 CMES, vol.19, no.1, pp.55-68, 2007

Weight Function Shape Parameter Optimization in Meshless Methods for Non-uniform Grids

J. Perko[1] **and B. Šarler**[2]

Abstract: This work introduces a procedure for automated determination of weight function free parameters in moving least squares (MLS) based meshless methods for non-uniform grids. The meshless method used in present work is Diffuse Approximate Method (DAM). The DAM is structured in 2D with the one or two parameter Gaussian weigh function, 6 polynomial basis and 9 noded domain of influence. The procedure consists of three main elements. The first is definition of the reference quality function which measures the difference between the MLS approximation on non-uniform and hypothetic uniform node arrangements. The second is the construction of the object function from the reference quality function which has to be minimized for optimum performance of the method on non-uniform node arrangement. The third is the optimization procedure for obtaining the minimum of the object function. The main idea of this paper is demonstrated on solution of the transient Burgers equation on stretched non-uniform grid and three types of random non-uniform grids. The inverse Gauss function is used as a reference quality function, object function is built from the second partial derivatives, and the k-ary heap like tree procedure is used for optimization. A substantial improvement of the accuracy of the method is achieved with the locally optimized values of the weight function compared to the fixed value that was exclusively used in previous DAM literature. With statistical comparison it was shown that in addition to improvement of accuracy also the stability of simulation is substantially improved.

[1] Belgian Nuclear Research Centre SCK·CEN, Mol, Belgium

[2] Laboratory for Multiphase Processes, University of Nova Gorica, Slovenia

Keyword: Optimization, shape parameters, meshless methods, Diffuse Approximate Method.

1 Introduction

The meshless numerical methods represent the most progressively developing field of computational science in the last years. The fast development is stimulated by their great geometrical flexibility and ease of grid generation, which is of utmost importance in computation of practical engineering and scientific problems. The complex three dimensional geometries are very common in real problems and a lot of manpower is needed for their spatial discretization (polygonization) in traditional numerical methods. Meshless methods, on the other hand, are characterized by the absence of polygonization and are thus very appealing for the use in problems with complex geometries or, for example, moving and/or free boundary problems.

The polygonization in meshless methods is avoided by the use of meshless shape functions. The shape function is constructed from a set of nodal values of the field variables. However, the construction of the most appropriate shape function is still an open issue. Shape function may be constructed in a variety of ways, see [Atluri (2004), Atluri and Shen (2002)]. They can be constructed for example by rational Shepard functions [Shepard (1968)], polynomials or radial basis functions (RBF). Coefficients of the shape function can be determined by the collocation [Kansa (1990)], by the least squares approximation or by the quasi-interpolation [Pollandt (1997)]. In this paper the convective-diffusive PDE is solved by the DAM [Nayroles, Touzot, and Villon (1991)]. The shape function in DAM is constructed by the polynomials and the de-

termination of the coefficients follows the least squares approximation. In the least squares approximation, the relative importance of the neighboring nodes to a calculated node is determined by the weight function. The weight function can contain one or more shape parameters, depending on its type. The choice of shape parameter values are often left to either the developers or the end-users and is based on their experience. Finding the best values may be as cumbersome as the mesh generation in traditional methods. In addition, shape parameters may vary from one subdomain to another. This is particularly important in non-uniform grids. Setting the optimal values of the shape parameters represents almost unfeasible task even for an experienced developer or end-user. In general, from the theoretical point of view, the non-uniform grid arrangements represent quite unresearched area and in the literature, most of the calculations using meshless methods are still performed on uniform grids. So far the research on this field was based on experimental determination [Perko, Šarler, and Rek (2000)], statistical evaluation [Golberg, Chen, and Karur (1996)] in Dual Reciprocity Method (DRM) and theoretical determination of shape parameter for example in global interpolation Radial Basis Function Collocation Method [Kansa and Hon (2000)]. Recently the solution of finding the optimal radius of the support for 4^{th} order spline weight function was elaborated in [Nie, Atluri, and Zou (2006)]. The idea for automated optimization of weight function shape parameters is implemented to the Moving Least Squares (MLS) based DAM with the Gaussian weight function of the form $\exp(-c\,r)$, where r is an Euclidian distance between nodes, and c is a shape parameter.

The scaling parameter c is usually heuristically defined by authors and depends mostly on the grid density and on the distribution of nodes. There is a range of different values used by different authors. Focusing on DAM, heuristically defined c parameters range from $\ln 100 \approx 4.6$ [Prax, Salagnac, and Sadat (1998)] to 6.25 found in [Belytscko, Krongauz, Fleming, Organ, and Liu (1996)] or almost 7 [Sadat and Couturier (2000)]. In case of randomly distributed nodes, however, the appropriate shape parameter value can vary substantially from one set of nodes to another.

It should be noted that not only optimization of shape parameter, but also a good searching algorithm for nodes in the domain of influence plays a key role in stability and accuracy of meshless methods. All simulations presented in this paper use the simplest search algorithm based on the nearest nodes searching only.

2 Problem description

Consider a domain Ω with boundary Γ both containing N calculation nodes. The representation of the field variables in the problem domain is made via construction of the shape function, which for the local approximation meshless methods is formed from I neighboring nodes in data sites $\boldsymbol{p}_i$ on the $n'th$ subdomain ${}_n\Omega \in \Omega + \Gamma$. The neighboring nodes can be chosen based on two principles. In the first principle, the total number of neighboring nodes I is set. In the second principle, the shape and the size of the subdomain ${}_n\Omega$ is set. The shape is usually chosen to be a sphere (3D) or a circle (2D) with the radius (size) σ centered around the node $\boldsymbol{p}_n$. In former case ${}_n\Omega$ is denoted as the domain of influence, while in latter case ${}_n\Omega$ is named the support [Liu (2003)]. The first strategy, preferable for strongly non-uniform node arrangements, is used in present work.

In this work we focus on the DAM developed by [Nayroles, Touzot, and Villon (1991)] which is based on the MLS local approximation technique using polynomials as basis functions and Gaussian weight function. This is a very promising meshless numerical method due to its simplicity and ability to solve complicated large-scale problems [Couturier and Sadat (1999), Sadat and Couturier (2000), Sophy and Sadat (2002), Šarler, Vertnik, and Perko (2004), Šarler, Vertnik, and Perko (2005), Perko (2005)]. In DAM, the function value and the corresponding derivatives of a variable ϕ in the reference node $\boldsymbol{p}_n$ are calculated from K basis functions φ_k as

$$\phi(\boldsymbol{p}) \approx \sum_{k=1}^{K} \alpha_k(\boldsymbol{p})\,\varphi_k(\|\boldsymbol{p}-\boldsymbol{p}_n\|), \qquad \boldsymbol{p} \in_n \Omega, \quad (1)$$

with derivatives, calculated as

$$\frac{\partial \phi(\boldsymbol{p})}{\partial \xi} \approx \sum_{k=1}^{K} \alpha_k(\boldsymbol{p}) \frac{\partial \varphi_k(\|\boldsymbol{p}-\boldsymbol{p}_n\|)}{\partial \xi}; \quad \xi \in \{p_x, p_y\} \quad (2)$$

$$\frac{\partial^2 \phi(\boldsymbol{p})}{\partial \xi\, \partial \zeta} \approx \sum_{k=1}^{K} \alpha_k(\boldsymbol{p}) \frac{\partial^2 \varphi_k(\|\boldsymbol{p}-\boldsymbol{p}_n\|)}{\partial \xi\, \partial \zeta}; \quad \xi, \zeta \in \{p_x, p_y\} \quad (3)$$

$\|\boldsymbol{p}-\boldsymbol{p}_n\|$ is Euclidian distance between nodes $\boldsymbol{p}$ and $\boldsymbol{p}_n$ where $\boldsymbol{p} = \boldsymbol{i}_x p_x + \boldsymbol{i}_y p_y$ is introduced in Cartesian coordinate system with base vectors $\boldsymbol{i}_x$ and $\boldsymbol{i}_y$. Functions φ_k are chosen as polynomials $\varphi_1 = 1; \varphi_2 = (p_x - p_{xn})/\sigma; \varphi_3 = (p_y - p_{yn})/\sigma; \varphi_4 = (p_x - p_{xn})^2/\sigma^2; \varphi_5 = (p_x - p_{xn})(p_y - p_{yn})/\sigma^2; \varphi_6 = (p_y - p_{yn})^2/\sigma^2$, i.e. $K = 6$.

In case of least squares approximation, coefficients α_k are obtained from the solution of the system of equations

$$\mathbf{A}\,\boldsymbol{\alpha} = \mathbf{b}. \quad (4)$$

where the matrix $\mathbf{A}$ and the vector $\mathbf{b}$ are obtained through the minimization of the functional

$$\mathscr{I}(\alpha_k(\boldsymbol{p})) = \sum_{i=1}^{I=9} \widehat{W}(\|\boldsymbol{p}-\boldsymbol{p}_i\|) \sum_{k=1}^{K} [\alpha_k(\boldsymbol{p})\, \varphi_k(\|\boldsymbol{p}-\boldsymbol{p}_i\|) - \phi(\boldsymbol{p}_i)]^2, \quad (5)$$

which leads to the system of $K \times K$ equations for the calculation of the unknown coefficients α at each node $\boldsymbol{p}$ in the subdomain ${}_n\Omega$. The number of nodes in the domain of influence I should be larger or equal to K, thus $I \geq K$ and is in our case fixed to 9. The influence of the neighboring nodes to the reference node is expressed in terms of weight function $\widehat{W}$ which defines the conditioning of matrix $\mathbf{A}$ and subsequent accuracy and stability of the simulation. The left-hand side matrix $\mathbf{A}$ and right-hand side vector $\mathbf{b}$ are written in explicit form for n'th subdomain as

$${}_nA_{jk} = \sum_{i=1}^{I} \varphi_k(\boldsymbol{p}_i - \boldsymbol{p})\, {}_n\widehat{W}(\boldsymbol{p}_i - \boldsymbol{p})\, \varphi_j(\boldsymbol{p}_i - \boldsymbol{p}), \quad (6)$$

while right-hand side vector ${}_nb_j$ is

$${}_nb_j = \sum_{i=1}^{I} \varphi_k(\boldsymbol{p}_i - \boldsymbol{p})\, {}_n\widehat{W}(\boldsymbol{p}_i - \boldsymbol{p})\, \phi(\boldsymbol{p}_i). \quad (7)$$

Gaussian weight function is defined as

$$\widehat{W}(\boldsymbol{p}-\boldsymbol{p}_n) = \begin{cases} e^{-c\frac{\|\boldsymbol{p}-\boldsymbol{p}_n\|^2}{\sigma^2}} & ; \|\boldsymbol{p}-\boldsymbol{p}_n\| \leq \sigma \\ 0 & ; \|\boldsymbol{p}-\boldsymbol{p}_n\| > \sigma. \end{cases}, \quad (8)$$

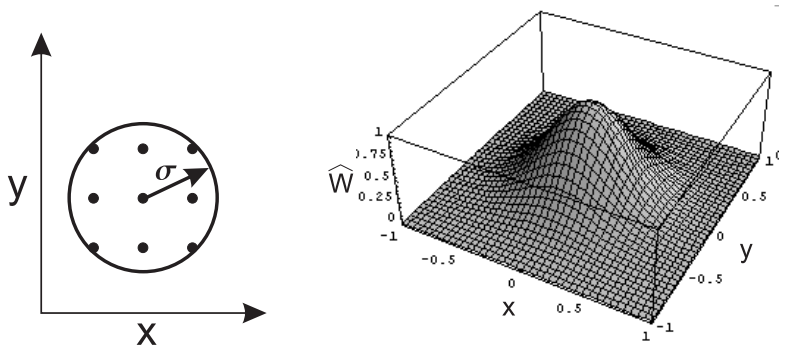
Gaussian weight function in equation (8) is characterized by the shape parameter c. This shape parameter, in general, may vary from one subdomain to another. In addition the shape parameter can be chosen differently in different directions. For example consider a 2D problem with 9 support nodes. In Figure 1a uniform distribution is presented. The most appropriate weight function for this case is intuitively radially symmetric, as shown in Figure 1b.

In Figure 2a coordinate direction dependent uniform distribution of nodes is shown where grid spacing is different in x and y directions. To obtain better conditioning of matrix and consequent more accurate spatial derivatives, the weight function for this case should be different in each direction. In other words, the parameter c have to be larger in x direction as in y direction as presented in Figure 2b. An extended formulation of the weight function for the described case can be written as

$$\widehat{W}(\boldsymbol{p}-\boldsymbol{p}_n) = \begin{cases} e^{-c_x\frac{(p_x-p_{xn})^2}{\sigma^2} - c_y\frac{(p_y-p_{yn})^2}{\sigma^2}} & ; \|\boldsymbol{p}-\boldsymbol{p}_n\| \leq \sigma \\ 0 & ; \|\boldsymbol{p}-\boldsymbol{p}_n\| > \sigma. \end{cases}, \quad (9)$$

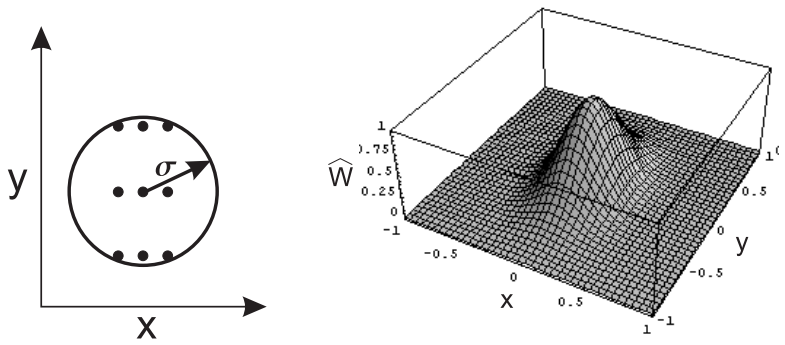
3 Solution procedure

The main problem of proper numerical implementation of DAM is the local selection of shape parameter(s) in weight function in particular where non-uniform grids are present. We introduce the following procedure for determination of corresponding shape parameter(s). The procedure consists of three main steps; First, the



(a) (b)

Figure 1: Uniform distribution. (a) Distribution of nodes, (b) Weight function



(a) (b)

Figure 2: Uniform non-equidistant distribution. (a) Distribution of nodes, (b) Weight function

measure against which the MLS approximation is compared to has to be defined. This measure is called the *reference quality function*. The reference quality function is used as a basis for evaluation of numerical derivatives on uniform and non-uniform grids. Second, a function called the *object function*, which needs to be minimized, need to be constructed. The object function is constructed from the spatial derivatives of the reference quality function on the hypothetic uniform and non-uniform grid used in the problem discretization. And finally, the object function minimum is obtained by the use of the appropriate *optimization procedure*.

Reference quality function: The reference quality function $\mathscr{F}$ is defined on each subdomain ${}_n\Omega$. The reference quality function should not be any of the used basis functions because in this case the numerically calculated spatial derivatives always coincide with the analytical spatial derivatives. In this work we define $\mathscr{F}$ in such a way that $\mathscr{F}(\boldsymbol{p}-\boldsymbol{p}_n) = 1/\widehat{W}(\boldsymbol{p}-\boldsymbol{p}_n)$ inside ${}_n\Omega$ and 0 outside ${}_n\Omega$. Explicitly, the reference quality function is

$$\mathscr{F}(\boldsymbol{p}-\boldsymbol{p}_n) = \begin{cases} e^{c_x\frac{(p_x-p_{xn})^2}{\sigma^2}+c_y\frac{(p_y-p_{yn})^2}{\sigma^2}} & ;\|\boldsymbol{p}-\boldsymbol{p}_n\| \le \sigma \\ 0 & ;\|\boldsymbol{p}-\boldsymbol{p}_n\| > \sigma \end{cases}. \tag{10}$$

The reference quality function (10) is chosen because the right-hand side values cancel and the expression (7) simplifies to

$$\begin{aligned} {}_nb_j &= \sum_{i=1}^{I} \varphi_j(\boldsymbol{p}_i-\boldsymbol{p}_n)\,{}_n\widehat{W}(\boldsymbol{p}_i-\boldsymbol{p}_n)\cdot\frac{1}{{}_n\widehat{W}(\boldsymbol{p}_i-\boldsymbol{p}_n)} \\ &= \sum_{i=1}^{I}\varphi_j(\boldsymbol{p}_i-\boldsymbol{p}_n). \end{aligned} \tag{11}$$

The domain of influence usually contains a relatively low number of nodes and the chosen refer-

ence quality function is relatively steep. Therefore the difference between the analytical and the numerical partial derivatives might be very different and the second partial derivatives of $\mathscr{F}$ calculated on a given non-uniform grid should not be compared to the analytically calculated partial derivatives. For this reason our second partial derivatives are obtained via least squares approximation of the reference quality function calculated on a uniform grid. The approximation of the reference quality function on the uniform grid is denoted by $\mathscr{F}_{uni}$, while the approximation of the reference quality function calculated on the posed irregular grid is marked by $\mathscr{F}_{irr}$. Both approximations of the functions and the partial derivatives are obtained from equations (1-3). The dependence on grid density is alleviated by scaling of Euclidian distance between nodes in the basis functions by the radius of influence in each subdomain ${}_n\Omega$. Coefficients α are thus calculated from (4) by the least square approximation.

The approximation of the reference quality function on the uniform grid $\mathscr{F}_{uni}$ is calculated with $c_x = c_y = 12.5$ on the grid shown in Figure 3(left). This value is chosen because it gave the most optimal results on all grids tested.

The evaluation of the second derivatives on the uniform grid gives the constant value $\frac{\partial^2 \mathscr{F}_{uni}}{\partial p_x^2} = \frac{\partial^2 \mathscr{F}_{uni}}{\partial p_y^2} = 6400$.

Object function: Object function $\mathscr{J}$ is a function which needs to be minimized. The object function proposed in this work seeks for the minimum difference between the second partial derivatives of the reference quality function calculated on the uniform grid (Figure 3(left)) and the second partial derivatives of the same function found on the real (non-uniform) grid (Figure 3(right)). Thus we seek for $min\{\mathscr{J}(\Phi_{uni}, \Phi_{irr})\}$, where Φ_{uni} and Φ_{irr} stand for the values or n'th derivatives of $\mathscr{F}_{uni}$ and $\mathscr{F}_{irr}$, respectively. Therefore if the difference between derivatives of the known reference quality function and reproduced shape function are minimal, then the solution of PDE will be the most accurate.

The selection of the object function has a strong influence on optimization results. Finding the most appropriate one is not an easy task. Therefore our object function is constructed from two parts. The first one is based on the assumption that the most accurate derivatives are obtained on a locally uniform grid. All results obtained on arbitrarily arranged grid nodes are then compared to the solution on the uniform grid. The second condition which should be satisfied is that second derivatives with respect to shape parameter in the x and y directions are equal. Because errors of the second derivatives are usually larger, our object function is built only using second derivatives as follows

$$\begin{aligned} \mathscr{J} &= \left(\frac{\partial^2 \mathscr{F}_{irr}}{\partial p_x^2} - \frac{\partial^2 \mathscr{F}_{uni}}{\partial p_x^2}\omega_x^4\right)^2 \\ &+ \left(\frac{\partial^2 \mathscr{F}_{irr}}{\partial p_y^2} - \frac{\partial^2 \mathscr{F}_{uni}}{\partial p_y^2}\omega_y^4\right)^2 \\ &+ 16\left(\omega_x^2+\omega_y^2\right)^2 \cdot \left(\frac{\partial^2 \mathscr{F}_{irr}}{c_x \partial p_x^2} - \frac{\partial^2 \mathscr{F}_{irr}}{c_y \partial p_y^2}\right)^2. \end{aligned} \tag{12}$$

Object function forcing parameters ω_x and ω_y are introduced to prevent the shape parameters from being too different and too extreme and are defined as $\omega_x = c_y/c_x$ and $\omega_y = c_x/c_y$. The first two terms in equation (12) are used to minimize the error between the second derivatives obtained on the hypothetic uniform grid and the second derivatives obtained on the arbitrary grid. The third term in equation (12), however, is used to force the second derivatives to be equal with respect to c_x and c_y. Equivalence of the second derivatives is forced more intensively when c_x and c_y are further apart. The constant 16 is set as a weight between the first two terms and the last term. For one parametric formulation, equation (12) simplifies to

$$\begin{aligned} \mathscr{J} &= \left(\frac{\partial^2 \mathscr{F}_{irr}}{\partial p_x^2} - \frac{\partial^2 \mathscr{F}_{uni}}{\partial p_x^2}\right)^2 \\ &+ \left(\frac{\partial^2 \mathscr{F}_{irr}}{\partial p_y^2} - \frac{\partial^2 \mathscr{F}_{uni}}{\partial p_y^2}\right)^2 \\ &+ 64\left(\frac{\partial^2 \mathscr{F}_{irr}}{c\,\partial p_x^2} - \frac{\partial^2 \mathscr{F}_{irr}}{c\,\partial p_y^2}\right)^2. \end{aligned} \tag{13}$$

The goal is to minimize the object function (13)

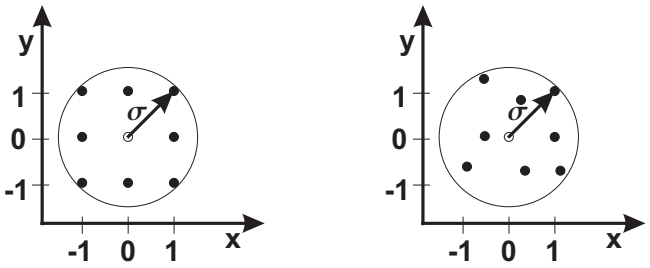


Figure 3: Scaled node distribution ($\sigma = \sqrt{2}$): uniform (left), non-uniform (right).

for one parametric weight function or (12) for two parametric weight function.

Optimization procedure: The optimization procedure can be chosen by the user. There are many algorithms available [Fortran (2003)]. In cases where $\mathscr{J}$ consists of more than one local minima, special precaution and/or special optimization procedures should be used. The choice of optimization procedure also has a great influence on the optimization time. For testing purposes a simple but robust k-ary heap procedure with fixed three children solutions is used:

1. Set the initial value of shape parameter c_0.
2. Search for minimum around initial value with large step λ_1: $c^1_{min} = c^0 \pm \lambda_1$.
3. Search for minimum around c^1_{min} with step λ_2: $c^2_{min} = c^1_{min} \pm \lambda_2$; $\quad \lambda_2 = \lambda_1 \varpi/2$.
4. Refine search around c^2_{min}: $\quad c^3_{min} = c^2_{min} \pm \lambda_3$; $\quad \lambda_3 = \lambda_2 \varpi/2$.
5. Update new minimum ($c^2_{min} = c^3_{min}$).
6. Return to Step 4 if $c^3_{min} < c^2_{min}$.
7. Exit when condition in Step 6 is not satisfied.

λ_1 is a user-defined step and $\varpi \approx 0.618$ is the golden section value. One parametric and two parametric weight functions are used in optimization procedure in this work. In the latter case the weight function (9) has two shape parameters, therefore shape parameters c and λ are second-order vectors. The starting position is set to $\mathbf{c}_0 = \{15, 15\}$ and initial step $\boldsymbol{\lambda}_1 = \{10, 10\}$. It is possible to expand the procedure to four degrees of freedom for optimization of each weight function quadrant in possible future work.

4 Numerical examples

The procedure to find optimum shape parameter of the weight function proposed above is tested on a convective-diffusive PDE using Dirichlet boundary conditions.

4.1 Problem description

The optimization procedure is tested on the 2D Burger PDE which describes the wave propagation in 2D domain. The Burger PDE is chosen because it contains first and second order derivatives and is time dependent which is important for the calculation of real convection-diffusion physical problems. The second important feature of the Burger equation is that the analytical solution is known. The dimensionless time period considered for calculation is t_{max}=4. In this time the "wave" propagates through the whole domain, stretching from 0 to 1 in x and y direction, which has two effects. The first is that each node is subjected to the strong nonlinearity of the propagating wave in a certain time. The second is that errors from one time step to another in an iterative process are accumulated. The Burger PDE is defined as

$$\frac{\partial \phi}{\partial t} + \phi \left(\frac{\partial \phi}{\partial p_x} + \frac{\partial \phi}{\partial p_y} \right) = \nu \left(\frac{\partial^2 \phi}{\partial p_x^2} + \frac{\partial^2 \phi}{\partial p_y^2} \right); \quad \{p_x, p_y\} \in \Omega \quad (14)$$

where $\nu = 0.05$ is the dimensionless diffusion constant. The analytical solution is

$$\phi(p_x, p_y, t) = \frac{1}{1 + \exp\left(\frac{p_x + p_y - t}{2\nu} \right)}. \quad (15)$$

Boundary conditions are Dirichlet boundary conditions with values from (15) at the boundary.

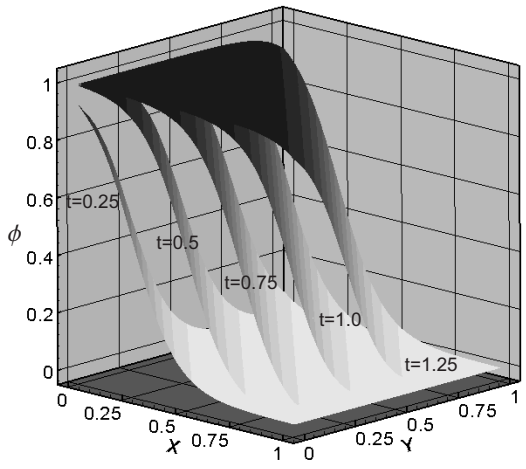


Figure 4: Analytical solution of the Burger equation at dimensionless times 0.25, 0.5, 0.75, 1, and 1.25.

At initial time $t_0 = 0$ values for ϕ are taken from the analytical solution, thus

$$\phi(p_x, p_y, 0) = \frac{1}{1 + \exp\left(\frac{p_x + p_y}{2\mu}\right)}. \tag{16}$$

From this initial condition equation (14) is temporarily discretized in an explicit manner. Variable $\phi(p_x, p_y)$ at time $t^{\tau+1}$ is calculated explicitly from variable ϕ at time t^{τ} as

$$\phi^{\tau+1} = \phi^{\tau} + \Delta t \left(\nu \left(\frac{\partial^2 \phi^{\tau}}{\partial p_x^2} + \frac{\partial^2 \phi^{\tau}}{\partial p_y^2} \right) - \phi^{\tau} \left(\frac{\partial \phi^{\tau}}{\partial p_x} + \frac{\partial \phi^{\tau}}{\partial p_y} \right) \right), \tag{17}$$

where the derivatives are calculated as described in equations (1-3). Time step Δt is 0.001 in all numerical examples.

4.2 Grids used

The solution of the Burger equation is sought on different node distributions. First, the tests are carried out on random node distribution, and later on grids with finer discretization at the boundaries. As proposed in [Lazzaro and Montefusco (2002)] we define two quantities to measure the density of the data set: separation distance δ_{sep} which is the half distance between the closest pair of nodes in the data set, and fill distance δ_{fill} which gives the radius of the largest inner empty space. Random grids are constructed here in a way that nodes are initially distributed uniformly with fill distance $\delta_{fill,uni}$. From this initial position each node is displaced randomly by δ_x in the x direction and δ_y in the y direction. Displacement is described by $\delta = 0\%$ for no displacement ($\delta_{fill} = \delta_{sep}$), i.e. uniform grid and $\delta = 50\%$ for maximal displacement to half of the initial distance $\delta_{fill,uni}/2$ between the original neighboring nodes ($\delta_{sep} = 0$, $\delta_{fill} = \delta_{fill,uni}/2$). In latter case the nodes can coincide at the same position. Grids with 30×30 domain nodes for $\delta = 0(\%)$ and $\delta = 45\%$ are shown in Figure 5 and Figure 6, respectively. For the sake of comparison the same seed number is used in the random generator (Fortran RANDOM_NUMBER routine) in all cases [Fortran (2003)]. Random grids are not used very frequently in practice. Usually grids are regular, but still non-uniform as presented in Figure 7. An example of such grid with finer node distribution near the boundary for improving calculation accuracy and for better representation of boundary layers is described for example in [Sadat and Couturier (2000)] and here expanded for variable stretching with

$$\boldsymbol{p} = \boldsymbol{p}_{min} + \frac{\boldsymbol{p}_{max} - \boldsymbol{p}_{min}}{2} \times \left(1 + \frac{\tanh\left(\frac{2\chi \boldsymbol{p}_r - \frac{\boldsymbol{p}_{max} + \boldsymbol{p}_{min}}{2}}{\boldsymbol{p}_{max} - \boldsymbol{p}_{min}} \right)}{\tanh \chi} \right), \tag{18}$$

where $\boldsymbol{p}_r$ corresponds to the position of the nodes on an $N \times N$ uniform grid, χ is stretching factor and $\boldsymbol{p}_{max}$ and $\boldsymbol{p}_{min}$ stand for maximum and minimum positions in the domain Ω, respectively.

Tests are carried out on four different random grids, each defined with theoretical δ parameter: $\delta = 0\%$, $\delta = 25\%$, $\delta = 35\%$, and $\delta = 45\%$ and one stretched non-uniform grid shown in Figure 7. Errors are calculated in terms of maximal relative error between analytical and numerical solution

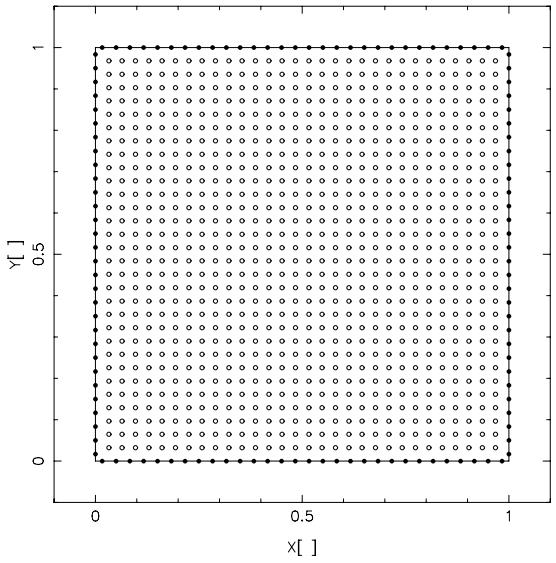


Figure 5: 30×30 with $\delta = 0\%$ (uniform) grid.

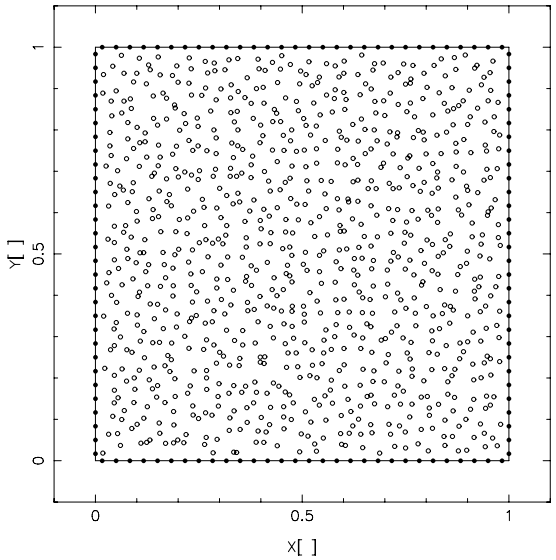


Figure 6: 30×30 with $\delta = 45\%$ random grid.

over the whole domain in each time step

$$\eta(\tau) = \max\{\frac{|\phi^{\tau}_{analytic} - \phi^{\tau}_{numeric}|}{\phi^{\tau}_{analytic}}\}, \qquad (19)$$

where $\tau \in \{0, t_{max}\}$ is time, and $n = 1, ..., N$.

Example 1: Uniform grid The results produced on uniform grids are quite insensitive on a wide range of parameter c (roughly from 10-30) although the error decreases with increasing c as

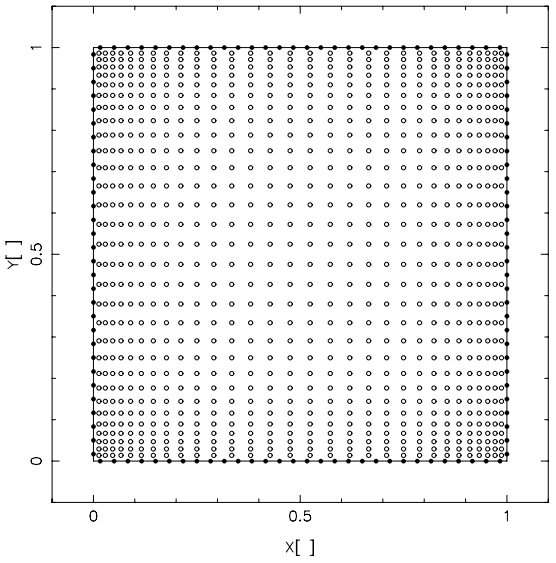


Figure 7: 30×30 with stretching factor 1.3 grid.

shown in Figure 8. The results in terms of $\eta(t)$ are presented for four cases of fixed c parameters and for optimized parameters c_x and c_y. Optimized results are inside this range with value 12.5 as set for the reference solution. Naturally, in this case c_x and c_y are equal after the optimization. Therefore both one parametric and two parametric optimizations gave the same results.

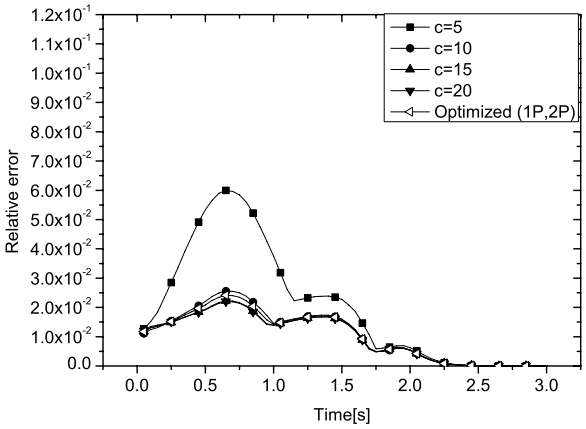


Figure 8: Uniform grid, i.e. 30×30 with $\delta = 0\%$ grid.

Example 2: Slightly non-uniform random grid ($\delta = 25\%$) Nodes are displaced from the initial

uniform distribution by up to 25% in this case. $\eta(t)$ is roughly twice the error on the uniform grid. Still, one can observe the same tendency of lower errors for larger c parameter values (Figure 9). Again optimized results are very close to the best fixed value of c. The variation of parameter c_x and c_y values is small, since the grid is not strongly non-uniform. However, the results obtained by two parametric (denoted by Optimized 2P) weight functions give slightly lower error than the results obtained by one parametric (denoted by Optimized 1P) weight functions.

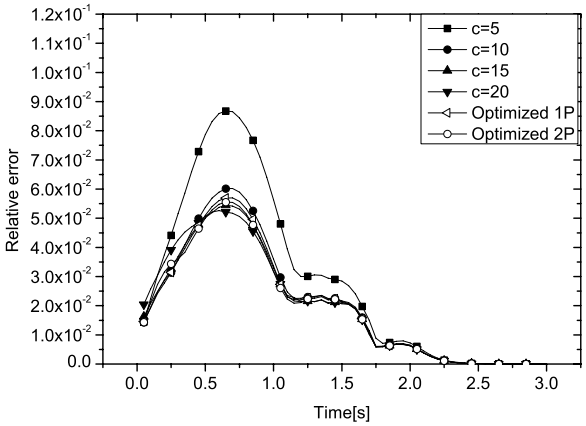


Figure 9: 30×30 with random $\delta = 25\%$ grid.

Example 3: Non-uniform random grid ($\delta = 35\%$) Behavior $\eta(t)$ in Figure 10 indicates that the spectrum of fixed shape parameter values with converged numerical solution is drastically narrowed and the best shape parameter value is around 10. At this point the optimized solution is better than any other solution obtained by the fixed shape parameter value.

Example 4: Extremely non-uniform random grid ($\delta = 45\%$) The range of valid shape parameters at $\delta = 45\%$ is further narrowed to values between 9 and 10 (Figure 11) (Note that curves are now marked with the same signs for different c values). Therefore, in real cases, when dealing with strongly unstructured grids, it is quite difficult to determine the most appropriate shape parameter value. Again, the optimized solution gave

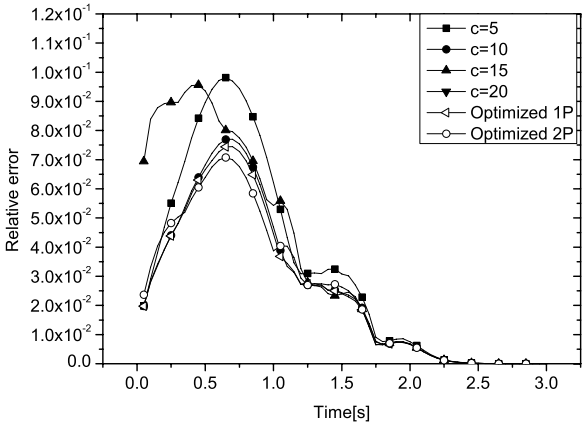


Figure 10: 30×30 with random $\delta = 35\%$ grid.

better results than any other solution obtained by unique shape parameter value. The range of shape parameters c_x and c_y in this case is between 5 and 20 and the variation of values in different subdomains is large.

For more scattered grid nodes the solution can not be obtained by any shape parameter value. With the optimization proposed in this work the stability of the solution is provided also for larger displacements for the same seed number.

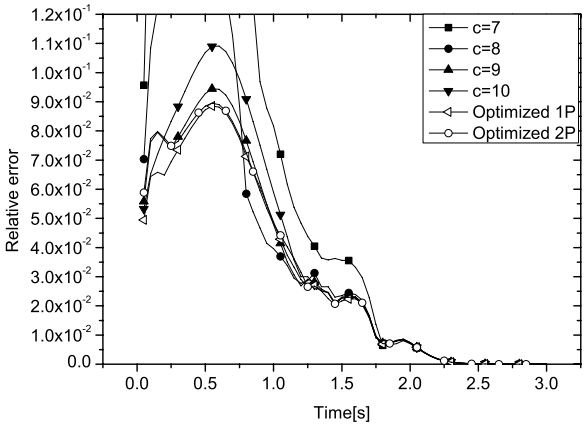


Figure 11: 30×30 with random $\delta = 45\%$ grid.

Maximal relative errors for three different grid densities are presented in Tables 1-3. Tabulated results are given for larger spectra of shape parameters and for different grid densities, 20×20, 30×30, and 40×40, respectively. Errors marked bold are the smallest.

Table 1: Maximal relative errors (in %) on 20×20 random grid as a function of fixed c, one parametric (1P) and two parametric (2P) optimization.

c	0%	25%	35%	45%
5	16.210	24.015	27.831	26.643
7	11.015	18.540	23.032	**23.885**
8	9.665	16.936	21.570	-
9	8.806	15.797	20.486	-
10	8.266	14.957	19.632	-
11	7.930	14.301	**19.454**	-
15	7.474	13.320	24.771	-
20	**7.419**	**12.553**	-	-
1P	7.631	14.444	21.023	22.583
2P	7.631	14.032	20.398	22.358

The tendency of shape parameter c to be smaller for a larger degree of non-uniformity is clearly seen. The solution in this case is obtained for the fixed shape parameter value c ranging between 5 and 7, which is the range given in the literature. Similar behavior is observed for the finer 30×30 grid in Table 2.

Table 2: Maximal relative errors (in %) on 30×30 random grid as a function of fixed c, one parametric (1P) and two parametric (2P) optimization.

c	0%	25%	35%	45%
5	5.997	8.680	9.817	-
7	3.739	6.642	8.373	34.310
8	3.157	6.602	8.056	13.580
9	2.790	6.274	7.852	**9.442**
10	2.557	6.021	**7.530**	10.910
11	2.421	5.856	7.639	-
15	2.227	5.444	9.560	-
20	**2.202**	**5.215**	-	-
1P	2.299	5.708	7.462	8.862
2P	2.299	5.555	7.256	8.952

The best stability is assured for c around 10. For the even finer 40×40 grid the range of valid parameters is narrowed down. Here results are presented only for up to $\delta = 35\%$, because for $\delta = 45\%$ the simulation diverged for all cases due to the larger probability of achieving the given theoretical limit δ_{sep}.

Table 3: Maximal relative errors (in %) on 40×40 random grid as a function of fixed c, one parametric (1P) and two parametric (2P) optimization.

c	0%	25%	35%
5	2.459	3.895	3.916
7	1.237	2.779	**3.135**
8	1.189	2.474	**3.135**
9	1.161	2.275	3.194
10	1.145	2.153	3.366
11	1.136	**2.084**	-
15	**1.127**	2.094	-
20	1.144	2.172	-
1P	1.138	1.976	3.034
2P	1.138	1.891	3.418

Example 5: Stretched non-uniform grid ($\chi = 1.3$) This more realistic node arrangement can be found more frequently in practice. Similar refined non-uniform grids are obtained for example with Delaney triangulation in Finite Element Method (FEM), weith remeshing of grids, etc. Errors are again smaller for larger values of c as observed in Example 1 and Example 2, but for larger values than 10 the simulation diverges (Figure 12), which is similar to critical values in Example 3 and Example 4. The variation of parameters can be presented graphically. Figure 13 represents the spatial distribution of shape parameters c_x and c_y.

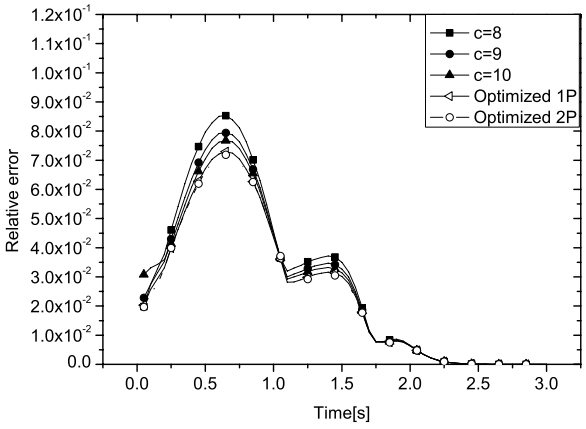


Figure 12: Regular stretched non-uniform 30×30 grid with stretching factor $\chi = 1.3$.

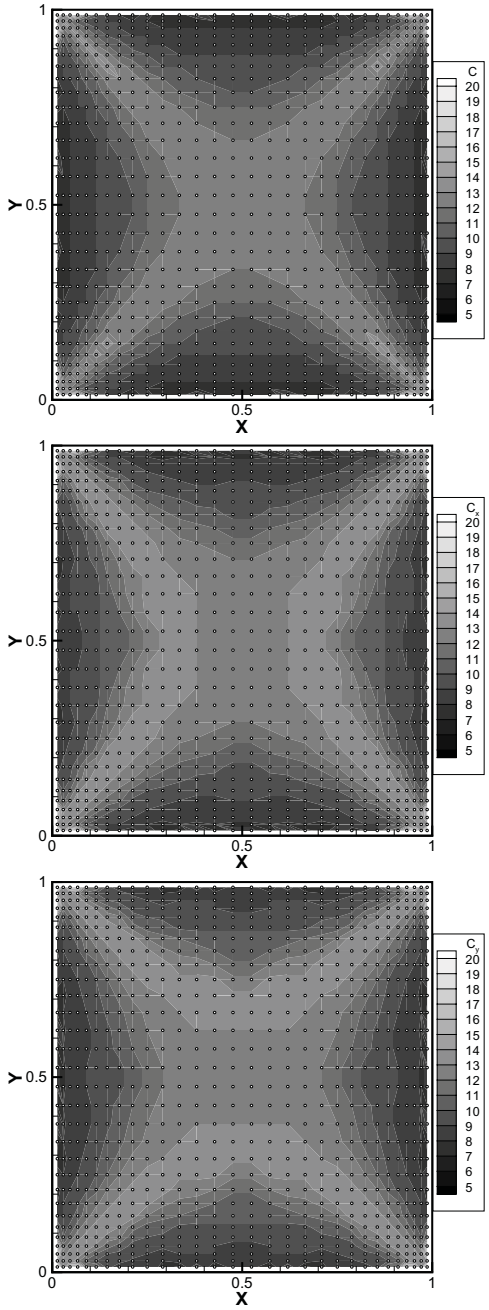


Figure 13: Shape parameter values c for one parametric optimization (top), and two parametric c_x (center) and c_y (bottom).

The parameters are symmetrical with respect to the x and y axis and range from 7.5 to 18.5. Similar results are obtained by using one parametric optimization.

Statistical comparison: For the sake of comparison, the results in Tables 1-3 are calculated on the node distribution with the same seed number, which determines the sequence of quasi-random numbers. As a consequence the node distribution is the same, only the displacement is bigger for larger δ values. For real comparison several runs have to be performed with random seed number and same δ parameter. Naturally maximal overall error will vary from one node distribution to another. In Table 4 results are presented for 100 runs and are given in terms of average error, standard error deviation and percentage of diverged runs for fixed c as well as for one and two parametric optimization. Maximal errors over 10% are excluded from the calculation of average error and standard deviation and are treated as diverged and listed separately as *diverged*. Parameter c for $\delta = 25\%$ is set to 15 as a balance between accuracy and reasonable stability (number of converged results). For $\delta = 35\%$ the shape parameter c is fixed to 10.

Table 4: Average overall maximal error, standard deviation and number of diverged simulations as a function of fixed c, one parametric (1P) and two parametric (2P) optimization.

			avg. error	std. dev.	% diverged
δ= 25%	grid 30×30	c=15	4.63	0.54	0
		1P	4.89	0.41	0
		2P	4.75	0.36	0
	grid 40×40	c=15	1.97	0.26	2
		1P	1.95	0.15	0
		2P	1.90	0.14	0
δ = 35%	grid 30×30	c=10	6.65	0.85	14
		1P	6.60	0.83	4
		2P	6.57	0.77	3
	grid 40×40	c=10	3.30	0.83	15
		1P	3.21	0.84	6
		2P	3.13	0.93	6

Table 4 clearly shows that stability (number of converged runs) of the optimized weight function is greatly improved with the optimization. Although the average errors are sometimes just a little bit bigger in case of optimized parameters than

for the fixed parameter, optimized shape parameters still give smaller standard deviation. While for $\delta = 25\%$ and grid density 30×30 the optimized and non-optimized solutions have 100% convergence of calculations, on the finer 40×40 grid the non-optimized solution diverged in 2% of cases. For $c = 20$ (which showed the best accuracy for the same conditions in Table 2) on the 30×30 grid, the simulation diverged in 14% of cases.

With strongly non-uniform grids ($\delta = 35\%$) the differences in the number of converged simulations is much larger for both grid densities. In the case of the 30×30 grid there are almost 4 times more converged results in the case of optimized shape parameters and in the case of the 40×40 grid more than 2 times.

The difference in the number of converged results between one parametric and two parametric optimization is negligible. However, the average accuracy and standard deviation is smaller in case of two parametric optimization. Optimization time, naturally, increases with the number of the optimization degree of freedom as shown in Table 5. The time for 100 calculation nodes and one parametric optimization is taken as reference time t_{ref} and all others are calculated as

$$\eta_t = \frac{t}{t_{ref}}, \tag{20}$$

Table 5: Relative optimization time for one parametric (1P) and two parametric optimization (2P).

Number of nodes	1P	2P
100	1.00	2.65
400	3.66	11.54
900	8.13	26.38
1600	15.48	49.33
2500	27.61	84.39
10000	173.62	404.47

5 Conclusions

In this work an idea about finding the optimal value of weight function shape parameters for non-uniform grids is presented. Numerical realization is made and tested on one meshless method (DAM) and one time dependent PDE. The idea developed in this work is original from two standpoints. First, the optimization procedure is set locally on each subdomain ${}_n\Omega$. Therefore each node is optimized separately by the same local reference quality function according to the specific node distribution in ${}_n\Omega$. Second, this idea is extended to two parametric optimization for radially nonsymmetric weight function. The most difficult problem is to find the appropriate reference quality and object function. The object function is in this work constructed intuitively and in great part on an experimental basis. The main conclusions, however, are very important. First, the reference quality function should be tested locally for each domain of influence or support, and second, the defined object function using the second derivatives is good indicator for testing. This statement may not be valid for some meshless numerical methods such as Indirect Radial Basis Function Network Method [Mai-Cao and Tran-Cong (2003), Mai-Cao and Tran-Cong (2005)] due to its unique property that second derivatives are better approximated as lower order derivatives. Optimization algorithm, as well as the reference quality function and the object function used in this work may not be the most optimal but they give promising results and the whole procedure is quite robust and easy to code.

In presented examples it was shown that the fixed values of shape parameters are far from being optimal but they rather range from larger values for uniform grids to lower values for increasingly non-uniform grids. In this work one and two parametric weight functions are used. The difference between one and two parametric weight functions in terms of accuracy of the calculations and stability of the simulation is small. Optimization time, however, increases considerably with the optimization degree level, but it can be reduced by tabulated object function values.

The same idea can be applied to any MLS based meshless method most probably with different reference quality and object functions.

References

Atluri, S. N. (2004): *The Meshless Methods (MLPG) for Domain & BIE Discretizations*. Tech Science Press.

Atluri, S. N.; Shen, S. (2002): *The Meshless Local Petrov-Galerkin (MLPG) Method*. Tech Science Press.

Belytscko, T.; Krongauz, Y.; Fleming, M.; Organ, D.; Liu, W. K. (1996): Smoothing and acceleration computations in the element free galerkin method. *Computational Methods in Applied Mathematics*, vol. 74, pp. 116–126.

Couturier, S.; Sadat, H. (1999): Melting driven by natural convection: A comparison exercise. *International Journal of Thermal Sciences*, vol. 38, pp. 5–26.

Fortran (2003): Compaq Visual Fortran 6.6, 2003.

Golberg, M. A.; Chen, C. S.; Karur, S. R. (1996): Improved multiquadric approximation for partial differential equations. *Engineering Analysis with Boundary Elements*, vol. 18, pp. 9–17.

Kansa, E. J. (1990): Multiquadrics - a scattered data approximation scheme with application to computational fluid dynamics-i. *Computers & Mathematics with Applications*, vol. 19, pp. 127–145.

Kansa, E. J.; Hon, Y. C. (2000): Circumventing the ill-conditioning problem with multiquadric radial basis functions: Applications to elliptic partial differential equations. *Computers & Mathematics with Applications*, vol. 39, pp. 123–137.

Lazzaro, D.; Montefusco, L. B. (2002): Radial basis functions for the multivariate interpolation of large scattered data sets. *Journal of Computational and Applied Mathematics*, vol. 140, pp. 521–636.

Liu, G. (2003): *Mesh Free Methods: Moving Beyond the Finite Element Method*. CRC Press, London, New York.

Mai-Cao, L.; Tran-Cong, T. (2003): *Solving of Time-dependent {PDE}s with Meshless IRBFN-based Method*.

Mai-Cao, L.; Tran-Cong, T. (2005): A meshless IRBFN-based method for transient problems. *CMES: Computer Modeling in Engineering & Science*, vol. 7, no. 2, pp. 149–171.

Nayroles, B.; Touzot, G.; Villon, P. (1991): L'approximation diffuse. *Mécanique des Milieux Continus*, vol. 2, pp. 293–296.

Nie, Y. F.; Atluri, S. N.; Zou, C. W. (2006): The optimal radius of the support of radial weights used in moving least squares approximation. *CMES: Computer Modeling in Engineering & Science*, vol. 12, no. 2, pp. 137–147.

Perko, J. (2005): *Modelling of Transport Phenomena by the Diffuse Approximate Method*. Doctoral dissertation, School of Applied Sciences, Nova Gorica Polytechnic, 2005.

Perko, J.; Šarler, B.; Rek, Z. (2000): Convergence study of dual reciprocity BEM for Navier-Stokes equations. *Zeitschrift fűr Angewandte Mathematik und Mechanik*, vol. 80, pp. 839–840.

Pollandt, R. (1997): Solving nonlinear differential equations of mechanics with the boundary element and radial basis functions. *International Journal for Numerical Methods in Engineering*, vol. 40, pp. 61–73.

Prax, C.; Salagnac, P.; Sadat, H. (1998): Diffuse approximation and control-volume-based finite-element methods: A comparative study. *Numerical Heat Transfer, Part B*, vol. 34, pp. 303–321.

Sadat, H.; Couturier, S. (2000): Performance and accuracy of a meshless method for laminar natural convection. *Numerical Heat Transfer, Part B*, vol. 37, pp. 455–467.

Shepard, D. (1968): A two dimensional interpolation function for irregularly spaced data. In *Proceedings of the 23rd National Conference*, pp. 517–523. ACM.

Sophy, T.; Sadat, H. (2002): A three dimensional formulation of a meshless method for solving fluid flow and heat transfer problems. *Numerical Heat Transfer, Part B: Fundamentals*, vol. 41, pp. 193–198.

Šarler, B.; Vertnik, R.; Perko, J. (2004): Solution of temperature field in DC cast aluminum alloy slab by the diffuse approximate method. In Tadeu, A.; Atluri, S. N.(Eds): *Advances in computational and experimental engineering and sciences: proceedings of the 2004 International conference on computational & experimental engineering & sciences*, pp. 1364–1371. Tech Science Press, Los Angels.

Šarler, B.; Vertnik, R.; Perko, J. (2005): Application of diffuse approximate method in convective-diffusive solidification problems. *CMC: Computers, Materials, & Continua*, vol. 2, pp. 77–83.