# From Global to Local Radial Basis Function Collocation Method for Transport Phenomena

Božidar Šarler

*Laboratory for Multiphase Processes, University of Nova Gorica, Vipavska 13, SI-5000 Nova Gorica, Slovenia; E-mail: bozidar.sarler@p-ng.si*

**Abstract.** This article introduces basic concepts of meshless methods for solving partial differential equations in their strong form by collocation or least squares approximation. Global and local formulations are defined. The current achievements, based on the local form and collocation with radial basis functions are explained in detail. Heat transfer and fluid flow problems are treated. These achievements represent a simple, and at the same time more efficient version of the classical meshless radial basis function collocation (Kansa) method. Instead of global, the collocation is made locally over a set of overlapping domains of influence and the time-stepping is performed in an explicit way. Only small systems of linear equations with the dimension of the number of nodes included in the domain of influence have to be solved for each node. The computational effort thus grows roughly linearly with the number of the nodes. The represented approach thus overcomes the principal large scale bottleneck of the original Kansa method and widely opens space for industrial applications of the method. The purpose of this article is to give a concentrated information on this new method, which has already been successfully applied in macroscopic and microscopic transport phenomena field, accompanied with research requirements for the future. It is devoted to practicing engineers and researchers.

**Key words:** Radial basis function collocation method, transport phenomena, strong formulation, multiquadrics.

## Nomenclature

*Latin Symbols*

| | | |
|---|---|---|
| $\mathbf{b}$ | = | augmented vector of the system of linear equations |
| $c$ | = | coefficient of the weight function |
| $\underline{\mathbf{D}}$ | = | diffusion tensor |
| $\mathbf{f}$ | = | body force |
| $K$ | = | number of global approximation functions |
| $N$ | = | number of gridpoints |

*V.M.A. Leitão et al. (eds), Advances in Meshfree Techniques,* 257–282.

| | |
|---|---|
| $\mathbf{p}$ | = position vector |
| $P$ | = pressure |
| $\tilde{P}$ | = pressure correction |
| $S$ | = source |
| $t$ | = time |
| $\underline{\mathbf{T}}$ | = deviatoric part of the stress tensor |
| $\mathbf{v}$ | = velocity |
| $\tilde{\mathbf{v}}$ | = velocity correction |
| $\hat{\mathbf{v}}$ | = velocity estimate |

*Greek Symbols*

| | |
|---|---|
| $\alpha$ | = coefficient of the global approximation function |
| $\Delta t$ | = time-step |
| $\Delta t_P$ | = pressure iteration time-step |
| $\rho$ | = density |
| $\psi$ | = global approximation function |
| $\underline{\boldsymbol{\psi}}$ | = system matrix |
| $\underline{\boldsymbol{\Psi}}$ | = system matrix (boundary condition information included) |
| $\Phi$ | = general transport variable |
| $\theta$ | = scalar inhomogeneous part in Poisson equation |
| $\boldsymbol{\Theta}$ | = vector inhomogeneous part in Poisson equation |
| $\mathsf{A}$ | = transported-diffused variable relation function |
| $\Upsilon$ | = boundary condition indicator function |
| $\Gamma$ | = boundary |
| $\omega$ | = sub-domain |
| $\Omega$ | = domain |

*Superscripts*

| | |
|---|---|
| $l$ | = sub-domain indicator |
| $D$ | = Dirichlet boundary indicator |
| $N$ | = Neumann boundary indicator |
| $R$ | = Robin boundary indicator |

*Subscripts*

| | |
|---|---|
| $0$ | = initial time |
| $x$ | = Cartesian coordinate |
| $y$ | = Cartesian coordinate |
| $z$ | = Cartesian coordinate |
| $\varsigma$ | = coordinate indicator |
| $\zeta$ | = coordinate indicator |
| $\xi$ | = coordinate indicator |
| $\tau$ | = coordinate indicator |
| $\iota$ | = time level indicator |

*Mathematical Symbols*

| | |
|---|---|
| $\nabla$ | = divergence operator |

| | |
|---|---|
| $\nabla^2$ | = Laplace operator |
| $\sum$ | = sum |

# 1 Introduction

## 1.1 Motivation

The development of efficient as well as simple algorithms for the numerical solution of partial differential equations (PDEs) is of major interest in applied sciences and engineering. The most popular discrete approximate methods for PDEs are nowadays the finite difference (FDM) [22], the finite volume (FVM) [35], the finite element (FEM) [41], the spectral (SM) [3], and the boundary-domain integral (BDIM) [40] methods. Despite the powerful features of these methods, there are often substantial difficulties in applying them to realistic, geometrically complex three-dimensional transient problems. A common element of the mentioned methods is the need to create a polygonisation, either in the domain and/or on its boundary. This type of meshing is often the most time consuming part of the solution process and is far from being fully automated.

Figure 1 shows the most common types of space discretisation arrangements in numerical methods used for non-linear transport phenomena. The FDM discretisation shown in Figure 1(a), involves pointisation only. However, the points are restricted to coordinate directions and uniformity. The FVM discretisation shown in Figure 1(b) includes polygonisation with rectangles. The rectangles are restricted to coordinate directions. The FEM discretisation is shown in Figure 1(c) and includes polygonisation with triangles. The triangles might be of arbitrary dimension and orientation. The triangles can be exchanged with other types of polygons. The BDIM discretisation is shown in Figure 1(c) as well. It includes discretisation of the domain with polygons (cells) and discretisation of the boundary with straight lines (boundary elements). The DRBEM discretisation is shown in Figure 1(d). Instead of the discretisation of the domain with polygons (cells), the domain is discretised by pointisation and the boundary is discretised with straight lines (boundary elements). This method belongs to the so-called mesh reduction techniques, since the domain polygonisation is replaced by the domain pointisation. However, boundary polygonisation remains. A discretisation used in the Meshless or Mesh-free (MSM) methods is shown in Figure 1(e). It includes pointisation of the domain and boundary. The gridpoints can be arbitrarily spaced and non-uniform. The question mark at the right lower corner of Figure 1 stands for our inability of solving PDEs without gridpoints (Grid-less, Grid-free methods).

The numerical solution of coupled heat and mass transfer problems is becoming increasingly important as a result of the computational modelling needs in diverse modern technologies. A broad class of such heat, mass, momentum, and solute

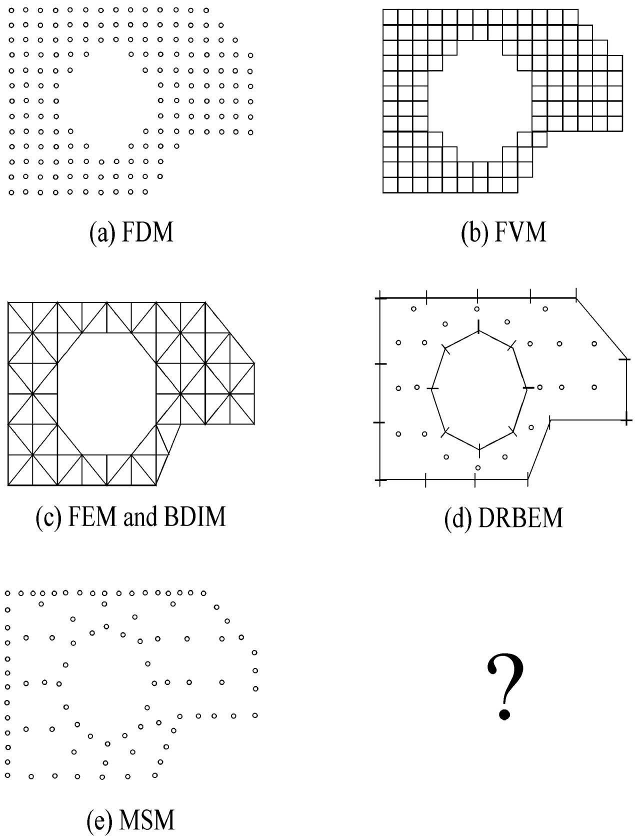

**Fig. 1.** Schematics of the domain discretisations used in numerical methods for the solution of partial differential equations. The question mark represents post-meshless methods.

transfer problems involves two or more phases, separated by free (steady-state) or moving (transient) interphase boundaries. Due to the existence of complex shaped interphase boundaries, most of the numerical simulations of engineering gas-liquid and liquid-solid two-phase flows conducted so far, have been based on averaged field equations with constitutive interphase relations solved on a fixed mesh. However, the diversity of the possibly involved length scales, inhomogeneities, and anisotropies, usually requires the adaptation of the mesh with respect to high field gradients and subsequent re-meshing. The most physically sound information can be directly perceived only from the numerical approaches that explicitly take into account the moving boundaries. The principal bottleneck in these types of numerical methods is the time consuming re-meshing of the evolving interphase boundaries and phase domains which limits such methods to problems with quite trivial phase patterns. The polygonisation problem is thus even more pronounced in such type of front-tracking approach.

### 1.2 Definition and Characteristics of Meshless Methods

The meshless, sometimes also called meshfree or mesh reduction methods establish a system of algebraic equations for the whole problem domain and boundary without polygonisation [1, 2, 15, 16]. Meshless methods use a set of nodes scattered within the problem domain as well as sets of nodes scattered on the boundaries of the domain to represent the problem domain and its boundaries. These sets of scattered nodes do not form a polygonisation (mesh), which means that no information on the geometrical connections between the nodes is required.

There exist a number of meshless methods such as the *Element free Galerkin methods*, *the Meshless local Petrov–Galerkin method*, *the Point interpolation method*, *the Point assembly method*, *the Finite point method*, *the Finite difference method with arbitrary irregular grids*, *Smoothed particle hydrodynamics*, *Reproducing kernel particle method*, *Kansa method*, etc.

The discussion in this article is limited to the very recent generation of meshless methods only. They are characterised by the following features:

- The governing equation is solved in its strong form.
- The formulation is almost independent on the problem dimension.
- The complicated geometry is easy to cope with.
- No polygonisation is needed.
- No integrations are needed.
- The method is very efficient.
- The method is very accurate.
- The methods is simple to learn.
- The method is simple to code.

## 2 Governing Equations

This part introduces the general transport equation, relevant types of boundary conditions and the reformulation of the governing transport equation into non-linear Poisson equation. It discusses the basic strategies of solving the non-linear Poisson equation in cases of weak and strong non-linearities.

### 2.1 General Transport Equation

Let us limit our discussion to solution of the general transport equation [32], defined on a fixed domain $\Omega$ with boundary $\Gamma$, standing for a reasonably broad spectra of mass, energy, momentum, and species transfer problems

$$\frac{\partial}{\partial t}[\rho \mathsf{A}(\Phi)] + \nabla \cdot [\rho \mathbf{v}\, \mathsf{A}(\Phi)] = \nabla \cdot (\underline{\mathbf{D}} \nabla \Phi) + S, \tag{1}$$

with $\rho$, $\Phi$, $t$, $\mathbf{v}$, $\underline{\mathbf{D}}$, and $S$ standing for density, transport variable, time, velocity, diffusion matrix

$$\underline{\mathbf{D}} \equiv \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ D_{31} & D_{32} & D_{33} \end{bmatrix}, \tag{2}$$

and source, respectively. The scalar function $\mathsf{A}$ stands for possible more involved constitutive relations between the conserved $\mathsf{A}(\Phi)$ and diffused $\Phi$ quantities. The solution of the governing equation for the transport variable at the final time $t_0 + \Delta t$ is sought, where $t_0$ represents the initial time and $\Delta t$ the positive time increment. The solution is constructed by the initial and boundary conditions that follow. The initial value of the transport variable $\Phi(\mathbf{p}, t)$ at a point with position vector $\mathbf{p}$ and time $t_0$ is defined through the known function $\Phi_0$

$$\Phi(\mathbf{p}, t) = \phi_0(\mathbf{p}, t); \quad \mathbf{p} \in \Omega + \Gamma. \tag{3}$$

The boundary $\Gamma$ is divided into not necessarily connected parts $\Gamma = \Gamma^D \cup \Gamma^N \cup \Gamma^R$ with Dirichlet, Neumann and Robin type boundary conditions, respectively. At the boundary point $\mathbf{p}$ with normal $\mathbf{n}_\Gamma$ and time $t_0 \leq t \leq t_0 + \Delta t$, these boundary conditions are defined through known functions $\Phi_\Gamma^D$, $\Phi_\Gamma^N$, $\Phi_\Gamma^R$, $\Phi_{\Gamma\,\mathrm{ref}}^R$

$$\Phi = \Phi_\Gamma^D; \quad \mathbf{p} \in \Gamma^D, \tag{4}$$

$$\frac{\partial}{\partial n_\Gamma}\Phi = \Phi_\Gamma^N; \quad \mathbf{p} \in \Gamma^N, \tag{5}$$

$$\frac{\partial}{\partial n_\Gamma}\Phi = \Phi_\Gamma^R(\Phi - \Phi_{\Gamma\,\mathrm{ref}}^R); \quad \mathbf{p} \in \Gamma^R. \tag{6}$$

## 2.2 Poisson Reformulation of the General Transport Equation

The general transport equation (Equation 1) can be reformulated into Poisson form. This form sometimes permits an easier theoretical treatment. The inhomogeneous part of the Poisson equation can be split into a scalar part and a divergence of the vector part

$$\nabla^2 \Phi = \theta + \nabla \cdot \mathbf{\Theta}, \tag{7}$$

$$\theta = \frac{\left\{\frac{\partial}{\partial t}[\rho\, \mathsf{A}(\Phi)] - S\right\}}{D}, \tag{8}$$

$$\mathbf{\Theta} = \frac{[\rho\, \mathbf{v}\, \mathsf{A}(\Phi) - \underline{\mathbf{D}}' \nabla \Phi]}{D}. \tag{9}$$

The diffusion matrix is split into a constant isotropic part $D\underline{\mathbf{I}}$ and a remaining anisotropic part $\underline{\mathbf{D}}'$,

$$\underline{\mathbf{D}} + D\underline{\mathbf{I}} + \underline{\mathbf{D}}', \underline{\mathbf{D}}' \equiv \begin{bmatrix} D_{11} - D & D_{12} & D_{13} \\ D_{21} & D_{22} - D & D_{23} \\ D_{31} & D_{32} & D_{33} - D \end{bmatrix}. \tag{10}$$

The partial time derivative might be approximated by a two

$$\frac{\partial}{\partial t}[\rho\ \mathsf{A}(\Phi)] \approx \frac{1}{\Delta t}[\rho\ \mathsf{A}(\Phi) - \rho_0\ \mathsf{A}(\Phi_0)], \tag{11}$$

or by a three-level finite difference scheme

$$\frac{\partial}{\partial t}[\rho\ \mathsf{A}(\Phi)] \approx \frac{1}{2\Delta t}[3\rho\ \mathsf{A}(\Phi) - 4\rho_0\ \mathsf{A}(\Phi_0) + \rho_{-1}\ \mathsf{A}(\Phi_{-1})], \tag{12}$$

with the following notation

$$\mathsf{A}_{-1} \equiv \mathsf{A}(t_0 - \Delta t), \tag{13}$$

$$\mathsf{A}_0 \equiv \mathsf{A}(t_0), \tag{14}$$

$$\mathsf{A} \equiv \mathsf{A}(t_0 + \Delta t). \tag{15}$$

The unknown can be discretized in time by a two level scheme with a time level indicator $0 \leq \iota \leq 1$. This gives the fully implicit scheme for $\iota = 1$, the fully explicit scheme for $\iota = 0$, and the Crank–Nicolson scheme for $\iota = 1/2$.

$$\nabla^2\Phi \approx \iota\nabla^2\Phi + (\iota - 1)\nabla^2\Phi_0, \tag{16}$$

$$\theta \approx \frac{\left\{\frac{\partial}{\partial t}[\rho\ \mathsf{A}(\Phi)] - \iota S - (\iota - 1)S_0\right\}}{D}, \tag{17}$$

$$\mathbf{\Theta} \approx \frac{[\iota\rho\mathbf{v}\ \mathsf{A}(\Phi) + (\iota - 1)\rho_0\mathbf{v}_0\ \mathsf{A}(\Phi_0) - \iota\underline{\mathbf{D}}'\nabla\Phi - (\iota - 1)\underline{\mathbf{D}}'_0\nabla\Phi_0]}{D}. \tag{18}$$

The involved parameters of the governing equation and boundary conditions are assumed to be dependent on the transport variable, space, and time. The solution of such type of non-linear equation requires iterations. Let us (for the sake of compact notation) assume further discussion in non-time discretized form. The inhomogeneous terms are, due to non-linearity, respectively expanded as

$$\theta = \theta + \frac{\partial\bar{\theta}}{\partial\Phi}(\Phi - \bar{\Phi}), \tag{19}$$

$$\mathbf{\Theta} = \bar{\mathbf{\Theta}} + \frac{\partial\bar{\mathbf{\Theta}}}{\partial\Phi}(\Phi - \bar{\Phi}), \tag{20}$$

with the over-bar denoting the value from previous iteration. The final form of the transformed equation, suitable for iterative solution then becomes

$$\nabla^2 \Phi = Q(\Phi), \tag{21}$$

$$Q(\Phi) = \bar{\theta} + \frac{\partial \bar{\theta}}{\partial \Phi}(\Phi - \bar{\Phi}) + \nabla \cdot \bar{\mathbf{\Theta}} + \nabla \cdot \frac{\partial \bar{\mathbf{\Theta}}}{\partial \Phi}(\Phi - \bar{\Phi}). \tag{22}$$

The source term can be treated by under-relaxation (recommended in case of weak non-linearity)

$$\nabla^2 \Phi = Q(\bar{\Phi}) + c_{\mathrm{rel}} \left[ Q(\bar{\Phi}) - Q(\bar{\bar{\Phi}}) \right], \tag{23}$$

with an under-relaxation coefficient $c_{\mathrm{rel}} < 1$. The two-fold over-bar denotes the known value two iterations ago. Alternatively, the source term can be treated explicitly (recommended in case of strong non-linearity)

$$\left[ \nabla^2 - \frac{\partial \bar{\theta}}{\partial \Phi} - \nabla \cdot \frac{\partial \bar{\mathbf{\Theta}}}{\partial \Phi} \right] \Phi = \bar{\theta} - \frac{\partial \bar{\theta}}{\partial \Phi}\bar{\Phi} + \nabla \cdot \bar{\mathbf{\Theta}} - \nabla \cdot \frac{\partial \bar{\mathbf{\Theta}}}{\partial \Phi}\bar{\Phi}. \tag{24}$$

The discretisation of Equations (23) and (24) and subsequent set-up of the algebraic equation systems is quite different. In case of Equation (23), the system matrix of the algebraic equation system need not to be recalculated each iteration and in the case of Equation (24), the system matrix of the algebraic equation system needs to be recalculated ateach iteration. Strategy used in Equation (24) usually requires less iterations than strategy used in Equation (23), however, each iteration is computationally more costly. Therefore, the strategy in Equation (23) is used for weak non-linear problems, and the strategy in Equation (24) is for the strong non-linear problems. For the strong non-linear problems the strategy in Equation (23) might require a very small under-relaxation and a huge number of iterations. The iterations over one time-step are completed when the criterion in Equation (25) is satisfied in all computational nodes $n = 1, 2, \ldots, N$

$$\max |\Phi_n - \bar{\Phi}_n| \leq \Phi_{\mathrm{itr}}. \tag{25}$$

The steady-state is achieved when the criterion in Equation (26) is satisfied in all computational nodes $n = 1, 2, \ldots, N$

$$\max |\Phi_n - \Phi_0| \leq \Phi_{\mathrm{ste}}. \tag{26}$$

The parameters $\Phi_{\mathrm{itr}}$ and $\Phi_{\mathrm{ste}}$ are denoted as iteration and steady-state limits. The iteration limit has to be an order of magnitude lower as the steady-state limit. If this is not true, a non-convergent virtual steady-state can be achieved. The derived formulas in this part are universally valid for all known spatial discretization types, including meshless.

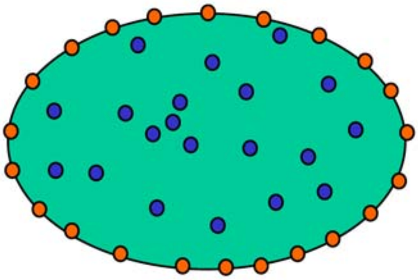

**Fig. 2.** Schematics of the polygon-free domain and boundary node arrangement.

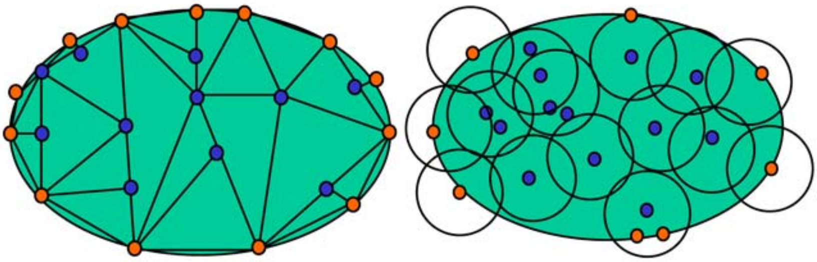

**Fig. 3.** Left: Schematics of the concept of contiguous (non-overlapping) sub-domains (typical for FEM). Right: Schematics of the concept of non-contiguous (overlapping) sub-domains (typical for MSM).

## 3 Spatial Discretisation in Meshless Methods

This part introduces the spatial discretisation (pointisation) in Meshless methods and the concept of overlapping sub-domain (Figures 2 to 6).

### 3.1 Pointisation

The value of the transport variable $\Phi_n$ is solved in a set of nodes $\mathbf{p}_n$; $n = 1, 2, \ldots, N$ of which $N_\Omega$ belong to the domain and $N_\Gamma$ to the boundary, i.e. $N = N_\Gamma + N_\Omega$.

## 4 Representation of Function over a Set of Nodes

This part introduces the concept of the representation of the function with global shape functions and the concepts of the calculation of shape function coefficients by collocation (interpolation) and by the approximation.

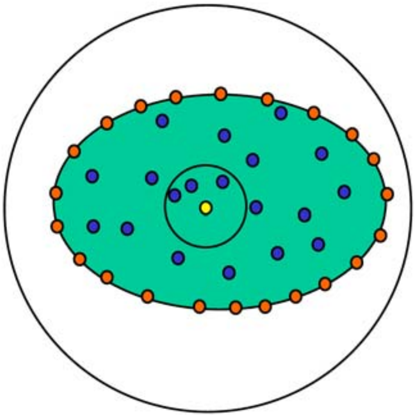

**Fig. 4.** Schematics of the local influence area (small circle) that encompasses only 4 nodes and global influence area (big circle) that encompasses all 43 nodes.

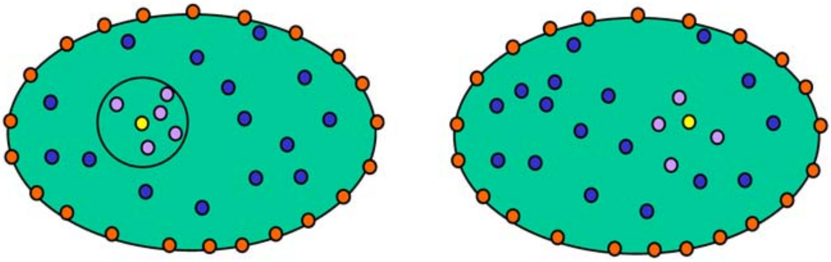

**Fig. 5.** Schematics of the sub-domains. In the left case the sub-domain is defined by the geometry of the circle. In the right case, the sub-domain is defined by the four nearest nodes to the central node.

### 4.1 Global Representation of Function

The representation of function over a set $l$ of (in general) non-equally spaced ${}_lN$ nodes ${}_l\mathbf{p}_n$; $n = 1, 2, \ldots, {}_lN$ is made in the following way

$$\Phi(\mathbf{p}) \approx \sum_{k=1}^{{}_lK} {}_l\psi_k(\mathbf{p})\, {}_l\alpha_k \tag{27}$$

${}_l\psi_k$ stand for the shape functions, ${}_l\alpha_k$ for the coefficients of the shape functions, and ${}_lK$ represents the number of the shape functions. The left lower index on entries of Equation (27) represents the sub-domain ${}_l\omega$ on which the coefficients ${}_l\alpha_k$ are determined. The total number of sub-domains is denoted by $L$. The sub-domains ${}_l\omega$ can in general be contiguous (overlapping) or non-contiguous (non-overlapping).

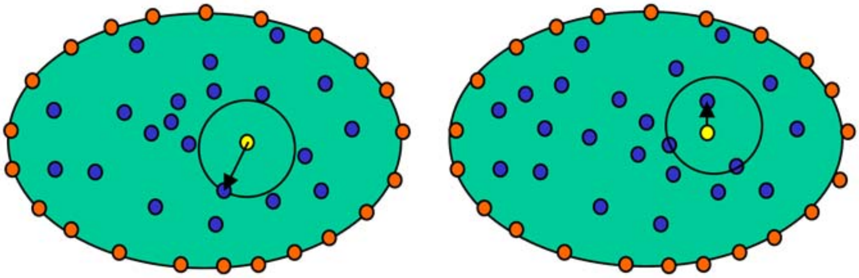

**Fig. 6.** Schematics of the definition of two quantities that measure the density of the data set. In the left case the fill distance is defined, which measures the maximum distance between two nodes in a sub-domain, and in the right case, the separation distance is defined, which measures the minimum distance between the two nodes in a sub-domain.

Each of the sub-domains ${}_l\omega$ includes ${}_lN$ grid-points of which ${}_lN_\Omega$ are in the domain and ${}_lN_\Gamma$ are on the boundary.

$$ {}_lN =_l N_\Omega +_l N_\Gamma . \tag{28}$$

For each of the sub-domains ${}_l\omega$, a separation distance is defined, which is the minimum distance between the two nodes, and the fill distance is defined, which is the maximum distance between the two nodes. In case the sub-domains are overlapping, the following is valid

$$\sum_{l=1}^{L} {}_lN > N. \tag{29}$$

The coefficients can be calculated from the sub-domain nodes in two distinct ways. The first way is collocation (interpolation) and the second way is approximation by the least squares method (see Figures 7 and 8, respectively).

Approximation is needed where data smoothing rather than interpolation is needed. This is because data are often inaccurate, contain noise or – as happens sometimes in practical applications – are too plentiful and cannot and need not be reasonably all interpolated at once. Moreover, smoothing is almost always required as long as problems are ill-posed, which means that their solution for theoretical reasons, is extremely sensitive to even the smallest changes in input data.

### 4.2 Calculation of Coefficients by Collocation

Let us assume the known function values ${}_l\Phi_n$ in the nodes ${}_l\mathbf{p}_n$ of sub-domain ${}_l\omega$. The collocation implies

$$\sum_{k=1}^{{}_lK} {}_l\psi_k({}_l\mathbf{p}_n)\, {}_l\alpha_k = \Phi({}_l\mathbf{p}_n);\ {}_lK = {}_lN. \tag{30}$$

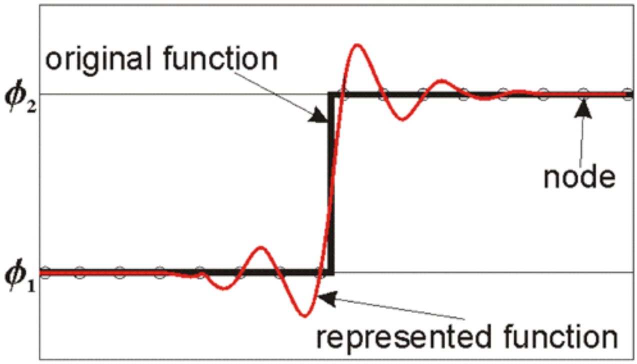

**Fig. 7.** A one-dimensional example of interpolation of the function.

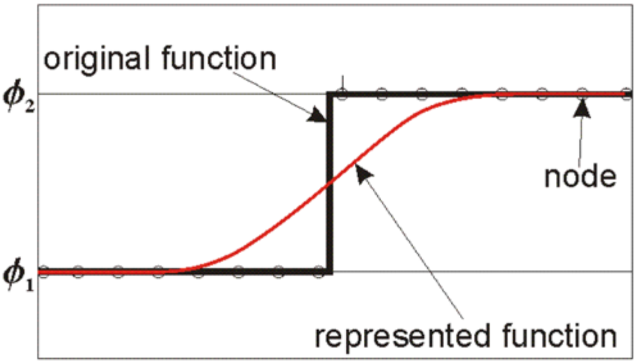

**Fig. 8.** A one-dimensional example of the least squares approximation of the function.

For the coefficients to be computable, the number of the shape functions has to match the number of the collocation points, and the collocation matrix has to be non-singular.

$$\sum_{k=1}^{{}_lK} {}_l\psi_k({}_l\mathbf{p}_n)\, {}_l\alpha_k = \Phi({}_l\mathbf{p}_n);\; {}_lK = {}_lN. \tag{31}$$

The system of Equations (31) can be written in matrix-vector notation

$${}_l\underline{\boldsymbol{\psi}}\, {}_l\boldsymbol{\alpha} = {}_l\boldsymbol{\Phi}, \tag{32}$$

with

$${}_l\underline{\psi}_{kn} = {}_l\psi_k({}_l\mathbf{p}_n), \tag{33}$$

$${}_l\Phi_n = \Phi({}_l\mathbf{p}_n). \tag{34}$$

The coefficients ${}_l\boldsymbol{\alpha}$ can be computed by inverting the system (32)

$$ {}_l\boldsymbol{\alpha} = {}_l\underline{\boldsymbol{\psi}}^{-1}\,{}_l\boldsymbol{\Phi}. \tag{35}$$

The matrices $\underline{\boldsymbol{\psi}}$ and $\underline{\boldsymbol{\psi}}^{-1}$ have dimension ${}_lN \times {}_lN$ and the vectors ${}_l\boldsymbol{\alpha}$ and ${}_l\boldsymbol{\Phi}$ have dimension ${}_lN$.

### 4.3 Calculation of Coefficients by Approximation

Let us assume the known function values ${}_l\Phi_n$ in the nodes ${}_l\mathbf{p}_n$ of sub-domain $\omega_l$. The least squares approximation implies that the following functional should be minimized

$$\Im[{}_l\boldsymbol{\alpha}(\mathbf{p})] = \sum_{n=1}^{{}_lN} {}_l\upsilon_n(\mathbf{p}) \sum_{k=1}^{{}_lK}[\psi_k({}_l\mathbf{p}_n)\,{}_l\alpha_k(\mathbf{p}) - \Phi({}_l\mathbf{p}_n)]^2, \tag{36}$$

with respect to change of ${}_l\boldsymbol{\alpha}(\mathbf{p})$, i.e.

$$\frac{\partial}{\partial\, {}_l\boldsymbol{\alpha}(\mathbf{p})}\Im[{}_l\boldsymbol{\alpha}(\mathbf{p})] = 0. \tag{37}$$

The number of the nodes ${}_lN$ used in the approximation has to be greater or equal to the number of the shape functions ${}_lK$. Weight functions ${}_l\upsilon_n$ have been introduced in Equation (36) in addition to the shape functions. The weight functions measure the relative importance of the node. The coefficients ${}_l\boldsymbol{\alpha}(\mathbf{p})$ depend on the center of the weight function $\mathbf{p}$. The coefficients ${}_l\boldsymbol{\alpha}(\mathbf{p})$ are respectively calculated from the system of linear equations

$$ {}_l\overset{\frown}{\underline{\boldsymbol{\psi}}}(\mathbf{p})\,{}_l\boldsymbol{\alpha}(\mathbf{p}) = {}_l\overset{\smile}{\underline{\boldsymbol{\psi}}}(\mathbf{p})\,{}_l\boldsymbol{\Phi}. \tag{38}$$

The matrices ${}_l\overset{\frown}{\underline{\boldsymbol{\psi}}}$ and ${}_l\overset{\smile}{\underline{\boldsymbol{\psi}}}$ are of the dimensions ${}_lK \times {}_lK$ and ${}_lK \times {}_lN$ respectively. Their entries are

$$ {}_l\overset{\frown}{\underline{\psi}}_{ik}(\mathbf{p}) = \sum_{n=1}^{{}_lN} {}_l\upsilon_n(\mathbf{p})\,{}_l\psi_i({}_l\mathbf{p}_n)\,{}_l\psi_k({}_l\mathbf{p}_n), \tag{39}$$

$$ {}_l\overset{\smile}{\underline{\psi}}_{in}(\mathbf{p}) = {}_l\upsilon_n(\mathbf{p})\,{}_l\psi_i({}_l\mathbf{p}_n). \tag{40}$$

The coefficients ${}_l\boldsymbol{\alpha}(\mathbf{p})$ in general depend on the weight function center $\mathbf{p}$. For practical reasons, their calculation is fixed to the central node of the sub-domain ${}_l\omega$, denoted by ${}_l\mathbf{p}$

$$ {}_l\overset{\frown}{\underline{\boldsymbol{\psi}}} \equiv {}_l\overset{\frown}{\underline{\boldsymbol{\psi}}}({}_l\mathbf{p}), \tag{41}$$

$$ {}_l\overset{\smile}{\underline{\boldsymbol{\psi}}} \equiv {}_l\overset{\smile}{\underline{\boldsymbol{\psi}}}({}_l\mathbf{p}), \tag{42}$$

$$ {}_l\boldsymbol{\alpha} \equiv {}_l\boldsymbol{\alpha}({}_l\mathbf{p}). \tag{43}$$

Similarly as in the case of collocation, an explicit expression for the calculation of the coefficients ${}_l\boldsymbol{\alpha}$ can be written

$$ {}_l\boldsymbol{\alpha} = {}_l\underline{\boldsymbol{\psi}}^{-1}\, {}_l\boldsymbol{\Phi}, \tag{44}$$

with

$$ {}_l\underline{\boldsymbol{\psi}}^{-1} = {}_l\,\overset{\frown}{\underline{\boldsymbol{\psi}}}{}^{-1}\, {}_l\,\overset{\smile}{\underline{\boldsymbol{\psi}}}\,. \tag{45}$$

The main difference between collocation and approximation is as follows: In collocation, the representation of the function exactly satisfies the nodal values ${}_l\Phi_n$. In approximation, the nodal values are approximated in the least squares sense. The advantage of approximation compared to collocation is that it usually gives better estimation of the derivatives, particularly in cases with sharp gradients (see Figures 7 and 8). The drawback over collocation is obvious. The nodal values are not exactly satisfied in approximation.

### 4.4 From the Representation of the Function to the Representation of the Partial Derivatives

By taking into account the expressions for the calculation of the coefficients ${}_l\boldsymbol{\alpha}$ (by collocation or by approximation), the representation of function $\Phi(\mathbf{p})$ can be expressed as

$$ \Phi(\mathbf{p}) \approx \sum_{k=1}^{{}_lK} {}_l\psi_k(\mathbf{p}) \sum_{n=1}^{{}_lN} {}_l\underline{\psi}_{kn}^{-1}\; {}_l\,\Phi_n \tag{46}$$

The first partial derivatives of $\Phi(\mathbf{p})$ can be expressed as

$$ \frac{\partial}{\partial p_\varsigma}\Phi(\mathbf{p}) \approx \sum_{k=1}^{{}_lK} \frac{\partial}{\partial p_\varsigma}\psi_k(\mathbf{p}) \sum_{n=1}^{{}_lN} {}_l\underline{\psi}_{kn}^{-1}\, {}_l\phi_n; \qquad \varsigma = x, y, z. \tag{47}$$

The second partial derivatives of $\Phi(\mathbf{p})$ can be expressed as

$$ \frac{\partial^2}{\partial p_\varsigma p_\xi}\Phi(\mathbf{p}) \approx \sum_{k=1}^{{}_lK} \frac{\partial^2}{\partial p_\varsigma p_\xi}\, {}_l\psi_k(\mathbf{p}) \sum_{n=1}^{{}_lN} {}_l\underline{\psi}_{kn}^{-1}\, {}_l\Phi_n; \qquad \varsigma, \xi = x, y, z. \tag{48}$$

### 4.5 Selection of Global Representation Functions

Let us introduce the Cartesian coordinate system with base vectors $\mathbf{i}_\varsigma$; $\varsigma = x, y, z$ and coordinates $p_\varsigma$; $\varsigma = x, y, z$, i.e.

$$ \mathbf{p} = \mathbf{i}_\varsigma p_\varsigma; \qquad \varsigma = x, y, z. \tag{49}$$

The following six polynomials can be used in 2D to represent the quadratic basis:

$$\Psi_1 = 1,\ \psi_2 = p_x,\ \psi_3 = p_y,\ \psi_4 = p_x p_x,\ \psi_5 = p_x p_y,\ \psi_6 = p_y p_y. \tag{50}$$

The following additional four polynomials have to be used in 3D to represent the quadratic basis:

$$\psi_7 = p_z,\ \psi_8 = p_x p_z,\ \psi_9 = p_y p_z,\ \psi_{10} = p_z p_z. \tag{51}$$

The radial basis functions [4, 28, 33] such as multi-quadrics

$$\psi_k(\mathbf{p}) = [(\mathbf{p} - \mathbf{p}_k) \cdot (\mathbf{p} - \mathbf{p}_k) + c^2]^{1/2}, \tag{52}$$

or inverse multi-quadrics

$$\psi_k(\mathbf{p}) = [(\mathbf{p} - \mathbf{p}_k) \cdot (\mathbf{p} - \mathbf{p}_k) + c^2]^{1/2}, \tag{53}$$

can be used in 2D or 3D. The calculation of the coefficients ${}_l\boldsymbol{\alpha}$ can be made by collocation or by approximation.

## 4.6 Selection of Weight Functions

As a weight function typically a radial basis function is chosen. A typical example is the polynomial-like shape function

$${}_l\upsilon_n(\mathbf{p}) = \begin{cases} \left[\frac{(\mathbf{p} -\ {}_l\mathbf{p}_n)\cdot(\mathbf{p} -\ {}_l\mathbf{p}_n) -\ {}_l\sigma_n}{(\mathbf{p} -\ {}_l\mathbf{p}_n)\cdot(\mathbf{p} -\ {}_l\mathbf{p}_n)\ {}_l\sigma_n}\right]; & |\mathbf{p} -\ {}_l\mathbf{p}_n| < \sigma_n, \\ 0; & |\mathbf{p} -\ {}_l\mathbf{p}_n| \geq \sigma_n, \end{cases} \tag{54}$$

or the Gaussian function

$${}_l\upsilon_n(\mathbf{p}) = \begin{cases} \exp(-{}_lc(\mathbf{p} -\ {}_l\mathbf{p}_n) \cdot (\mathbf{p} -\ {}_l\mathbf{p}_n)/\sigma_n^2; & |\mathbf{p} -\ {}_l\mathbf{p}_n| < \sigma_n, \\ 0; & |\mathbf{p} -\ {}_l\mathbf{p}_n| \geq \sigma_n. \end{cases} \tag{55}$$

The size of the circular (2D) or spherical (3D) support ${}_l\sigma_n$ is chosen to contain an appropriate number of nodes. The weight function shape factor ${}_lc_n$ depends on the mesh (non-)uniformity. In case of a uniform mesh, the best results are obtained with a large shape factor. In case of the non-uniform node arrangements, a smaller shape factor has to be chosen. The size of the shape factor depends on the non-uniformity of the node arrangement (see [23]).

## 4.7 Diffuse Approximation

A limited number of applications using the polynomial basis functions in connection with the least squares approximation and strong formulation has been developed under the name Diffuse Approximation (DA) [21, 25] for heat transfer and fluid flow problems.

### 4.8 Kansa Method

A broad class of meshfree methods in development today are based on Radial Basis Functions (RBFs) [4]. The RBF collocation method or Kansa method [11] is the simplest of them. This method has been further upgraded to symmetric collocation [10, 24], to modified collocation [8] and to indirect collocation [20]. The method has been already used in a broad spectrum of computational fluid dynamics problems [29] such as the solution of Navier–Stokes equations [17, 18] or porous media flow [28, 31] and the solution of solid-liquid phase change problems [12]. In contrast to advantages over mesh generation, all the listed attempts unfortunately fail to perform for large problems, because they produce fully populated matrices, sensitive to the choice of the free parameters in RBFs. Sparse matrices can be generated by the introduction of the compactly supported RBFs and the accuracy of such approach can be improved by the multilevel technique [7]. One of the possibilities for mitigating the large fully populated matrix problem is to employ the domain decomposition [19]. However, the domain decomposition re-introduces some sort of meshing which is not attractive. The concept of local collocation in the context of RBF-based solution of Poisson equation has been introduced in [14, 34]. For interpolation of the function value in a certain node the authors use only data in the (neighbouring) nodes that fall into the domain of influence of this node. The procedure results in a matrix that is of the same size as the matrix in the original Kansa method, however it is sparse. Circular domains of influence have been used in [14] and stencil-shaped domains in [34]. In [14], the one-dimensional and two-dimensional Poisson equation has been solved by using multiquadrics and inverse multiquadrics RBFs with a detailed analysis of the influence of the free parameter on the results. In [34], a class of linear and non-linear elasticity problems have been solved with a fixed free parameter. The differential quadrature method, that calculates the derivatives of a function by a weighted linear sum of functional values at its neighbouring nodes has been structured with the RBFs in [26]. Despite the local properties, the matrix still has a similar form as in [14, 34]. This paper reviews a new, even more simple meshfree solution procedure for solving the transport phenomena, which overcomes even the solution of the large sparse matrices.

## 5 Semi-Explicit Solution of the General Transport Equation

### 5.1 Reformulation

This part elaborates the semi-explicit solution of the general transport equation (Equation 1), subject to the initial condition presented in Equation (3) and the boundary conditions presented in Equations (4–6). The general transport equation can be transformed into following expression by taking into account the explicit discretization

$$\frac{\rho \mathsf{A}(\Phi) - \rho_0 \mathsf{A}_0}{\Delta t} + \nabla \cdot [\rho_0 \mathbf{v}_0 \mathsf{A}_0] = \nabla \cdot (\underline{\mathbf{D}}_0 \nabla \Phi_0) + S. \tag{56}$$

At time $t = t_0 + \Delta t$, the functions $\mathsf{A}(\Phi)$ and $S(\Phi)$ can be expanded as

$$\mathsf{A}(\Phi) \approx \bar{\mathsf{A}} + \frac{\partial \bar{\mathsf{A}}}{\partial \Phi}(\Phi - \bar{\Phi}), \tag{57}$$

$$S(\Phi) \approx \bar{S} + \frac{\partial \bar{S}}{\partial \phi}(\Phi - \bar{\Phi}), \tag{58}$$

$$\frac{\rho \bar{\mathsf{A}} + \rho \bar{\mathsf{A}}_{l\Phi}(\Phi - \bar{\Phi}) - \rho_0 \ \mathsf{A}_0}{\Delta t} + \nabla \cdot [\rho_0 \mathbf{v}_0 \ \mathsf{A}_0]$$
$$= \nabla \cdot (\underline{\mathbf{D}}_0 \nabla \Phi_0) + \bar{S} + \bar{S}_{l\Phi}(\Phi - \bar{\Phi}). \tag{59}$$

The unknown function value can be extracted from the above equation

$$\Phi = \frac{\rho_0 \ \mathsf{A}_0 - \rho \bar{\mathsf{A}} + \rho \frac{\partial \bar{\mathsf{A}}}{\partial \Phi} \bar{\Phi}}{\rho \frac{\partial \bar{\mathsf{A}}}{\partial \Phi} - \Delta \frac{\partial \bar{S}}{\partial \Phi}}$$
$$+ \frac{\Delta t \left[ \nabla \cdot (\underline{\mathbf{D}}_0 \nabla \Phi_0) - \nabla \cdot (\rho_0 \mathbf{v}_0 \ \mathsf{A}_0) + \bar{S} - \frac{\partial \bar{S}}{\partial \Phi} \bar{\Phi} \right]}{\rho \frac{\partial \bar{\mathsf{A}}}{\partial \Phi} - \Delta t \frac{\partial \bar{S}}{\partial \Phi}}. \tag{60}$$

The solution of the above equation in grid-point $\mathbf{p}_n$ can be calculated as

$$\Phi_n = \frac{\rho_{n0} \ \mathsf{A}_{n0} - \rho_n \bar{\mathsf{A}}_n + \rho_n \frac{\partial \bar{\mathsf{A}}}{\partial \Phi_n} \bar{\Phi}_n}{\rho \frac{\partial \bar{\mathsf{A}}}{\partial \Phi_n} - \Delta t \frac{\partial \bar{S}}{\partial \Phi_n}}$$

$$+ \frac{\Delta t \left[ \nabla \cdot (\underline{\mathbf{D}}_{0n} \nabla \Phi_{0n}) - \nabla \cdot (\rho_{0n} \mathbf{v}_{0n} \ \mathsf{A}_{0n}) + \bar{S}_n - \frac{\partial \bar{S}}{\partial \Phi_n} \bar{\Phi}_n \right]}{\rho \frac{\partial \bar{\mathsf{A}}}{\partial \Phi_n} - \Delta t \frac{\partial \bar{S}}{\partial \Phi_n}}. \tag{61}$$

The calculation of the convective term $\nabla \cdot (\rho_{0n} \mathbf{v}_{0n} \mathsf{A}_{0n})$ includes derivatives of the form (see Appendix)

$$\frac{\partial}{\partial p_\varsigma} \rho_{0n} \ v_{\varsigma 0n} \ \mathsf{A}_{0n} \approx \varsigma = x, y, z. \tag{62}$$

These derivatives can be evaluated as

$$\frac{\partial}{\partial p_\varsigma} \rho_{0n} \ v_{\varsigma 0n} \ \mathsf{A}_{0n} \approx \sum_{k=1}^{K} \frac{\partial}{\partial p_\varsigma} \psi_k(\mathbf{p}_n) \sum_{n=1}^{{}_lN} {}_l\psi_{kn}^{-1} \ {}_l(\rho_0 \ v_{\varsigma 0} \ \mathsf{A}_0)_n; \quad \varsigma = x, y, z. \tag{63}$$

The calculation of the diffusive term $\nabla \cdot (\underline{\mathbf{D}}_{0n}\ \nabla\ \Phi_{0n})$ includes derivatives of the form (see Appendix)

$$\frac{\partial}{\partial p_\varsigma} D_{0n\zeta\xi},\ \frac{\partial^2}{\partial p_\varsigma \partial p_\tau} D_{0n\zeta\xi},\ \frac{\partial}{\partial p_\varsigma} \Phi_{0n\zeta},\ \frac{\partial^2}{\partial p_\varsigma \partial p_\tau} \Phi_{0n\zeta};\qquad \varsigma, \tau, \zeta, \xi = x, y, z. \quad (64)$$

These derivatives can be evaluated as

$$\frac{\partial}{\partial p_\varsigma} D_{0n\xi\zeta} \approx \sum_{k=1}^{K} \frac{\partial}{\partial p_\varsigma} \psi_k(\mathbf{p}_n) \sum_{n=1}^{{}_lN} {}_l\psi_{kn}^{-1}\ {}_lD_{0n\xi\zeta};\qquad \varsigma = x, y, z, \quad (65)$$

$$\frac{\partial^2}{\partial p_\varsigma \partial p_\tau} D_{0n\xi\zeta} \approx \sum_{k=1}^{K} \frac{\partial^2}{\partial p_\varsigma \partial p_\tau} \psi_k(\mathbf{p}_n) \sum_{n=1}^{{}_lN} {}_l\psi_{kn}^{-1}\ {}_lD_{0n\xi\zeta};\qquad \varsigma = x, y, z. \quad (66)$$

The solution procedure for the governing Equation (1) and the boundary conditions presented in Equations (4–6) now follow the below defined steps 1–5:

**Step 1**:
First, the initial conditions are set in the domain and boundary nodes and the derivatives required in the convective and diffusive terms are calculated from the known nodal values.

**Step 2**:
Equation (61) is employed to calculate the new values of the variable $\Phi_n$ at time $t_0 + \Delta t$ in the domain nodes.

What follows defines variable $\Phi_n$ at time $t_0 + \Delta t$ in the Dirichlet, Neumann, and Robin boundary nodes. For this purpose, the coefficients ${}_l\boldsymbol{\alpha}$ have to be determined from the new values in the domain and from the information on the boundary conditions.

Let us introduce domain, Dirichlet, Neumann, and Robin boundary indicators. These indicators are defined as

$$\Upsilon_{\Omega n} = \begin{cases} 1; & \mathbf{p}_n \in \Omega, \\ 0; & \mathbf{p}_n \notin \Omega, \end{cases} \qquad \Upsilon_{\Gamma n}^{D} = \begin{cases} 1; & \mathbf{p}_n \in \Gamma^D, \\ 0; & \mathbf{p}_n \notin \Gamma^D, \end{cases}$$

$$\Upsilon_{\Gamma n}^{N} = \begin{cases} 1; & \mathbf{p}_n \in \Gamma^N, \\ 0; & \mathbf{p}_n \notin \Gamma^N, \end{cases} \qquad \Upsilon_{\Gamma n}^{R} = \begin{cases} 1; & \mathbf{p}_n \in \Gamma^R, \\ 0; & \mathbf{p}_n \notin \Gamma^R. \end{cases} \quad (67)$$

**Step 3: The collocation version**
Consider the collocation version for the calculation of the coefficients. This implies

$$\sum_{k=1}^{{}_lN} {}_l\Upsilon_{\Omega n}\ {}_l\psi_k({}_l\mathbf{p}_n)\ {}_l\alpha_k + \sum_{k=1}^{{}_lN} {}_l\Upsilon_{\Gamma n}^{D}\ {}_l\psi_k({}_l\mathbf{p}_n)\ {}_l\alpha_k$$

$$
\begin{aligned}
&+\sum_{k=1}^{{}_lN} {}_l\Upsilon_{\Gamma n}^{N}\,\frac{\partial}{\partial n_\Gamma}\,{}_l\psi_k({}_l\mathbf{p}_n)\,{}_l\alpha_k+\sum_{k=1}^{{}_lN} {}_l\Upsilon_{\Gamma n}^{R}\,\frac{\partial}{\partial n_\Gamma}\,{}_l\psi_k({}_l\mathbf{p}_n)\,{}_l\alpha_k\\
&= {}_l\Upsilon_{\Omega n}\,{}_l\Phi_n+\,{}_l\Upsilon_{\Gamma n}^{D}\,\,{}_l\Phi_n^{D}+\,{}_l\Upsilon_{\Gamma n}^{N}\,\,{}_l\Phi_n^{N}\\
&+\,{}_l\Upsilon_{\Gamma n}^{R}\,{}_l\Phi_{\Gamma n}^{R}\left(\sum_{k=1}^{{}_lN}\,{}_l\psi_k({}_l\mathbf{p}_n)\,{}_l\alpha_k-\,{}_l\Phi_{\Gamma\,\mathrm{ref}n}^{R}\right).
\end{aligned}
\tag{68}
$$

The Robin boundary conditions have been represented by

$$
{}_l\Phi_{\Gamma n}^{R}({}_l\Phi_n-\,{}_l\Phi_{\Gamma\,\mathrm{ref}n}^{R})\approx\,{}_l\Phi_{\Gamma n}^{R}\left(\sum_{k=1}^{{}_lN}\,{}_l\psi_k({}_l\mathbf{p}_n)\,{}_l\alpha_k-\,{}_l\Phi_{\Gamma\,\mathrm{ref}n}^{R}\right).
\tag{69}
$$

The calculation of the coefficients ${}_l\boldsymbol{\alpha}$ follows from the following system of linear equations

$$
{}_l\underline{\boldsymbol{\Psi}}_l\boldsymbol{\alpha}=\,{}_l\mathbf{b},
\tag{70}
$$

with the system matrix coefficients

$$
\begin{aligned}
{}_l\underline{\Psi}_{nk}&={}_l\Upsilon_{\Omega n}\,{}_l\psi_k({}_l\mathbf{p}_n)+\,{}_l\Upsilon_{\Gamma n}^{D}\,{}_l\psi_k({}_l\mathbf{p}_n)+\,{}_l\Upsilon_{\Gamma n}^{N}\frac{\partial}{\partial n_\Gamma}\,{}_l\psi_k({}_l\mathbf{p}_n)\\
&\qquad+\,{}_l\Upsilon_{\Gamma n}^{R}\left[\frac{\partial}{\partial n_\Gamma}\,{}_l\psi_k({}_l\mathbf{p}_n)-\,{}_l\Phi_{\Gamma n}^{R}\sum_{k=1}^{{}_lN}\,{}_l\psi_k({}_l\mathbf{p}_n)\right],
\end{aligned}
\tag{71}
$$

and with the augmented right-hand side vector

$$
{}_l\mathbf{b}_n=\,{}_l\Upsilon_{\Omega n}\Phi_n+\,{}_l\Upsilon_{\Gamma n}^{D}\Phi_n^{D}+\,{}_l\Upsilon_{\Gamma n}^{N}\Phi_n^{N}-\,{}_l\Upsilon_{\Gamma n}^{R}\,{}_l\Phi_{\Gamma n}^{R}\,{}_l\Phi_{\Gamma\,\mathrm{ref}n}^{R}.
\tag{72}
$$

**Step 4: The approximation version**

Consider the least squares approximation version for the calculation of the coefficients. This implies

$$
\begin{aligned}
\Im[{}_l\boldsymbol{\alpha}]&=\sum_{n=1}^{{}_lN}\,{}_l\Upsilon_{\Omega n}\,{}_l\upsilon_n({}_l\mathbf{p})\sum_{k=1}^{{}_lN}[{}_l\psi_k({}_l\mathbf{p}_n){}_l\alpha_k-\,{}_l\Phi_n]^2\\
&+\sum_{n=1}^{{}_lN}\,{}_l\Upsilon_{\Gamma n}^{D}\,{}_l\upsilon_n({}_l\mathbf{p})\sum_{k=1}^{{}_lK}[{}_l\psi_k({}_l\mathbf{p}_n){}_l\alpha_k-\,{}_l\Phi_n]^2\\
&+\sum_{n=1}^{{}_lN}\,{}_l\Upsilon_{\Gamma n}^{N}\,{}_l\upsilon_n({}_l\mathbf{p})\sum_{k=1}^{{}_lK}\left[\frac{\partial}{\partial n_\Gamma}\,{}_l\psi_k({}_l\mathbf{p}_n){}_l\alpha_k-\,{}_l\Phi_{\Gamma n}^{N}\right]^2\\
&+\sum_{n=1}^{N_l}\,{}_l\Upsilon_{\Gamma n}^{R}\,{}_l\upsilon_n({}_l\mathbf{p})\sum_{k=1}^{{}_lK}\left[\frac{\partial}{\partial n_\Gamma}\,{}_l\psi_k({}_l\mathbf{p}_n){}_l\alpha_k-\,{}_l\Phi_{\Gamma n}^{R}({}_l\psi_k({}_l\mathbf{p}_n){}_l\alpha_k-\,{}_l\Phi_{\Gamma\,\mathrm{ref}n}^{R})\right]^2.
\end{aligned}
\tag{73}
$$

The same representation of the Robin boundary conditions is made as in the collocation case.

The minimisation of the functional $\Im[{}_l\boldsymbol{\alpha}]$ with respect to the coefficients ${}_l\boldsymbol{\alpha}$ gives the following system of linear equations

$$
{}_l\underline{\boldsymbol{\Psi}}\, {}_l\boldsymbol{\alpha} = {}_l\mathbf{b}, \tag{74}
$$

with the system matrix coefficients

$$
\begin{aligned}
{}_l\underline{\Psi}_{ik} = {} & \sum_{n=1}^{{}_lN} {}_l\Upsilon_{\Omega n} \upsilon_n({}_l\mathbf{p})\, {}_l\psi_k(\mathbf{p}_n)\, {}_l\psi_k(\mathbf{p}_n) \\
& + \sum_{n=1}^{{}_lN} {}_l\Upsilon^D_{\Gamma n} \upsilon_n({}_l\mathbf{p})\, {}_l\psi_i(\mathbf{p}_n)\, {}_l\psi_k(\mathbf{p}_n) \\
& + \sum_{n=1}^{{}_lN} {}_l\Upsilon^N_{\Gamma n} \upsilon_n({}_l\mathbf{p}) \frac{\partial}{\partial n_\Gamma}\, {}_l\psi_i({}_l\mathbf{p}_n) \frac{\partial}{\partial n_\Gamma}\, {}_l\psi_k({}_l\mathbf{p}_n) \\
& + \sum_{n=1}^{{}_lN} {}_l\Upsilon^R_{\Gamma n} \upsilon_n({}_l\mathbf{p}) \left( {}_l\Phi^R_{\Gamma\,\mathrm{ref}n} \psi_i({}_l\mathbf{p}_n) + \frac{\partial}{\partial n_\Gamma}\, {}_l\psi_i({}_l\mathbf{p}_n) \right) \\
& \times \left( {}_l\Phi^R_{\Gamma\,\mathrm{ref}n}\, {}_l\psi_k({}_l\mathbf{p}_n) + \frac{\partial}{\partial n_\Gamma}\, {}_l\psi_k({}_l\mathbf{p}_n) \right)
\end{aligned} \tag{75}
$$

and with the augmented right-hand side vector

$$
\begin{aligned}
{}_l\mathbf{b}_i = {} & \sum_{n=1}^{{}_lN} {}_l\Upsilon_{\Omega n}\, {}_l\upsilon_n({}_l\mathbf{p})\, {}_l\psi_i({}_l\mathbf{p}_n)\, {}_l\Phi_n + \sum_{n=1}^{{}_lN} {}_l\Upsilon^D_{\Gamma n}\, {}_l\upsilon_n({}_l\mathbf{p})\, {}_l\psi_i({}_l\mathbf{p}_n)\, {}_l\Phi^D_{\Gamma n} \\
& + \sum_{n=1}^{{}_lN} {}_l\Upsilon^N_{\Gamma n}\, {}_l\upsilon_n({}_l\mathbf{p}) \frac{\partial}{\partial n_\Gamma}\, {}_l\psi_i({}_l\mathbf{p}_n)\, {}_l\Phi^N_{\Gamma n} \\
& + \sum_{n=1}^{{}_lN} {}_l\Upsilon^R_{\Gamma n}\, {}_l\upsilon_n({}_l\mathbf{p}) \left( ({}_l\Phi^R_{\Gamma n})^2\, {}_l\Phi^R_{\Gamma\,\mathrm{ref}n}\, {}_l\psi_i({}_l\mathbf{p}_n) + {}_l\Phi^R_{\Gamma n}\, {}_l\Phi^R_{\Gamma\,\mathrm{ref}n} \frac{\partial}{\partial n_\Gamma}\, {}_l\psi_i({}_l\mathbf{p}_n) \right).
\end{aligned} \tag{76}
$$

After the coefficients are calculated, the unknown values of the unknown ${}_l\boldsymbol{\Phi}$ in the Dirichlet, Neumann, and Robin nodes at time $t_0 + \Delta t$ are determined from the global representation Equation (27).

### Step 5

The iteration and steady-state checks are performed. In case the iteration criterion is passed, calculation of the new time-step is performed. In case the steady-state criterion is passed or the time of calculation exceeds the predetermined time of interest, the calculation is stopped.

The performance of the presented method for diffusion problems can be found in [27], for convection-diffusion problems in [36], and for moving boundary problems with moving interfaces and moving domains in [38]. The microscopic species transfer problems have been dealt with by Kovačević and Šarler [13], based on the dynamic r-adaptivity of the nodal arrangement.

## 6 Explicit Solution of the Coupled Mass and Momentum Transport Equations

This part discusses the special issues encountered in mesh-free solution of the coupled mass

$$\frac{\partial}{\partial t}\rho + \nabla \cdot (\rho \mathbf{v}) = 0, \tag{77}$$

and momentum conservation equation

$$\frac{\partial}{\partial t}(\rho \mathbf{v}) + \nabla \cdot (\rho \mathbf{v}\mathbf{v}) = -\nabla P + \nabla \cdot \underline{\mathbf{T}} + \mathbf{f}, \tag{78}$$

where $P$, $\underline{\mathbf{T}}$, and $\mathbf{f}$ represent the pressure, the deviatoric part of the stress tensor, and the body force, respectively. For the sake of brevity, initial conditions and only the Dirichlet velocity boundary conditions are assumed

$$\mathbf{v}(\mathbf{p}, t) = \mathbf{v}_0(\mathbf{p}, t); \quad \mathbf{p} \in \Omega + \Gamma, \tag{79}$$

$$\mathbf{v} = \mathbf{v}_\Gamma^D; \quad \mathbf{p} \in \Gamma^D. \tag{80}$$

We seek the solution of the pressure field at time $t_0$ and the pressure and velocity fields at time $t_0 + \Delta t$. The main goal is to formulate the equations that can be solved by the "local" methodology, described in the previous part.

The pressure field is solved by taking the divergence of the momentum conservation equation

$$\nabla^2 P = -\frac{\partial}{\partial t}\nabla \cdot (\rho \mathbf{v}) - \nabla \cdot [\nabla \cdot (\rho \mathbf{v}\mathbf{v})] + \nabla \cdot (\nabla \cdot \underline{\mathbf{T}}) + \nabla \cdot \mathbf{f}. \tag{81}$$

The pressure is actually calculated from a false transient of the following equation towards the steady-state:

$$\frac{\partial}{\partial t}P = -\nabla^2 P - \frac{\partial}{\partial t}\nabla \cdot (\rho \mathbf{v}) - \nabla \cdot [\nabla \cdot (\rho \mathbf{v}\mathbf{v})] + \nabla \cdot (\nabla \cdot \underline{\mathbf{T}}) + \nabla \cdot \mathbf{f}. \tag{82}$$

Neumann boundary conditions for the Pressure Poisson equation are obtained by multiplication of the momentum equation with the normal derivative

$$\frac{\partial P}{\partial n_\Gamma} = \left[ -\frac{\partial}{\partial t}(\rho \mathbf{v}_\Gamma^D) - \nabla \cdot (\rho \mathbf{v}_\Gamma^D \mathbf{v}_\Gamma^D) + \nabla \cdot (\nabla \cdot \underline{\mathbf{T}}) + \nabla \cdot \mathbf{f} \right] \cdot \mathbf{n}_\Gamma. \tag{83}$$

The initial pressure is calculated as

$$P_0 = \bar{P}_0 + \big[ -\nabla^2 \bar{P}_0 - \nabla \cdot [\nabla \cdot (\rho_0 \mathbf{v}_0 \mathbf{v}_0] + \nabla \cdot [\nabla \cdot \underline{\mathbf{T}}_0) + \nabla \cdot \mathbf{f}_0 \big] \Delta t_P, \tag{84}$$

where $\bar{P}$ represents the value from previous iteration and $\Delta t_P$ the artificial pressure transient time-step. The initial value of $\bar{P}_0$ can be set to 0. After the calculation of the pressure field at time $t_0$, the new velocity field at time $t_0 + \Delta t$ is calculated from

$$\hat{\mathbf{v}} = \left[ \frac{\rho_0 \mathbf{v}_0}{\Delta t} - \nabla \cdot (\rho \mathbf{v}_0 \mathbf{v}_0) - \nabla P_0 + \nabla \cdot \underline{\mathbf{T}} + \mathbf{f} \right] \frac{\Delta t}{\rho}. \tag{85}$$

The calculated velocity is denoted by $\hat{\mathbf{v}}$, because it does not comply with the mass conservation in general. In order to assure the compliance of the new velocity field with the mass conservation, the following pressure and velocity corrections are made. The calculated velocity is corrected by $\tilde{\mathbf{v}}$, which ensures

$$\nabla \cdot (\rho \mathbf{v}) = \nabla \cdot [\rho(\hat{\mathbf{v}} + \tilde{\mathbf{v}})] = -\frac{\partial \rho}{\partial t}. \tag{86}$$

Consider that the velocity correction occurs exclusively due to action of the pressure correction

$$\frac{\rho}{\Delta t} \tilde{\mathbf{v}} = -\nabla \tilde{P}, \tag{87}$$

$$\nabla^2 \tilde{P} = \frac{1}{\Delta t} \nabla \cdot (\rho \hat{\mathbf{v}}) + \frac{1}{\Delta t} \frac{\partial \rho}{\partial t} \tag{88}$$

with the pressure correction Neumann boundary conditions

$$\frac{\partial \tilde{P}}{\partial n_\Gamma} = 0. \tag{89}$$

The pressure correction is actually calculated from a false transient of the following equation towards the steady-state:

$$\frac{\partial \tilde{P}}{\partial t} = -\nabla^2 \tilde{P} + \frac{1}{\Delta t} \nabla \cdot (\rho \hat{\mathbf{v}}) + \frac{1}{\Delta t} \frac{\partial \rho}{\partial t}. \tag{90}$$

The pressure correction is calculated as

$$\tilde{P} = \bar{\tilde{P}} + \left[ -\nabla^2 \tilde{P} + \frac{1}{\Delta t} \nabla \cdot (\rho \hat{\mathbf{v}}) + \frac{\rho - \rho_0}{\Delta t^2} \right] \Delta t_P. \tag{91}$$

After calculation of the pressure correction, the pressure and velocity fields are updated

$$P = P + \tilde{P}, \tag{92}$$

$$\mathbf{v} = \hat{\mathbf{v}} - \nabla \tilde{P} \frac{\Delta t}{\rho}, \tag{93}$$

and the solution procedure is advanced to a new time-step. The formulation, described here, for solving fluid flow problems, has been numerically implemented by Perko [23] in the context of Diffuse Approximation and by Divo and Kassab [9] in the context of local Kansa method.

## 7 Conclusions

This article reviews a new (very) simple meshfree formulation for solving a wide range of transport phenomena. The numerical tests [6], included in the cited references of the new method, show much higher accuracy of the method as compared with the classical FDM. The only exception observed is the solution at short times immediately after Dirichlet jump where similar numerical observation properties are observed [27]. The time-marching is performed in a simple explicit way. The governing equation is solved in its strong form. No polygonisation and integrations are needed. The developed method is almost independent of the problem dimension. The complicated geometry can easily be coped with. The method appears efficient, because it does not require a solution of a large system of equations like the original Kansa method. Instead, small systems of linear equations have to be solved in each time-step for each node and associated domain of influence, probably representing the most natural and automatic domain decomposition. This feature of the developed method represents its principal difference from the other related local approaches, where the resultant matrix is large and sparse [7, 14, 19, 34]. The method is simple to learn and simple to code. The method can cope with very large problems since the computational effort grows approximately linear with the number of the nodes. The local approach, described in this article could be extended in a straightforward way to tackle other types of partial differential equations. Despite the fact that the represented method behaves excellent, the underlying basic mathematical theory is still lacking. First systematic numerical experiments with uniform and non-uniform local collocation for very simple cases are expected to appear in [5].

## Appendix

Calculation of the convective term in three-dimensional Cartesian coordinates

$$\nabla \cdot (\rho \mathbf{v} \Phi) = \frac{\partial}{\partial p_x}(\rho v_x \Phi) + \frac{\partial}{\partial p_y}(\rho v_y \Phi) + \frac{\partial}{\partial p_z}(\rho v_z \Phi).$$

Calculation of the diffusive term in three-dimensional Cartesian coordinates

$$\nabla \cdot (\underline{\mathbf{D}} \cdot \nabla \Phi) =$$

$$+ \frac{\partial D_{xx}}{\partial p_x}\frac{\partial \Phi_x}{\partial p_x} + \frac{\partial D_{xy}}{\partial p_x}\frac{\partial \Phi_y}{\partial p_y} + \frac{\partial D_{xz}}{\partial p_x}\frac{\partial \Phi_z}{\partial p_z} + D_{xx}\frac{\partial^2 \Phi_x}{\partial p_x^2} + D_{xy}\frac{\partial^2 \Phi_y}{\partial p_x p_y} + D_{xz}\frac{\partial^2 \Phi_z}{\partial p_x p_z}$$

$$+ \frac{\partial D_{yx}}{\partial p_y}\frac{\partial \Phi_x}{\partial p_x} + \frac{\partial D_{yy}}{\partial p_y}\frac{\partial \Phi_y}{\partial p_y} + \frac{\partial D_{yz}}{\partial p_y}\frac{\partial \Phi_z}{\partial p_z} + D_{yx}\frac{\partial^2 \Phi_x}{\partial p_x \partial p_y} + D_{yy}\frac{\partial^2 \Phi_y}{\partial p_y^2} + D_{yz}\frac{\partial^2 \Phi_z}{\partial p_x \partial p_z}$$

$$+ \frac{\partial D_{zx}}{\partial p_z}\frac{\partial \Phi_x}{\partial p_x} + \frac{\partial D_{zy}}{\partial p_z}\frac{\partial \Phi_y}{\partial p_y} + \frac{\partial D_{zz}}{\partial p_z}\frac{\partial \Phi_z}{\partial p_z} + D_{zx}\frac{\partial^2 \Phi_x}{\partial p_x \partial p_z} + D_{zy}\frac{\partial^2 \Phi_y}{\partial p_y \partial p_z} + D_{zz}\frac{\partial^2 \Phi_z}{\partial p_z^2}.$$

## Acknowledgements

The author wishes to thank his students Igor Kovačević, Dr. Janez Perko, Dr. Miha Založnik, Robert Vertnik and Miran Dragar for assistance in preparing this article.

## References

1. Atluri S.N. *The Meshless Method (MLPG) for Domain and BIE Discretizations*. Tech Science Press, Forsyth, 2004.
2. Atluri S.N. and Shen S. *The Meshless Method*. Tech Science Press, Encino, 2002.
3. Boyd J.P. *Chebishev and Fourier Spectral Methods*. Dover, New York, 2001.
4. Buhmann M.D. *Radial Basis Function: Theory and Implementations*. Cambridge University Press, Cambridge, 2003.
5. Bulinsky Z., Nowak A. and Šarler B. (2007) Numerical experiments with the local radial basi function collocation method. *Computers, Materials, Continua*, in review, 2007.
6. Cameron A.D., Casey J.A. and Simpson G.B. *Benchmark Tests for Thermal Analysis*. NAFEMS National Agency for Finite Element Methods and Standards, Department of Trade and Industry, National Engineering Laboratory, Glasgow, 1986.
7. Chen C.S., Ganesh M., Golberg M.A. and Cheng A.H.-D. Multilevel compact radial basis functions based computational scheme for some elliptic problems. *Computers and Mathematics with Application*, 43:359–378, 2002.
8. Chen W. New RBF collocation schemes and kernel RBFs with applications. *Lecture Notes in Computational Science and Engineering*, 26:75–86, 2002.
9. Divo E. and Kassab A.J. An efficient localized RBF meshless method for fluid flow and conjugate heat transfer. *ASME Journal of Heat Transfer*, in print, 2006.
10. Fasshauer G.E. (1997) Solving partial differential equations by collocation with radial basis functions. In: Mehaute A.L., Rabut C. and Schumaker L.L. (Eds), *Surface Fitting and Multiresolution Methods*, 1997, pp. 131–138.

11. Kansa E.J. Multiquadrics – A scattered data approximation scheme with applications to computational fluid dynamics – II. Solutions to parabolic, hyperbolic and elliptic partial differential equations. *Computers and Mathematics with Application*, 19:147–161, 1990.
12. Kovačević I., Poredoš A. and Šarler B. Solving the Stefan problem by the RBFCM. *Numer. Heat Transfer, Part B: Fundamentals*, 44:575–599, 2003.
13. Kovačević I. and Šarler B. Solution of a phase-field model for dissolution of primary particles in binary aluminium alloys by an r-adaptive mesh-free method. *Materials Science and Engineering*, 413/414A:423–428, 2005.
14. Lee C.K., Liu X. and Fan S.C. Local muliquadric approximation for solving boundary value problems. *Computational Mechanics*, 30:395–409, 2003.
15. Liu G.R. *Mesh Free Methods*. CRC Press, Boca Raton, FL, 2003.
16. Liu G.R. and Gu Y.T. *An Introduction to Meshfree Methods and Their Programming*. Springer, Dordrecht, 2005.
17. Mai-Duy N. and Tran-Cong T. Numerical solution of differential equations using multiquadrics radial basis function networks. *International Journal for Numerical Methods in Engineering*, 23:1807–1829, 2001.
18. Mai-Duy N. and Tran-Cong T. Numerical solution of Navier–Stokes equations using multiquadric radial basis function networks. *Neural Networks*, 14:185–199, 2001.
19. Mai-Duy N. and Tran-Cong T. Mesh-free radial basis function network methods with domain decomposition for approximation of functions and numerical solution of Poisson's equations. *Engineering Analysis with Boundary Elements*, 26:133–156, 2002.
20. Mai-Duy N. and Tran-Cong T. Indirect RBFN method with thin plate splines for numerical solution of differential equations. *Computer Modeling in Engineering and Sciences*, 4:85–102, 2003.
21. Nayroles B., Touzot G. and Villon P. The diffuse approximation. *C.R. Acad. Sci. Paris* 313-II:293–296, 1991.
22. Özisik M.N. *Finite Difference Methods in Heat Transfer*. CRC Press, Boca Raton, FL, 1994.
23. Perko J. *Modelling of Transport Phenomena by the Diffuse Approximate Method*. Ph.D. Thesis, School of Applied Sciences, Nova Gorica Polytechnic, Nova Gorica, 2005.
24. Power H. and Barraco W.A. Comparison analysis between unsymmetric and symmetric RBFCMs for the numerical solution of PDE's. *Computers and Mathematics with Applications*, 43:551–583, 2002.
25. Sadat H. and Prax C. Application of the diffuse approximation for solving fluid flow and heat transfer problems. *International Journal of Heat and Mass Transer*, 39:214–218, 1996.
26. Shu C., Ding H. and Yeo K.S. Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 192:941–954, 2003.
27. Šarler B. and Vertnik R. Meshfree explicit local radial basis function collocation method for diffusion problems. *Computers and Mathematics with Applications*, 51:1269–1282, 2006.
28. Šarler B., Jelić N., Kovačević I., Lakner M. and Perko J. Axisymmetric multiquadrics. *Engineering Analysis with Boundary Elements*, 30:137–142, 2006.
29. Šarler B. A radial basis function collocation approach in computational fluid dynamics. *Computer Modeling in Engineering and Sciences*, 7:185–193, 2005.

30. Šarler B., Vertnik R. and Perko J. Application of diffuse approximate method in convective-diffusive solidification problems. *Computers, Materials, Continua*, 2(1):77–84, 2005.
31. Šarler B., Perko J. and Chen C.S. Radial basis function collocation method solution of natural convection in porous media. *International Journal of Numerical Methods for Heat and Fluid Flow*, 14:187–212, 2004.
32. Šarler B. Towards a mesh-free computation of transport phenomena. *Engineering Analysis with Boundary Elements*, 26:731–738, 2002.
33. Šarler B., Kovačević I. and Chen C.S. A mesh-free solution of temperature in direct-chill cast slabs and billets. In: Mammoli A.A. and Brebbia C.A. (Eds), *Moving Boundaries VI*. WIT Press, Southampton, 2004, pp. 271–280.
34. Tolstykh A.I. and Shirobokov D.A. On using radial basis functions in a "finite difference" mode with applications to elasticity problems. *Computational Mechanics*, 33:68–79, 2003.
35. Versteeg H.K. and Malalasekera W. *Computational Fluid Dynamics: The Finite Volume Method*. Prentice Hall, Harlow, 1995.
36. Vertnik R. and Šarler B. Meshless local radial basis function collocation method for convective-diffusive solid-liquid phase change problems. *International Journal of Numerical Methods for Heat and Fluid Flow* (Special Issue: European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS 2004, Jyväskylä, 24–28 July 2004), 16:617–640, 2005.
37. Vertnik R., Perko J. and Šarler B. Solution of temperature field in DC cast aluminium alloy billet by the Diffuse Approximate Method. *Materials and Technology*, 38:257–261, 2004.
38. Vertnik R., Založnik M. and Šarler B. Solution of transient direct chill aluminium billet casting problem with simultaneous material and interphase moving boundaries. *Engineering Analysis with Boundary Elements*, 30: 847–855, 2006.
39. Wang J.G. and Liu G.R. On the optimal shape parameter of radial basis functions used for 2-D meshless method. *Computer Methods in Applied Mechanics and Engineering*, 26:2611–2630, 2002.
40. Wrobel L.C. *The Boundary Element Method – Volume 1: Applications in Thermofluids and Acoustics*. Wiley, New York, 2001.
41. Zienkiewicz O.C. and Taylor R.L. *The Finite Element Method*. Butterworth-Heinemann, Oxford, 2002.