83. I.G. Graham, T.Y. Hou, O. Lakkis, R. Scheichl (eds.), *Numerical Analysis of Multiscale Problems*.

84. A. Logg, K.-A. Mardal, G. Wells (eds.), *Automated Solution of Differential Equations by the Finite Element Method*.

85. J. Blowey, M. Jensen (eds.), *Frontiers in Numerical Analysis - Durham 2010*.

86. O. Kolditz, U.-J. Gorke, H. Shao, W. Wang (eds.), *Thermo-Hydro-Mechanical-Chemical Processes in Fractured Porous Media - Benchmarks and Examples*.

87. S. Forth, P. Hovland, E. Phipps, J. Utke, A. Walther (eds.), *Recent Advances in Algorithmic Differentiation*.

88. J. Garcke, M. Griebel (eds.), *Sparse Grids and Applications*.

89. M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations VI*.

90. C. Pechstein, *Finite and Boundary Element Tearing and Interconnecting Solvers for Multiscale Problems*.

91. R. Bank, M. Holst, O. Widlund, J. Xu (eds.), *Domain Decomposition Methods in Science and Engineering XX*.

92. H. Bijl, D. Lucor, S. Mishra, C. Schwab (eds.), *Uncertainty Quantification in Computational Fluid Dynamics*.

93. M. Bader, H.-J. Bungartz, T. Weinzierl (eds.), *Advanced Computing*.

94. M. Ehrhardt, T. Koprucki (eds.), *Advanced Mathematical Models and Numerical Techniques for Multi-Band Effective Mass Approximations*.

95. M. Azaïez, H. El Fekih, J.S. Hesthaven (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2012*.

96. F. Graziani, M.P. Desjarlais, R. Redmer, S.B. Trickey (eds.), *Frontiers and Challenges in Warm Dense Matter*.

97. J. Garcke, D. Pflüger (eds.), *Sparse Grids and Applications – Munich 2012*.

98. J. Erhel, M. Gander, L. Halpern, G. Pichot, T. Sassi, O. Widlund (eds.), *Domain Decomposition Methods in Science and Engineering XXI*.

99. R. Abgrall, H. Beaugendre, P.M. Congedo, C. Dobrzynski, V. Perrier, M. Ricchiuto (eds.), *High Order Nonlinear Numerical Methods for Evolutionary PDEs - HONOM 2013*.

100. M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations VII*.

101. R. Hoppe (ed.), *Optimization with PDE Constraints – ESF Networking Program 'OPTPDE'*.

102. S. Dahlke, W. Dahmen, M. Griebel, W. Hackbusch, K. Ritter, R. Schneider, C. Schwab, H. Yserentant (eds.), *Extraction of Quantifiable Information from Complex Systems*.

103. A. Abdulle, S. Deparis, D. Kressner, F. Nobile, M. Picasso (eds.), *Numerical Mathematics and Advanced Applications - ENUMATH 2013*.

For further information on these books please have a look at our mathematics catalogue at the following URL: www.springer.com/series/3527

Monographs in Computational Science and Engineering

1. J. Sundnes, G.T. Lines, X. Cai, B.F. Nielsen, K.-A. Mardal, A. Tveito, *Computing the Electrical Activity in the Heart.*

For further information on this book, please have a look at our mathematics catalogue at the following URL: www.springer.com/series/7417

Texts in Computational Science and Engineering

1. H. P. Langtangen, *Computational Partial Differential Equations.* Numerical Methods and Diffpack Programming. 2nd Edition
2. A. Quarteroni, F. Saleri, P. Gervasio, *Scientific Computing with MATLAB and Octave.* 4th Edition
3. H. P. Langtangen, *Python Scripting for Computational Science*. 3rd Edition
4. H. Gardner, G. Manduchi, *Design Patterns for e-Science*.
5. M. Griebel, S. Knapek, G. Zumbusch, *Numerical Simulation in Molecular Dynamics*.
6. H. P. Langtangen, *A Primer on Scientific Programming with Python.* 4th Edition
7. A. Tveito, H. P. Langtangen, B. F. Nielsen, X. Cai, *Elements of Scientific Computing.*
8. B. Gustafsson, *Fundamentals of Scientific Computing.*
9. M. Bader, *Space-Filling Curves.*
10. M. Larson, F. Bengzon, *The Finite Element Method: Theory, Implementation and Applications.*
11. W. Gander, M. Gander, F. Kwok, *Scientific Computing: An Introduction using Maple and MATLAB.*

For further information on these books please have a look at our mathematics catalogue at the following URL: www.springer.com/series/5151