

# Mortar Methods for Single- and Multi-Field Applications in Computational Mechanics

Alexander Popp, Michael W. Gee, and Wolfgang A. Wall

**Abstract** Mortar finite element methods are of great relevance as a non-conforming discretization technique in various single-field and multi-field applications. In computational contact analysis, the mortar approach allows for a variationally consistent treatment of non-penetration and frictional sliding constraints despite the inevitably non-matching interface meshes. Other single-field and multi-field problems, such as fluid–structure interaction (FSI), also benefit from the increased modeling flexibility provided by mortar methods. This contribution gives a review of the most important aspects of mortar finite element discretization and dual Lagrange multiplier interpolation for the aforementioned applications. The focus is on parallel efficiency, which is addressed by a new dynamic load balancing strategy and tailored parallel search algorithms for computational contact mechanics. For validation purposes, simulation examples from solid dynamics, contact dynamics and FSI will be discussed.

## 1 Introduction

Mortar finite element methods, and particularly their application to computational contact mechanics and several other single-field and multi-field problems, have seen a great thrust of research over the last decade. In this contribution, we give a review of the most important features of non-conforming finite element discretization

---

A. Popp (✉) · W.A. Wall  
Institute for Computational Mechanics, Technische Universität München, Boltzmannstr. 15,  
85747 Garching, Germany  
e-mail: [popp@lnm.mw.tum.de](mailto:popp@lnm.mw.tum.de); [wall@lnm.mw.tum.de](mailto:wall@lnm.mw.tum.de)

M.W. Gee  
Mechanics and High Performance Computing Group, Technische Universität München,  
Boltzmannstr. 15, 85747 Garching, Germany  
e-mail: [gee@tum.de](mailto:gee@tum.de)

based on mortar methods with a special emphasis on dual Lagrange multiplier interpolation, parallel efficiency of the devised numerical algorithms and high performance computing. Thus, the present work combines and extends different aspects of our previous work in [5, 8–11], to which we refer for technical details, more profound discussion of the presented methods and further numerical examples.

Originally introduced as a domain decomposition technique for spectral elements in [2], mortar methods are nowadays also widely used within finite element formulations for many different problem classes. The first investigations on mortar finite element methods were typically performed for model problems of Laplace operator type, e.g., the Poisson equation, and formulated as non-conforming variational problem with the coupling constraints directly introduced into the global solution space. Yet, an alternative formulation soon became popular, which is not based on a constrained solution space, but rather introduces Lagrange multipliers in the sense of constrained minimization, thus leading to a typical saddle point formulation. Details of both approaches can, for example, be found in [1, 18, 19].

The mortar approach is characterized by an imposition of the occurrent interface constraints in a weak sense and by the possibility to prove its mathematical optimality. This means that suitable inf-sup conditions and *a priori* error estimates for the consistency error and the best approximation error have been established for the most widely used finite element discretizations and for different choices of the discrete Lagrange multiplier space. If considering first-order finite elements as an example, their optimal spatial convergence of order  $\mathcal{O}(h^2)$  measured in the  $L^2$ -norm is preserved by mortar methods, despite the fact that non-conforming interfaces are involved. Establishing optimal *a priori* error bounds for unilateral contact problems is more intricate due to the typically reduced regularity of the solution. The interested reader is exemplarily referred to [20] and the references therein.

Especially the choice of the discrete Lagrange multiplier space is an essential question with great implications on the actual algorithmic realization of mortar methods. Standard mortar methods suffer from a serious drawback: they generate high computational costs due to the global character of the resulting interface coupling conditions, see [19] for a detailed explanation. However, this issue can be completely resolved by introducing so-called *dual* Lagrange multiplier spaces as proposed in [18], which are constructed based on a biorthogonality condition such that the interface coupling conditions reduce to purely local constraints.

This approach has been applied very successfully to impose interface constraints for finite deformation contact analysis in [10–12, 20], thus allowing for a variationally consistent treatment of non-penetration and frictional sliding conditions despite the inevitably non-matching interface meshes for finite deformations and large sliding motions. Recently, the focus of research in the field of mortar finite element methods and dual Lagrange multiplier interpolation has been extended towards other single-field applications and especially to coupled multiphysics problems. A computational framework for fluid–structure interaction has been proposed in [8] and the combination of FSI and contact interaction for capturing physical phenomena such as elasto-hydrodynamic lubrication is discussed in [9, 16, 24]. The

coupling of several subdomains with non-matching meshes in computational fluid dynamics can also be efficiently carried out with dual mortar methods, see [4].

The remainder of this contribution is organized as follows: Sect. 2 provides an introduction to mortar finite element methods in the exemplary context of mesh tying for nonlinear solid mechanics. The main part in Sect. 3 then describes several aspects of the efficient parallel implementation of mortar methods within a high performance computing framework. Concretely, dynamic load balancing techniques as well as search algorithms for computational contact mechanics will be addressed. With these general findings at hand, different application scenarios ranging from classical mesh tying in solid mechanics to finite deformation contact and coupled problems such as FSI including contact are highlighted in Sect. 4. Finally, Sect. 5 gives a summary and outlook on current and future research directions.

## 2 Mortar Finite Element Methods

Mesh tying in solid mechanics (also referred to as tied contact) serves as a model problem for the introduction to mortar finite element methods here. The motivation for such mortar mesh tying algorithms is to connect dissimilar meshes in nonlinear solid mechanics in a variationally consistent manner, see also [13] for a comprehensive overview. Reasons for the occurrence of non-matching meshes can be manifold and range from different resolution requirements in the individual subdomains to the use of different types of finite element interpolations.

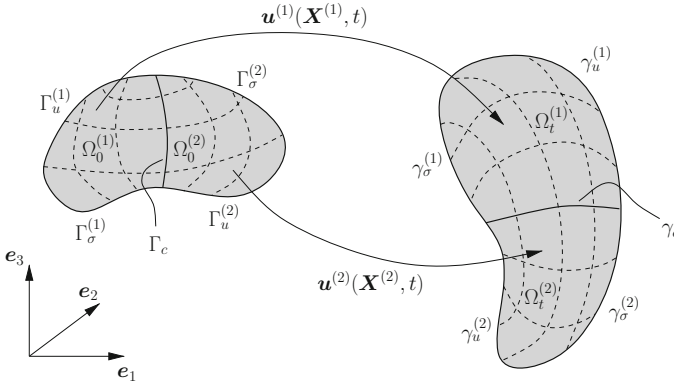
Without losing generality, only the case of a body with one sole tied contact interface is considered. A generalization to the case of multiple interfaces is however possible without conceptual differences. Figure 1 gives an overview of the general problem setup and introduces some basic notation. The open sets  $\Omega_0^{(i)} \subset \mathbb{R}^3$  and  $\Omega_t^{(i)} \subset \mathbb{R}^3$ ,  $i = 1, 2$ , represent the two subdomains of the contemplated body in the reference and current configuration, respectively. The surfaces  $\partial\Omega_0^{(i)}$  are divided into three disjoint subsets  $\Gamma_u^{(i)}$ ,  $\Gamma_\sigma^{(i)}$  and  $\Gamma_c^{(i)}$ , where  $\Gamma_u^{(i)}$  is the Dirichlet boundary,  $\Gamma_\sigma^{(i)}$  is the Neumann boundary, and  $\Gamma_c^{(i)}$  represents the mesh tying interface. Any gaps and overlaps between the subdomains are excluded, i.e.,  $\Gamma_c^{(1)} \equiv \Gamma_c^{(2)} \equiv \Gamma_c$ . The superscript (1) is commonly referred to as slave side of the problem, whereas the superscript (2) denotes the master side.

On each subdomain  $\Omega_0^{(i)}$ , the initial boundary value problem (IBVP) of finite deformation elastodynamics needs to be satisfied, i.e.,

$$\text{Div} \mathbf{P}^{(i)} + \hat{\mathbf{b}}_0^{(i)} = \varrho_0^{(i)} \ddot{\mathbf{u}}^{(i)} \quad \text{in } \Omega_0^{(i)} \times [0, T], \quad (1)$$

$$\mathbf{u}^{(i)} = \hat{\mathbf{u}}^{(i)} \quad \text{on } \Gamma_u^{(i)} \times [0, T], \quad (2)$$

$$\mathbf{P}^{(i)} \mathbf{N}^{(i)} = \hat{\mathbf{t}}_0^{(i)} \quad \text{on } \Gamma_\sigma^{(i)} \times [0, T], \quad (3)$$



**Fig. 1** Configurations, kinematics and basic notation for a mesh tying problem in 3D

$$\mathbf{u}^{(i)}(\mathbf{X}^{(i)}, 0) = \hat{\mathbf{u}}_0^{(i)}(\mathbf{X}^{(i)}) \quad \text{in } \Omega_0^{(i)}, \quad (4)$$

$$\dot{\mathbf{u}}^{(i)}(\mathbf{X}^{(i)}, 0) = \hat{\dot{\mathbf{u}}}_0^{(i)}(\mathbf{X}^{(i)}) \quad \text{in } \Omega_0^{(i)}. \quad (5)$$

The mesh tying constraint formulated in the reference configuration is given as

$$\mathbf{u}^{(1)} = \mathbf{u}^{(2)} \quad \text{on } \Gamma_c \times [0, T]. \quad (6)$$

Equations (1)–(6) represent the final strong form of a mesh tying problem in nonlinear solid mechanics. In the course of deriving a weak formulations, the balance of linear momentum at the mesh tying interface  $\Gamma_c$  is typically exploited and a Lagrange multiplier vector  $\boldsymbol{\lambda}$  is introduced, thus setting the basis for a mixed variational approach. To start the derivation of a weak formulation of (1)–(6), appropriate solution spaces  $\mathcal{U}^{(i)}$  and weighting spaces  $\mathcal{V}^{(i)}$  need to be defined as

$$\mathcal{U}^{(i)} = \left\{ \mathbf{u}^{(i)} \in H^1(\Omega) \mid \mathbf{u}^{(i)} = \hat{\mathbf{u}}^{(i)} \text{ on } \Gamma_u \right\}, \quad (7)$$

$$\mathcal{V}^{(i)} = \left\{ \delta \mathbf{u}^{(i)} \in H^1(\Omega) \mid \delta \mathbf{u}^{(i)} = \mathbf{0} \text{ on } \Gamma_u \right\}. \quad (8)$$

Moreover, the Lagrange multiplier vector  $\boldsymbol{\lambda} = -\mathbf{t}_c^{(1)}$ , which represents the *negative* slave side contact traction  $\mathbf{t}_c^{(1)}$ , is chosen from a corresponding solution space denoted as  $\mathcal{M}$ . In terms of its classification in functional analysis,  $\mathcal{M}$  represents the dual space of the trace space  $\mathcal{W}^{(1)}$  of  $\mathcal{V}^{(1)}$ . In the given context, this means that  $\mathcal{M} = H^{-\frac{1}{2}}(\Gamma_c)$  and  $\mathcal{W}^{(1)} = H^{\frac{1}{2}}(\Gamma_c)$ . Based on these considerations, the saddle point type weak formulation is derived next. Basically, this can be done by extending the standard weak formulation of solid mechanics to two subdomains and combining it with Lagrange multiplier coupling terms. Find  $\mathbf{u}^{(i)} \in \mathcal{U}^{(i)}$  and  $\boldsymbol{\lambda} \in \mathcal{M}$  such that

$$\delta\mathcal{W}_{kin}(\dot{\mathbf{u}}^{(i)}, \delta\mathbf{u}^{(i)}) + \delta\mathcal{W}_{int,ext}(\mathbf{u}^{(i)}, \delta\mathbf{u}^{(i)}) + \delta\mathcal{W}_{mt}(\boldsymbol{\lambda}, \delta\mathbf{u}^{(i)}) = 0 \quad \forall \delta\mathbf{u}^{(i)} \in \mathcal{V}^{(i)}, \quad (9)$$

$$\delta\mathcal{W}_{\lambda}(\mathbf{u}^{(i)}, \delta\boldsymbol{\lambda}) = 0 \quad \forall \delta\boldsymbol{\lambda} \in \mathcal{M}. \quad (10)$$

Herein, the kinetic contribution  $\delta\mathcal{W}_{kin}$  as well as the internal and external contributions  $\delta\mathcal{W}_{int,ext}$  are well-known from standard weak formulations in nonlinear solid mechanics. The mesh tying interface contribution  $\delta\mathcal{W}_{mt}$  and the weak mesh tying constraint  $\delta\mathcal{W}_{\lambda}$  have been abbreviated as

$$\delta\mathcal{W}_{mt} = \int_{\Gamma_c} \boldsymbol{\lambda} (\delta\mathbf{u}^{(1)} - \delta\mathbf{u}^{(2)}) \, dA_0, \quad \delta\mathcal{W}_{\lambda} = \int_{\Gamma_c} \delta\boldsymbol{\lambda} (\mathbf{u}^{(1)} - \mathbf{u}^{(2)}) \, dA_0. \quad (11)$$

For the spatial discretization of the tied contact problem (9)–(10), standard isoparametric finite elements are employed. This defines the usual finite dimensional subspaces  $\mathcal{Q}_h^{(i)}$  and  $\mathcal{V}_h^{(i)}$  being approximations of  $\mathcal{Q}^{(i)}$  and  $\mathcal{V}^{(i)}$ , respectively. Within our implementation, both first-order and second-order interpolation in 2D and 3D are considered. The subscript  $h$  refers to a spatially discretized quantity. Obviously, there exists a direct connection between the employed finite elements in the domains and the resulting surface facets on the mortar interfaces  $\Gamma_{c,h}^{(i)}$ . For example, a 3D finite element mesh composed of *tet4* and *hex8* elements yields *tri3* and *quad4* facets on the surface of tied contact. Consequently, the following general form of displacement interpolation on the discrete mesh tying surfaces holds:

$$\mathbf{u}_h^{(1)}|_{\Gamma_{c,h}^{(1)}} = \sum_{k=1}^{n^{(1)}} N_k^{(1)}(\xi^{(1)}, \eta^{(1)}) \mathbf{d}_k^{(1)}, \quad \mathbf{u}_h^{(2)}|_{\Gamma_{c,h}^{(2)}} = \sum_{l=1}^{n^{(2)}} N_l^{(2)}(\xi^{(2)}, \eta^{(2)}) \mathbf{d}_l^{(2)}. \quad (12)$$

The total number of slave nodes on  $\Gamma_{c,h}^{(1)}$  is  $n^{(1)}$ , and the total number of master nodes on  $\Gamma_{c,h}^{(2)}$  is  $n^{(2)}$ . Discrete nodal displacements are given by  $\mathbf{d}_k^{(1)}$  and  $\mathbf{d}_l^{(2)}$ . The shape functions  $N_k^{(1)}$  and  $N_l^{(2)}$  are defined with respect to the usual finite element parameter space, usually denoted as  $\xi^{(i)}$  for two-dimensional problems (i.e., 1D mesh tying interfaces) and as  $\boldsymbol{\xi}^{(i)} = (\xi^{(i)}, \eta^{(i)})$  for three-dimensional problems (i.e., 2D mesh tying interfaces). In addition, an adequate discretization of the Lagrange multiplier vector  $\boldsymbol{\lambda}$  is needed, too, and will be based on a discrete Lagrange multiplier space  $\mathcal{M}_h$  being an approximation of  $\mathcal{M}$ . A general notation reads:

$$\boldsymbol{\lambda}_h = \sum_{j=1}^{m^{(1)}} \Phi_j(\xi^{(1)}, \eta^{(1)}) \boldsymbol{\lambda}_j, \quad (13)$$

with the (still to be defined) shape functions  $\Phi_j$  and the discrete nodal Lagrange multipliers  $\boldsymbol{\lambda}_j$ . The total number of slave nodes carrying additional Lagrange multiplier degrees of freedom is  $m^{(1)}$ . Typically for mortar methods, every slave node also serves as coupling node, and thus in the majority of cases  $m^{(1)} = n^{(1)}$  will

hold. However, in the context of second-order finite element interpolation it may be favorable to chose  $m^{(1)} < n^{(1)}$  in certain cases, see e.g., [12, 20].

Two different families of discrete Lagrange multipliers, namely *standard* and so-called *dual* Lagrange multipliers are commonly distinguished. Standard Lagrange multipliers represent the classical approach for mortar methods (cf. [1, 14]) and lead to identical shape functions for Lagrange multiplier and slave displacement interpolation, i.e.,  $\Phi_j = N_j^{(1)}$ . In contrast, the dual approach is motivated by the observation that the Lagrange multipliers physically represent fluxes (tractions) on the mesh tying interface in the continuous setting. This duality argument is then reflected by constructing dual Lagrange multiplier shape functions based on a so-called biorthogonality condition with the displacements in  $\mathscr{W}_h^{(1)}$ , see e.g., [18]. While they are in general not continuous and cannot be interpreted as a trace of conforming finite elements, the biorthogonality condition assures that the Lagrange multiplier shape functions  $\Phi_j$  are again well-defined and satisfy all required approximation properties. One crucial advantage of the dual approach lies in the fact that it heavily facilitates the treatment of typical mortar coupling conditions at the interface, while at the same time preserving the optimality of the method.

Substituting (12) and (13) into the interface virtual work  $\delta\mathscr{W}_{mt}$  in (9) yields

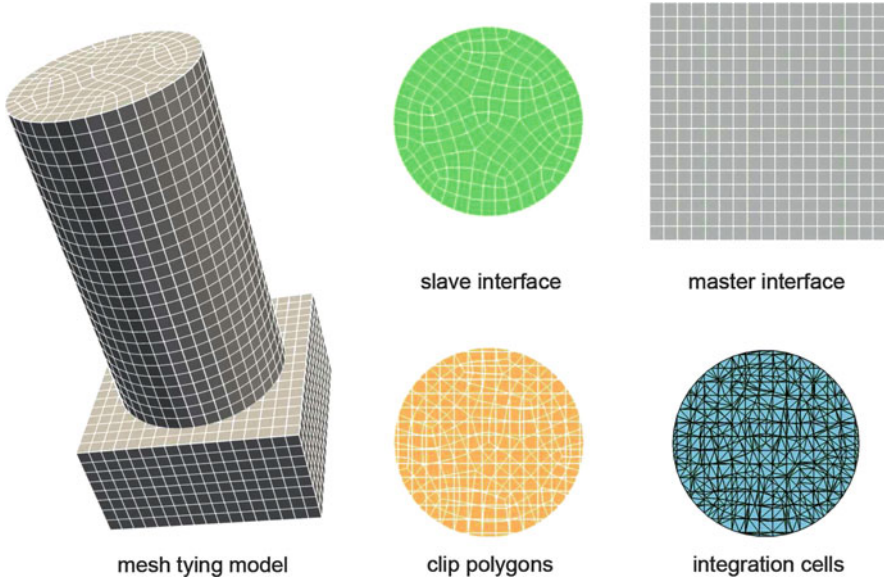
$$\begin{aligned} \delta\mathscr{W}_{mt,h} = & \sum_{j=1}^{m^{(1)}} \sum_{k=1}^{n^{(1)}} \lambda_j^\top \left( \int_{\Gamma_{c,h}^{(1)}} \Phi_j N_k^{(1)} dA_0 \right) \delta \mathbf{d}_k^{(1)} \\ & - \sum_{j=1}^{m^{(1)}} \sum_{l=1}^{n^{(2)}} \lambda_j^\top \left( \int_{\Gamma_{c,h}^{(1)}} \Phi_j (N_l^{(2)} \circ \chi_h) dA_0 \right) \delta \mathbf{d}_l^{(2)}, \end{aligned} \quad (14)$$

where  $\chi_h : \Gamma_{c,h}^{(1)} \rightarrow \Gamma_{c,h}^{(2)}$  defines a suitable discrete mapping from slave to master side of the mesh tying interface. Such a mapping (or projection) becomes necessary due to the fact that the discretized coupling surfaces  $\Gamma_{c,h}^{(1)}$  and  $\Gamma_{c,h}^{(2)}$  are, in general, no longer geometrically coincident. In (14), nodal blocks of the two mortar integral matrices commonly denoted as  $\mathbf{D}$  and  $\mathbf{M}$  can be identified. This leads to the following definitions:

$$\mathbf{D}[j, k] = \int_{\Gamma_{c,h}^{(1)}} \Phi_j N_k^{(1)} dA_0 \mathbf{I}_{ndim}, \quad j = 1, \dots, m^{(1)}, \quad k = 1, \dots, n^{(1)}, \quad (15)$$

$$\mathbf{M}[j, l] = \int_{\Gamma_{c,h}^{(1)}} \Phi_j (N_l^{(2)} \circ \chi_h) dA_0 \mathbf{I}_{ndim}, \quad j = 1, \dots, m^{(1)}, \quad l = 1, \dots, n^{(2)}. \quad (16)$$

Note that  $\mathbf{I}_{ndim} \in \mathbb{R}^{ndim \times ndim}$  is an identity matrix whose size is determined by the global problem dimension  $ndim$ , i.e., either  $ndim = 2$  or  $ndim = 3$ . In general, both mortar matrices  $\mathbf{D}$  and  $\mathbf{M}$  have a rectangular shape, however  $\mathbf{D}$  becomes a square matrix for the common choice  $m^{(1)} = n^{(1)}$ . Dual Lagrange



**Fig. 2** Main steps of 3D mortar coupling for a mesh tying example—projection and intersection of slave and master surfaces, determination of clip polygons and triangulation into so-called integration cells, see [11, 13] for details

multiplier interpolation based on a biorthogonality condition allows for the beneficial simplification of  $\mathbf{D}$  reducing to diagonal form. All details concerning the actual numerical integration of the mass matrix type of entries in  $\mathbf{D}$  and  $\mathbf{M}$  as well as the implementation of the interface mapping  $\chi_h$  for both 2D and 3D can be found in our recent work [10, 11]. Here, Fig. 2 only illustrates the generation of integration cells for 3D mortar coupling with an exemplary mesh tying model.

For the ease of notation, all nodes of the two subdomains  $\Omega_0^{(1)}$  and  $\Omega_0^{(2)}$ , and correspondingly all degrees of freedom (DOFs) in the global discrete displacement vector  $\mathbf{d}$ , are sorted into three groups: a group  $\mathcal{S}$  containing all slave interface quantities, a group  $\mathcal{M}$  of all master quantities and a group denoted as  $\mathcal{N}$ , which comprises all remaining nodes or DOFs. The global discrete displacement vector can be sorted accordingly, yielding  $\mathbf{d} = (\mathbf{d}_{\mathcal{N}}, \mathbf{d}_{\mathcal{M}}, \mathbf{d}_{\mathcal{S}})$ . Going back to (14), this allows for the following definition:

$$\delta \mathcal{W}_{mt,h} = \delta \mathbf{d}_{\mathcal{S}}^{\top} \mathbf{D}^{\top} \boldsymbol{\lambda} - \delta \mathbf{d}_{\mathcal{M}}^{\top} \mathbf{M}^{\top} \boldsymbol{\lambda} = \delta \mathbf{d}^{\top} \mathbf{B}_{mt}^{\top} \boldsymbol{\lambda} = \delta \mathbf{d}^{\top} \mathbf{f}_{mt}(\boldsymbol{\lambda}). \quad (17)$$

Herein, the discrete mortar mesh tying operator  $\mathbf{B}_{mt}$  and the resulting discrete vector of mesh tying forces  $\mathbf{f}_{mt}(\boldsymbol{\lambda}) = \mathbf{B}_{mt}^{\top} \boldsymbol{\lambda}$  acting on slave and master side of the interface are introduced. Due to the saddle point characteristics and resulting symmetry of the mixed variational formulation (9)–(10), the final formulation of the weak constraint contribution  $\delta \mathcal{W}_{\lambda}$  can directly be given as

$$\delta \mathcal{W}_{\lambda,h} = \delta \boldsymbol{\lambda}^{\top} \mathbf{D} \mathbf{d}_{\mathcal{S}} - \delta \boldsymbol{\lambda}^{\top} \mathbf{M} \mathbf{d}_{\mathcal{M}} = \delta \boldsymbol{\lambda}^{\top} \mathbf{B}_{mt} \mathbf{d} = \delta \boldsymbol{\lambda}^{\top} \mathbf{g}_{mt}(\mathbf{d}), \quad (18)$$

with  $\mathbf{g}_{mt}(\mathbf{d}) = \mathbf{B}_{mt}\mathbf{d}$  representing the discrete mesh tying constraint at the coupling interface. Taking into account the finite element discretization of all remaining contributions to the first part of the weak formulation (9), the semi-discrete equations of motion including tied contact forces and the constraint equations emerge as

$$\mathbf{M}\ddot{\mathbf{d}} + \mathbf{C}\dot{\mathbf{d}} + \mathbf{f}_{int}(\mathbf{d}) + \mathbf{f}_{mt}(\boldsymbol{\lambda}) - \mathbf{f}_{ext} = \mathbf{0}, \quad (19)$$

$$\mathbf{g}_{mt}(\mathbf{d}) = \mathbf{0}. \quad (20)$$

Mass matrix  $\mathbf{M}$ , damping matrix  $\mathbf{C}$ , internal forces  $\mathbf{f}_{int}(\mathbf{d})$  and external forces  $\mathbf{f}_{ext}$  result from standard FE discretization. It is important to point out, that the actual mortar based interface coupling described here is completely independent of the concrete choice of an underlying finite element formulation.

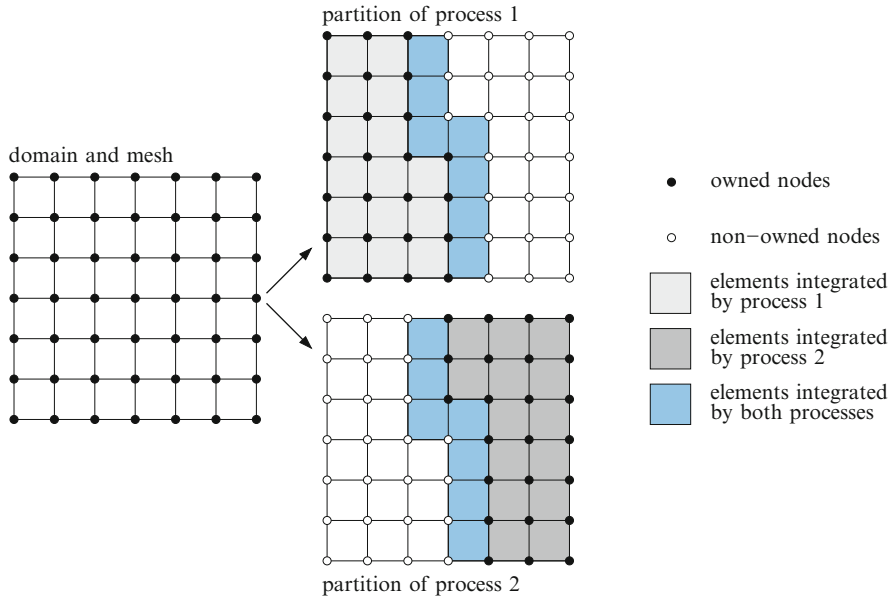
### 3 Aspects of Implementation and High Performance Computing

The following section aims at addressing and outlining some of the most important implementation and software design issues associated with the proposed mortar finite element methods. We first put an emphasis on inter-processor redistribution and dynamic load balancing strategies for mortar methods, which in turn requires a short introduction to the employed paradigm of parallel programming. Furthermore, the basic concepts of efficient parallel search algorithms for two body contact, self contact and multiple bodies (e.g., agglomerations of elastic particles) will be presented. All explanations exclusively refer to implementations devised in the context of the present contribution, and subsequently integrated into the in-house finite element software package BACI (cf. [17]), developed at the Institute for Computational Mechanics at Technische Universität München.

#### 3.1 Parallel Redistribution and Dynamic Load Balancing

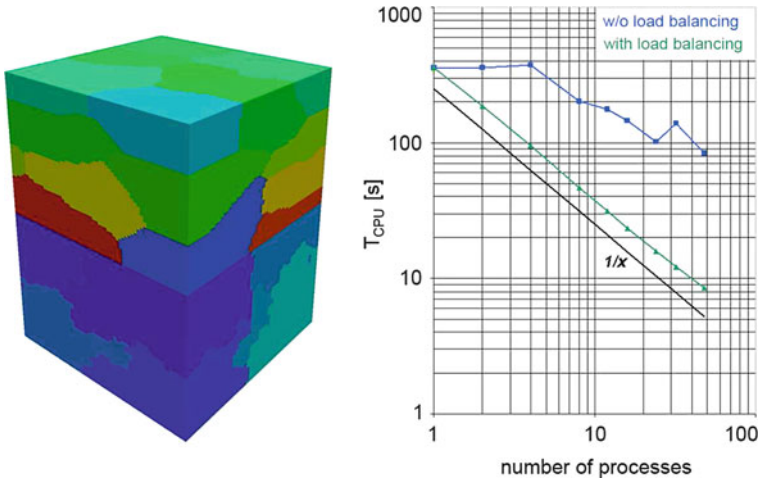
The presented mortar based mesh tying and contact algorithms are designed for the use on large interconnected computer systems (clusters) with many CPUs and a distributed main memory. Being able to efficiently run large simulations in parallel requires strategies for the partitioning and parallel distribution of the problem data, i.e., finite element meshes (consisting of nodes and elements) as well as global vectors and matrices, into several independent processes. Within BACI, this so-called domain (or data) decomposition is provided by the third-party library ParMETIS, see e.g., [7], and all communication tasks are implemented through MPI.





**Fig. 3** An example of overlapping domain decomposition and parallel assembly involving two independent processes

An example of such decompositions is visualized in Fig. 3 for a simple partitioning including only two processes. It can be seen that each node in the mesh is uniquely assigned to one specific process, and the same holds true for the elements. In addition, some nodes and elements at the transition between different processes must be stored redundantly within all adjacent processes. Therefore, this type of partitioning is commonly denoted as *overlapping* decomposition. Here, it is sufficient to consider only the most straightforward case of minimal overlap between the individual partitions, i.e., an overlap of one layer of elements or nodes, respectively. Obviously, this concept of overlapping decomposition fits quite naturally to the typical tasks within a finite element program: first, each process performs an elementwise integration of its own partition of the computational domain including the (relatively few) elements at the inter-process boundaries. Then, the resulting quantities (e.g., local element load vectors and stiffness matrices) are assembled into the respective FE nodes of each process. Thus, overlapping domain decomposition as described above provides a very elegant way of processing finite element integration and assembly, which is completely free of communication due to the distributed storage of the resulting global vector and matrix objects. For further details on the C++ based implementation of parallel (i.e., distributed) matrix and vector objects as well as the associated linear algebra, the interested reader is exemplarily referred to the extensive documentation of open-source libraries of the Trilinos Project conducted by Sandia National Laboratories (cf. [6]).



**Fig. 4** Parallel redistribution and dynamic load balancing—initial partitioning for exemplary mesh tying problem setup using 32 processes (*left*) and strong scaling diagram (*right*)

Returning to the efficient parallel treatment of mortar methods and the derived mesh tying and contact algorithms, we now examine an exemplary mesh tying problem setup consisting of two cubic bodies, which are distributed in parallel among several processes as depicted in Fig. 4. This partitioning, generated via the ParMETIS library, is in a sense optimal for the integration and assembly of the individual volume finite elements in the two bodies, i.e., the corresponding workload is equally distributed among all processes. For mortar coupling, however, additional (but conceptually similar) tasks have to be performed locally at the mesh tying interface. Computing the mortar contributions to the overall discrete problem formulation involves numerical integration and assembly of the mortar matrices  $\mathbf{D}$  and  $\mathbf{M}$ , to name only the most important task. Unfortunately, due to its locality, the parallel distribution of the mortar interface itself is not optimal at all.

Figure 4 also illustrates typical results for the parallel efficiency of the presented mortar algorithms in a so-called *strong scaling* diagram. Therein, the CPU time for numerical integration and assembly of all interface-related quantities  $T_{CPU}$  is plotted against the total number of processes  $N$  with logarithmic scales applied to both axes. Perfect scalability of the examined numerical algorithm is represented by a straight line with a negative slope of  $-1$ , thus representing the evident relation that  $T_{CPU} \sim 1/N$ . It can clearly be seen from Fig. 4 that initially no perfect scalability is achieved with the presented algorithms. This is due to the non-optimal distribution of the slave surface among the participating processes as already described above. The results clearly motivate the need to develop an efficient parallel redistribution and load balancing strategy for mortar finite element methods. The approach proposed in the following is based on three steps, where the first step is

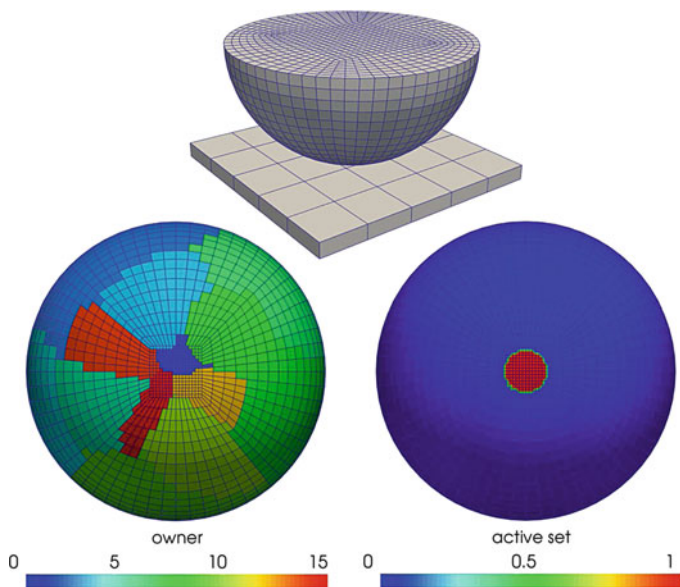
of fundamental importance and is therefore needed for both mesh tying and contact applications. In contrast, the second and third step are purely contact-specific.

The rather simple basic idea of the first step is an *independent* parallel distribution of the finite elements in the domain and the mortar elements at the coupling interface in order to achieve optimal parallel scalability of the computational tasks associated with both, i.e., FE integration and assembly in  $\Omega^{(1)}$  and  $\Omega^{(2)}$  as well as mortar integration and assembly on  $\Gamma_c^{(1)}$  and  $\Gamma_c^{(2)}$ . Again using ParMETIS, this redistribution of the interface elements can readily be performed during problem initialization at  $t = 0$ . Results for the test model introduced above are also visualized in Fig. 4, thus demonstrating that this simple modification already allows for perfect parallel scalability within a wide range concerning the number of processes. However, dependent on the considered problem size, parallel redistribution only makes sense up to a certain number of processes. It is quite natural that such a limit exists, because there are of course some computational costs associated with the proposed redistribution procedure itself. If too many processes are used in relation to the problem size, these costs (mainly due to communication) become dominant and parallel redistribution is no longer profitable beyond this point.

As already mentioned, this strategy can be further refined for contact applications. In contrast to mesh tying, contact interfaces are characterized by two additional complexities: the actual contact zone is not known *a priori* and it may constantly and significantly vary over time. Thus, in a second and third step, the proposed redistribution strategy is adapted such that it accommodates these additional complexities. Concretely, it can be seen from Fig. 5 that parallel redistribution must be limited to the actual contact area instead of the potential contact area, because the entire computational effort of numerical integration and assembly is connected with the former. Moreover, whenever finite deformations and large sliding motions occur, the redistribution needs to be performed dynamically, i.e., over and over again. Such a dynamic load balancing strategy is then typically triggered by a suitable measure for the workload of each individual process. The parallel balance of the workload among all processes is monitored and a simple criterion whether to apply dynamic load balancing within the current time step or not can be formulated as

$$IF \left( \frac{T_{CPU}^{max}}{T_{CPU}^{min}} > \varepsilon \right) \rightsquigarrow \text{redistribute.} \quad (21)$$

Herein, the minimum and maximum CPU times of one individual process in the last time step are denoted as  $T_{CPU}^{min}$  and  $T_{CPU}^{max}$ , respectively. The parameter  $\varepsilon > 1$  represents a user-defined tolerance. For example, choosing  $\varepsilon = 1.2$  implies that at most 20 % unbalance of the parallel workload distribution are tolerated. Of course, the rather simple condition in (21) can easily be extended to incorporate more sophisticated criteria for dynamic load balancing.



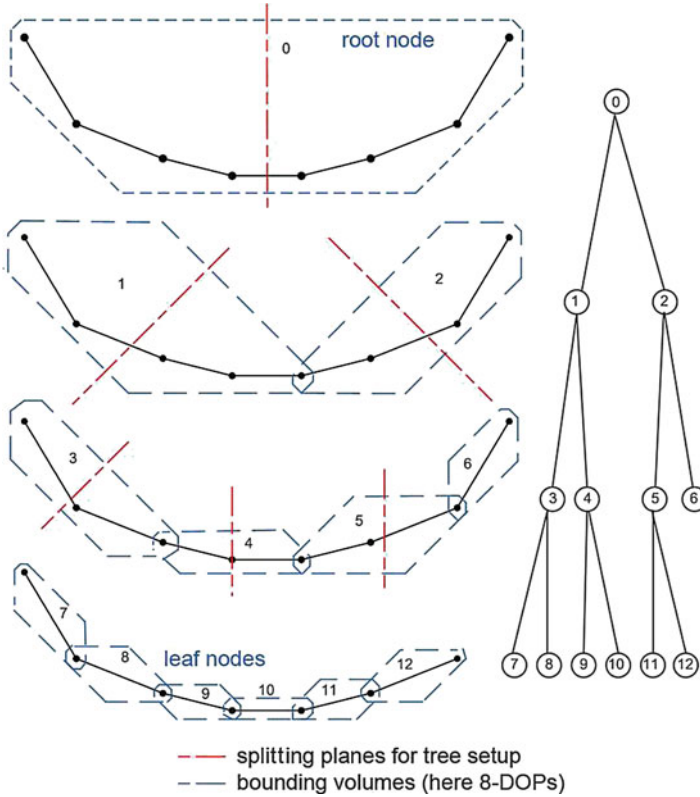
**Fig. 5** Motivation for parallel redistribution exemplified with a Hertzian contact example—the active contact region (*bottom right*) is relatively small as compared with the potential contact surface, i.e., the whole hemisphere. Without redistribution only six out of 16 processes would carry the entire workload associated with contact evaluation (*bottom left*)

### 3.2 Search Algorithms for Two-Body Contact and Self Contact

The search for bodies or individual finite elements that might possibly come into contact is an important algorithmic aspect of any FEM contact formulation. In particular, this is true in the context of finite deformations and large sliding motions as primarily considered throughout the present work, because the contact situation continuously changes in such scenarios. Search algorithms have been a subject of intensive research since the beginnings of the computational treatment of contact mechanics problems, see e.g., [21] for a comprehensive overview on the topic.

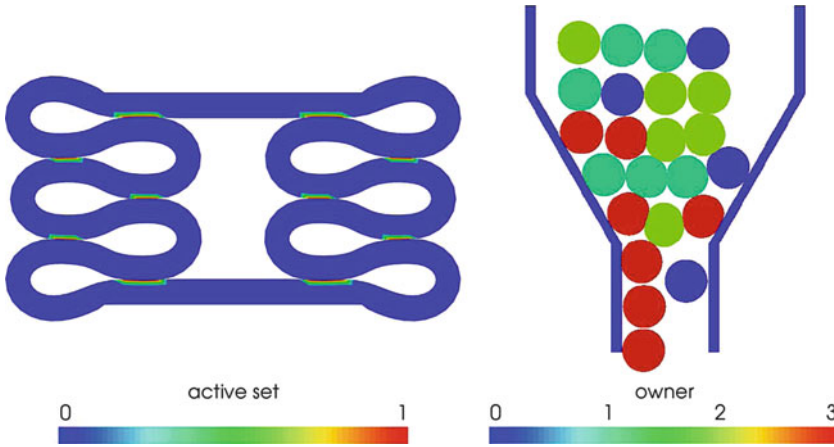
The basic motivation for efficient contact search algorithms can be easily understood. A naive search approach for two-body contact would require to check *all* finite elements on the slave side against *all* finite elements on the master side for proximity. Thus, the associated number of operations would be  $N \times M$ , where  $N$  and  $M$  are the total numbers of slave and master elements, respectively. Assuming  $M \approx N$  for the sake of simplicity, the resulting computational complexity of such so-called brute force search algorithms is  $\mathcal{O}(N^2)$ . Clearly, this makes contact search inacceptably slow already for rather moderate problem sizes.

In the following, a short overview of the parallel search algorithm for two-body contact used in this contribution will be given, which is closely related to the work in [22]. Basically, most contact search algorithms consist of two components, i.e., a



**Fig. 6** Search algorithm for two-body contact—two-dimensional example based on 8-DOPs as bounding volumes and a hierarchical binary tree structure

hierarchical global search structure (so-called search tree) and an efficient local geometry representation (so-called bounding volumes). Here, discretized orientation polytopes with  $k$  edges ( $k$ -DOPs) serve as bounding volumes. Compared to the commonly employed axis-aligned bounding boxes (AABBs), the  $k$ -DOPs allow for a much tighter and thus more efficient geometrical representation of the contact surfaces. For 2D simulations, the bounding volumes are typically 8-DOPs, while 18-DOPs are employed in the 3D case. Figure 6 provides a schematic illustration of these ideas for a two-dimensional setting. Further details and comprehensive illustrations can be found in [22]. As can also be seen from Fig. 6, both slave and master surface are then organized and stored within hierarchical binary tree structures, which allow for very fast search and update procedures. The search tree is typically only built once during problem initialization in a top-down way. This process starts from a so-called root node, which contains the entire slave or master contact surface, and then the considered surfaces are continuously divided in halves until arriving at the individual finite elements (so-called leaf nodes of the



**Fig. 7** Examples for self contact and contact of multiple bodies. The active set is visualized for self contact (*left*) and the parallel distribution is shown for the multiple body case (*right*)

search tree). An update of the tree, i.e., of the contact geometry, must be done after each nonlinear iteration step due to the fact finite deformations are considered here.

The search procedure itself basically consists of a recursive algorithm starting with an intersection test of slave and master root nodes. Wherever necessary, i.e., wherever an overlap of the corresponding bounding volumes is detected, the search algorithm proceeds into the lower tree levels until the leaf level is reached. A theoretical analysis in [22] predicts the resulting algorithm complexity to be  $\mathcal{O}(N \cdot \log(N^2))$ , which has also been confirmed in various numerical investigations. Going beyond the implementation in [22], which is limited to the single-processor case, the presented search algorithm has been extended to fit into a parallel FE simulation framework. As explained above, slave and master surface are then distributed among several independent processes and the tree update as well as search procedures are only performed on the part of the slave contact surface that is actually part of the problem partition of the respective process. This generates a distributed search algorithm with optimal parallel scalability, where only the geometry of the (likewise distributed) master surface needs to be communicated among all processes in order to detect all possible contact pairs.

For the sake of completeness, it is mentioned that two special problem classes in computational contact mechanics, namely self contact and contact of multiple bodies, can be treated within the same algorithmic framework as described above. Two characteristic examples for the mentioned problem classes are illustrated in Fig. 7. The search algorithm for self contact and multiple bodies employed here is again closely related to the work given in [22, 23]. The only algorithmic difference to the two-body case is the way in which the contact search is realized. As can be seen in Fig. 7, self contact is characterized by only one *single* potential self contact surface, so that no a priori definition of slave and master surfaces is possible, but this

assignment rather needs to be done in a dynamic manner. Thus, the search procedure needs to be adapted in order to accommodate possible self contact. Moreover, Fig. 7 indicates that the case of multiple bodies lies somewhere in between two-body contact and self contact with regard to its numerical treatment. Again, it is not possible to find a unique a priori definition of slave and master surfaces. However, once slave and master pairs are (dynamically) assigned, this scenario can basically be interpreted and treated as a great many of simple two-body contacts.

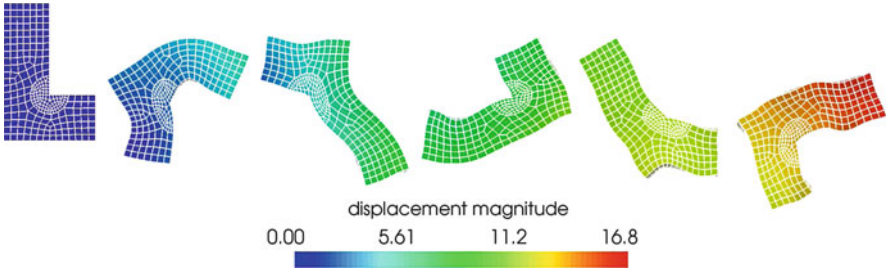
## 4 Exemplary Single-Field and Multi-Field Applications

We present several numerical examples to illustrate the capabilities of the proposed mortar finite element approach. All simulations are based on a parallel implementation of the mesh tying and contact algorithms described above in our in-house research code BACI [17]. The chosen set of examples demonstrates the versatility of mortar finite element methods as non-conforming discretization approach for a manifold of applications in computational mechanics, ranging from single-field problems in solid and fluid mechanics to challenging multi-field problems.

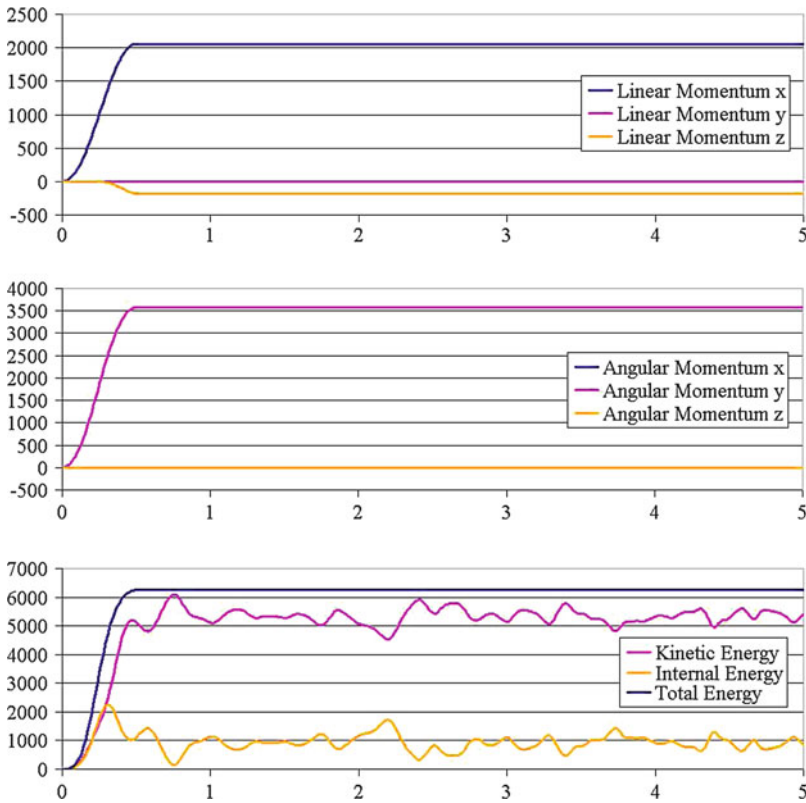
### 4.1 Mesh Tying in Solid Mechanics

The first numerical example investigates mortar finite element algorithms for 3D mesh tying in the most general context of transient solid dynamics with finite deformations and nonlinear material behavior. As illustrated in Fig. 8, the model consists of an L-shaped block, whose larger part has the dimensions  $1.2 \times 1.2 \times 3.6$ , while the smaller part is simply a cube with side length 1.2. Constitutive behavior is modeled according to a compressible Neo–Hookean law (Young’s modulus  $E = 10.000$ , Poisson’s ratio  $\nu = 0.4$ ), and the density is set to  $\varrho_0 = 100$ . The left and right surfaces of the L-shaped block are both subject to a pressure load  $p(t) = 2,000 \times \sin(2\pi t)$  in the time interval  $t \in [0, 0.5]$ , and the body then moves freely in the time interval  $t \in [0.5, 5]$ , which adds up to a total of 500 time steps with the step size  $\Delta t = 0.01$ . A curved non-matching mortar interface is introduced to make the mortar setting as general as possible, with the outer surface of the cylindrical inclusion being chosen as slave side.

In order to assure exact algorithmic conservation of linear and angular momentum as well as mechanical energy, the energy-momentum method (EMM) initially proposed in [15] is employed as time integration scheme here. Some characteristic stages of deformation are also visualized in Fig. 8 and emphasize the strong nonlinearities involved in this simulation. However, the main focus of interest for the presented example lies in the mechanical conservation properties. As can be seen from Fig. 9, linear and angular momentum as well as mechanical energies are *exactly* conserved when combining the proposed dual mortar finite element discretization



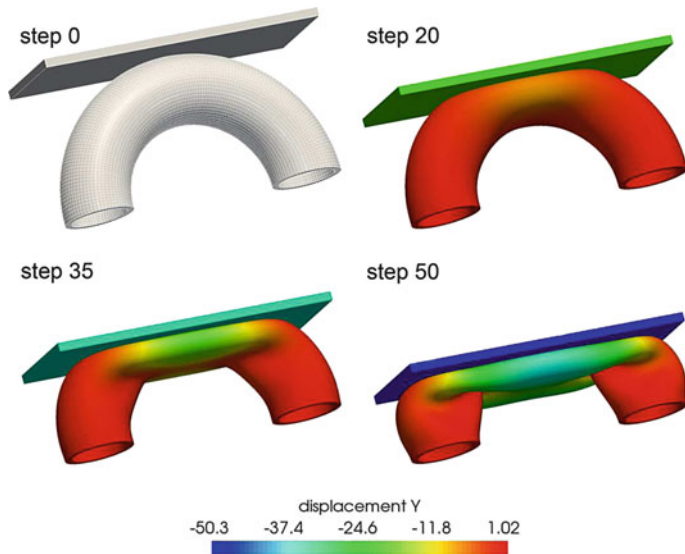
**Fig. 8** L-shaped block—problem setup, characteristic stages of deformation at  $t = 0, 1, 2, 3, 4, 5$  and numerical solution for the displacement magnitude  $\|u\|$



**Fig. 9** L-shaped block—exact algorithmic conservation of linear momentum (*top*), angular momentum (*middle*) and mechanical energies (*bottom*) over time  $t$

and EMM time integration. Linear momentum conservation is assured by using the same integration procedure for both mortar matrices  $\mathbf{D}$  and  $\mathbf{M}$ , while the mesh initialization procedure suggested in [13] and the EMM together guarantee angular momentum conservation. Finally, energy conservation is a direct consequence of





**Fig. 10** Torus impact—problem setup and characteristic stages of deformation

the employed time integration scheme, and could not be achieved when using other well-known implicit time integrators such as the generalized- $\alpha$  method [3].

## 4.2 Finite Deformation Contact Mechanics

The second example illustrates the robustness of the described mortar approach in the case of finite deformation contact with significant active set changes. Second-order finite element interpolation in combination with novel, recently proposed dual Lagrange multipliers is employed, see [12, 20] for details.

The considered test setup consists of a hollow half-torus (Neo–Hookean material model with Young’s modulus  $E = 100$ , Poisson’s ratio  $\nu = 0.3$ ) and a rigid planar surface. The major and minor radii of the half-torus are 76 and 24, respectively, and the wall thickness is 4.5. The bottom surfaces of the half-torus are completely fixed, and an impact is generated by moving the rigid wall towards the elastic body with a prescribed displacement  $u = 50$  accumulated over 50 quasi-static load steps. Figure 10 shows the finite element mesh consisting of 20-node hexahedral elements (with 50.720 nodes in total) as well as some characteristic stages of deformation.

The given impact situation can be considered extremely challenging for any employed active set strategy. However, as Table 1 exemplarily confirms for one representative load step, the semi-smooth Newton type active set strategy presented in [10, 11] and also used here does not have any problems with the described scenario, but resolves all nonlinearities (including the search for the correct active

**Table 1** Torus impact—convergence behavior of the semi-smooth Newton method in terms of the relative  $L^2$ -norm of the total residual for a representative load step

Step	Relative $L^2$ -norm of residual
1	7.31e+01 (*)
2	6.67e+01 (*)
3	3.54e+01 (*)
4	8.16e+00 (*)
5	4.76e−01
6	8.34e−05
7	9.66e−09

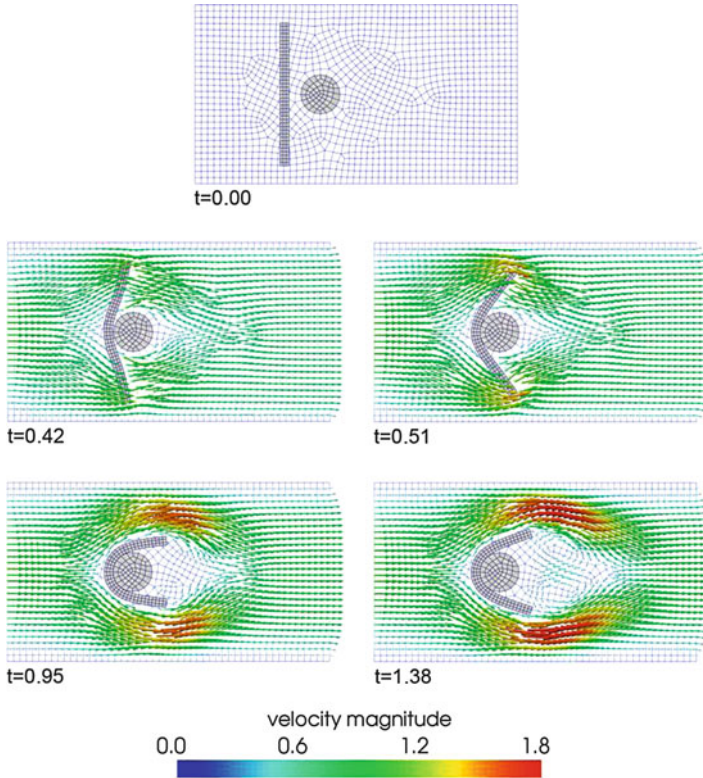
(\*) = change in active contact set

set) within only a few Newton–Raphson iteration steps. Owing to the underlying consistent linearization, quadratic convergence is obtained in the limit.

### 4.3 Fluid–Structure–Contact Interaction (FSCI)

The third numerical example presented here demonstrates the effectiveness of the proposed mortar finite element framework for fluid–structure–contact interaction (FSCI) and especially its ability to deal with finite structural deformations and contact in combination with classical fluid–structure interaction (FSI). A variety of problems in engineering and applied sciences require the simulation of unilateral contact of solids surrounded by an incompressible fluid. Important fields of application include machine parts, such as gaskets or sliding-contact bearings, and biomechanical systems, such as heart valves or capillary flow of red blood cells. All details on the computational framework for such coupled problems can be found in [9]. A beam-like structure (Young’s modulus  $E = 2,000$ , Poisson’s ratio  $\nu = 0.4$ ) is positioned in a two-dimensional channel flow, see Fig. 11. It should be pointed out that the given implementation in BACI is inherently three-dimensional, so that this 2D example is actually modeled as a 3D problem with just one layer of elements in the third direction. A parabolic inflow profile is applied as Dirichlet boundary condition at the left and a zero traction Neumann boundary condition is assumed at the outflow. All remaining channel boundaries are rigid walls with contact occurring between the beam-like structure and a circular obstacle. Exemplarily, 8-node hexahedral elements are used for both the fluid mesh and the structural discretization.

The resulting flow field and structural deformation including contact are illustrated in Fig. 11, giving an impression of this highly dynamic fluid–structure–contact interaction (FSCI) process. The beam-like structure exhibits large deformations: At first, they are primarily induced by fluid stresses resulting from the increasing fluid pressure, i.e., a typical fluid–structure interaction process is initiated. At later stages the gap between beam and obstacle closes completely and the structural deformation is then dominated by contact interaction. It should be mentioned that the given example represents more of a qualitative proof of concept for the successful integration of the mortar contact formulation into a

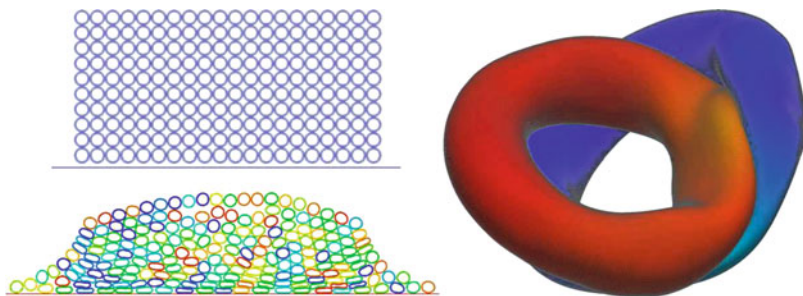


**Fig. 11** Beam-like structure in channel flow—finite element mesh (*top*), fluid velocity and structural deformation in several characteristic time steps

fixed-grid FSI framework. Further investigations can also be found in [9]. While the obtained preliminary results are definitely promising towards the simulation of more challenging FSCI applications, the complex physical phenomena occurring during the approach of the two bodies and the associated transition of boundary conditions from FSI type to contact type are not yet fully captured here due to an insufficient fluid mesh resolution.

#### 4.4 Large-Scale Simulations

The mortar finite element methods presented in this contribution are readily applicable to large-scale simulations of complex processes involving fluid dynamics, structural dynamics or several coupled physical fields. Especially the design of the numerical algorithms as described in Sect. 3, i.e., including parallel search procedures with hierarchic binary tree structures and dynamic load balancing,



**Fig. 12** Exemplary large-scale simulations—contact and self contact of 200 elastic rings in 2D (*left*) and impact of two thin-walled tori in 3D (*right*)

assures an excellent parallel scalability when solving very large simulation models with up to several million degrees of freedom. To give an impression of typical problem sizes, snapshots of two large-scale examples from computational contact dynamics with finite deformations and nonlinear material behavior are shown in Fig. 12. The finite element mesh for the 3D impact model, for example, consists of 4,255,360 hexahedral elements and 13,994,880 degrees of freedom in total. The numerical solution is performed in parallel on up to 120 cores, using an implicit time stepping scheme and 500 time increments to resolve all contact interactions.

## 5 Conclusions and Outlook

The most important aspects of mortar finite element methods as a non-conforming discretization technique for a wide range of applications in computational mechanics have been reviewed. In particular, several new approaches for the efficient parallel implementation of such mortar methods within a high performance computing framework have been addressed. Contact search algorithms based on hierarchical tree structures and especially a novel dynamic load balancing strategy specifically developed for mortar based discretization have been shown to be indispensable ingredients of efficient numerical algorithms.

Beyond classical applications in nonlinear solid mechanics and contact analysis, the application of mortar finite element methods within a simulation framework for fluid–structure–contact interaction has been addressed shortly. This can be seen as a first step towards more challenging multiphysics and multiscale simulations, which couple contact analysis with several other physical fields and take into account effects on different length scales. Ongoing and future work in this field focuses on modeling finite deformation contact between slender beams and coupled thermomechanical contact, with applications ranging from Brownian dynamics of polymers in fiber networks to heat conduction and mechanical dissipation due to frictional sliding in thermally loaded machine parts.

**Acknowledgements** The support of the first author (A.P.) by the TUM Graduate School is gratefully acknowledged.

## References

1. Ben Belgacem, F.: The mortar finite element method with Lagrange multipliers. *Numerische Mathematik* **84**(2), 173–197 (1999)
2. Bernardi, C., Maday, Y., Patera, A.T.: A new nonconforming approach to domain decomposition: the mortar element method. In: H. Brezis, J. Lions (eds.) *Nonlinear partial differential equations and their applications*, pp. 13–51. Pitman/Wiley: London/New York (1994)
3. Chung, J., Hulbert, G.M.: A time integration algorithm for structural dynamics with improved numerical dissipation: The generalized- $\alpha$  method. *Journal of Applied Mechanics* **60**, 371–375 (1993)
4. Ehrl, A., Popp, A., Gravemeier, V., Wall, W.A.: A mortar approach with dual Lagrange multipliers within a variational multiscale finite element method for incompressible flow. *Computer Methods in Applied Mechanics and Engineering*, submitted (2012)
5. Gitterle, M., Popp, A., Gee, M.W., Wall, W.A.: Finite deformation frictional mortar contact using a semi-smooth newton method with consistent linearization. *International Journal for Numerical Methods in Engineering* **84**(5), 543–571 (2010)
6. Heroux, M.A., Bartlett, R.A., Howle, V.E., Hoekstra, R.J., Hu, J.J., Kolda, T.G., Lehoucq, R.B., Long, K.R., Pawlowski, R.P., Phipps, E.T., Salinger, A.G., Thornquist, H.K., Tuminaro, R.S., Willenbring, J.M., Williams, A., Stanley, K.S.: An overview of the Trilinos project. *ACM Transactions on Mathematical Software* **31**(3), 397–423 (2005)
7. Karypis, G., Kumar, V.: A parallel algorithm for multilevel graph partitioning and sparse matrix ordering. *Journal of Parallel and Distributed Computing* **48**(1), 71–95 (1998)
8. Klöppel, T., Popp, A., Küttler, U., Wall, W.A.: Fluid–structure interaction for non-conforming interfaces based on a dual mortar formulation. *Computer Methods in Applied Mechanics and Engineering* **200**(45–46), 3111–3126 (2011)
9. Mayer, U.M., Popp, A., Gerstenberger, A., Wall, W.A.: 3D fluid–structure–contact interaction based on a combined XFEM FSI and dual mortar contact approach. *Computational Mechanics* **46**(1), 53–67 (2010)
10. Popp, A., Gee, M.W., Wall, W.A.: A finite deformation mortar contact formulation using a primal-dual active set strategy. *International Journal for Numerical Methods in Engineering* **79**(11), 1354–1391 (2009)
11. Popp, A., Gitterle, M., Gee, M.W., Wall, W.A.: A dual mortar approach for 3D finite deformation contact with consistent linearization. *International Journal for Numerical Methods in Engineering* **83**(11), 1428–1465 (2010)
12. Popp, A., Wohlmuth, B.I., Gee, M.W., Wall, W.A.: Dual quadratic mortar finite element methods for 3D finite deformation contact. *SIAM Journal on Scientific Computing*, accepted (2012)
13. Puso, M.A.: A 3D mortar method for solid mechanics. *International Journal for Numerical Methods in Engineering* **59**(3), 315–336 (2004)
14. Seshaiyer, P., Suri, M.: hp submeshing via non-conforming finite element methods. *Computer Methods in Applied Mechanics and Engineering* **189**(3), 1011–1030 (2000)
15. Simo, J.C., Tarnow, N.: The discrete energy-momentum method. Conserving algorithms for nonlinear elastodynamics. *Zeitschrift für Angewandte Mathematik und Physik* **43**(5), 757–792 (1992)
16. Stupkiewicz, S.: Finite element treatment of soft elastohydrodynamic lubrication problems in the finite deformation regime. *Computational Mechanics* **44**(5), 605–619 (2009)
17. Wall, W.A., Gee, M.W.: BACI: A multiphysics simulation environment. Tech. rep., Technische Universität München (2012)

18. Wohlmuth, B.I.: A mortar finite element method using dual spaces for the Lagrange multiplier. *SIAM Journal on Numerical Analysis* **38**(3), 989–1012 (2000)
19. Wohlmuth, B.I.: *Discretization methods and iterative solvers based on domain decomposition*. Springer-Verlag Berlin Heidelberg (2001)
20. Wohlmuth, B.I., Popp, A., Gee, M.W., Wall, W.A.: An abstract framework for a priori estimates for contact problems in 3D with quadratic finite elements. *Computational Mechanics*, DOI: 10.1007/s00466-012-0704-z (2012)
21. Wriggers, P.: *Computational contact mechanics*. John Wiley & Sons (2002)
22. Yang, B., Laursen, T.A.: A contact searching algorithm including bounding volume trees applied to finite sliding mortar formulations. *Computational Mechanics* **41**(2), 189–205 (2008)
23. Yang, B., Laursen, T.A.: A large deformation mortar formulation of self contact with finite sliding. *Computer Methods in Applied Mechanics and Engineering* **197**(6–8), 756–772 (2008)
24. Yang, B., Laursen, T.A.: A mortar-finite element approach to lubricated contact problems. *Computer Methods in Applied Mechanics and Engineering* **198**(47–48), 3656–3669 (2009)