

# Novel multilevel techniques for convergence acceleration in the solution of systems of equations arising from RBF-FD meshless discretizations

Riccardo Zamolo<sup>a,\*</sup>, Enrico Nobile<sup>a</sup>, Božidar Šarler<sup>b,c</sup>

<sup>a</sup>*Department of Engineering and Architecture, University of Trieste, via Alfonso Valerio 10,  
34127 Trieste, Italy*

<sup>b</sup>*Faculty of Mechanical Engineering, University of Ljubljana, Aškerčeva cesta 6, 1000  
Ljubljana, Slovenia*

<sup>c</sup>*Institute of Metals and Technology, Lepi pot 11, 1000 Ljubljana, Slovenia*

---

## Abstract

The present paper develops two new techniques, namely additive correction multicloud (ACMC) and smoothed restriction multicloud (SRMC), for the efficient solution of systems of equations arising from Radial Basis Function-generated Finite Difference (RBF-FD) meshless discretizations of partial differential equations (PDEs). RBF-FD meshless methods employ arbitrary distributed nodes, without the need to generate a mesh, for the numerical solution of PDEs. The proposed techniques are specifically designed for the RBF-FD data structure and employ simple restriction and interpolation strategies in order to obtain a hierarchy of coarse-level node distributions and the corresponding correction equations.

Both techniques are kept as simple as possible in terms of code implementation, which is an important feature of meshless methods. The techniques are verified on 2D and 3D Poisson equations, defined on non-trivial domains, showing very high benefits in terms of both time consumption and work to convergence when comparing the present techniques to the most common solver approaches. These benefits make the RBF-FD approach competitive with standard grid-based approaches when the number of nodes is very high, allowing large size problems to be tackled by the RBF-FD method.

*Keywords:* Meshfree methods, Partial differential equations, Linear solvers, Poisson equation, Collocation

*2010 MSC:* 65N06, 65N50

---

\*Corresponding author

*Email addresses:* [riccardo.zamolo@phd.units.it](mailto:riccardo.zamolo@phd.units.it) (Riccardo Zamolo), [nobile@units.it](mailto:nobile@units.it) (Enrico Nobile), [bozidar.sarler@fs.uni-lj.si](mailto:bozidar.sarler@fs.uni-lj.si) (Božidar Šarler)

*Non-standard abbreviations:* MC = multicloud; ACMC = additive correction multicloud; SRMC = smoothed restriction multicloud

## 1. Introduction

Complex geometries characterize most problems of engineering relevance, whose numerical solution with classical methods requires domain meshing, i.e., a spatial discretization of the domain with proper elements [1]. The global effort  
5 needed to generate such geometrical mesh can often be much greater than the one required by the remaining parts of the simulation chain, in particular when high quality grids are required: this fact led to the birth and growth of the so-called meshless methods covering a wide range of problems [2].

This work is focused on a particular class of meshless methods which uses  
10 local Radial Basis Function (RBF) trial functions and local collocation technique for the numerical solution of PDEs [3]: the required space derivatives are approximated on a local set of nodes (local support) by deriving the RBF expansion. This approach is known as RBF-generated Finite Differences (RBF-FD) [4–9] since it resembles the standard Finite Differences on irregular node  
15 arrangements [10]. Such approach is also known as Local Radial Basis Function Collocation Method (LRBFCM) [11]. When such RBF-FD approach is employed to precisely discretize an elliptic steady-state problem, i.e., no time dependency, or a parabolic time dependent problem with at least one implicit term in time, a large and sparse system of algebraic equations has to be solved.

The iterative solution of systems of equations in mesh reduction methods  
20 has already been investigated [12]. In the linear case, a typical choice [6] for an iterative solver is BiCGSTAB [13] since the resulting matrix is nonsymmetric, with an incomplete LU factorization (ILU) as preconditioner [14]. This choice can be prohibitive in terms of both time and memory consumption when the  
25 size of the problem, i.e., the number of meshless nodes, becomes very large and therefore more efficient solvers have to be used. In order to achieve such desired convergence acceleration, the basic concepts of multigrid (MG) methods [15] are in this paper extended to the linear systems arising from RBF-FD discretization of two-dimensional (2D) and three-dimensional (3D) Poisson equations. We use  
30 the name multicloud (MC) for the developed techniques since the multilevel approach is coupled to the meshless discretization and data structure. The multicloud denomination is also used by Katz and Jameson [16] to define meshless coarse-level operators within a similar multilevel approach. Algebraic MG (AMG) [17, 18] can be employed with meshless discretizations since it operates  
35 at equation level without any additional geometrical information. However, it is useful to develop simple MC approaches which are specifically designed and tuned for the RBF-FD discretizations. This is of utmost importance for the use of the RBF-FD approach in large scale problems [19].

The working principle of the MG approach, formally defined in the paper  
40 of Brandt [20], is to optimally reduce the various frequency components of the error on a hierarchy of coarser grids by means of proper interpolation, restriction and smoothing operators. The key idea behind this working principle was somehow previously discovered by some authors that implicitly introduced a two-level correction which is the foundation of the recursive definition of the  
45 MG algorithm. Southwell [21] proposed block/group relaxations for mechanical

frameworks and stated that such techniques are “almost essential to practical success”. Stiefel [22] proposed an analogous block relaxation for a FD discretization of a Poisson problem and observed that the steepest descent for the quadratic function of residuals is equivalent to impose zero-mean residuals on the block, i.e., the summation of block equations; it is then suggested to use smaller blocks within the starting block, i.e., a MG approach. Fedorenko [23, 24] formalized the first correction scheme with a two-level FD discretization of a 2D Poisson problem and bilinear interpolation. De la Vallee Poussin and Timlake [25] obtained an additive block correction strategy from the case of a FD discretized heat conduction problem with internal “slots” with infinite thermal conductivity and reported convergence acceleration also for the case of constant thermal conductivity, while Settari and Aziz [26] formalized a general additive correction strategy showing its feasibility to practical problems.

Even after the formalization of the MG approach, many additive correction MG approaches have been proposed and employed to solve practical problems [27–30] due to ease of implementation, since the involved interpolation/restriction operators are given by piecewise constant functions. Another slightly more complex, but still simple choice, for interpolation/restriction operators is given by smoothing the piecewise constant operators using the problem equations themselves [31–34]. These two types of AMG are particularly attractive for meshless applications because of their straightforward implementation, independence upon geometric discretization and because they require only a single fine-grid discretization.

The application of MG principles to meshless methods has been previously investigated only in a limited number of works: Leem, Oliveira and Stewart [35] studied the application of AMG with smoothed transfer operators to a Poisson problem discretized with the Reproducing Kernel Particle Method (RKPM). Seibold [36] studied the same problem with AMG but using a Generalized Finite Difference Method (GFDM). More recently, Katz and Jameson [16] developed a multigrid technique with meshless transfer operators at coarse levels, and called it multicloud (MC).

The work presented in this paper represents the first attempt to extend the multigrid solution approach to RBF-FD meshless methods in the case of straight additive correction and smoothed transfer operators strategies. Such strategies are employed in the present paper to develop two simple MC techniques: additive correction multicloud (ACMC) and smoothed restriction multicloud (SRMC) which differ for the type of restriction strategy only. In ACMC the restriction operator is piecewise constant over the restriction support while in SRMC it is smoothed from constant by applying one Jacobi iteration using the problem equations themselves, resulting in an increased support size. The implementation of such procedures is quite simple and straightforward within the RBF-FD meshless data structure. Both techniques can be used as standalone solvers and as preconditioners for iterative solvers such as BiCGSTAB, allowing the convergence acceleration in the case of RBF-FD discretizations of 2D and 3D Poisson equations.

A particular attention is given to the study of the factors influencing the

convergence properties of the proposed MC approaches. Tests are carried out with different domains and different boundary conditions for 2D and 3D Poisson equations in order to assess the characteristics of the developed techniques from a practical point of view. Excellent results have been obtained as compared to a standard BiCGSTAB solver with ILU preconditioning (BiCGSTAB/ILU). High benefits in terms of savings in computing time (up to 20 times in 2D) and amount of work to convergence (up to 10 times in 2D) have been achieved when a large number of nodes is employed ( $\approx 10^6$ ). APMC shows slightly worse performance than SRMC but it is less sensitive to the choice of MC parameters and its implementation is easier and extremely straightforward, while the coupling with a classical iterative solver such as BiCGSTAB allows additional gains in performance and in the reduction of sensitivity to MC parameters in 2D cases.

Benefits in terms of savings in computing time have also been obtained in the 3D case (up to 2-2.7 times) for large size problems, i.e.,  $N > 10^6$  nodes, while the convergence work for MC strategies is shown to grow slower than the BiCGSTAB/ILU convergence work when the size of the problem  $N$  is increased, in the range for  $N$  employed in this work.

## 2. Meshless discretization

### 2.1. Problem definition

The model problem is a Poisson equation in the unknown field  $u$ , defined on a domain  $\Omega \subset \mathcal{R}^D$  in  $D$  dimensions and subjected to mixed boundary conditions (BCs), i.e., Dirichlet BCs on  $\Gamma_D$  and Neumann BCs on  $\Gamma_N$

$$\left\{ \begin{array}{ll} -\nabla^2 u = q & \text{in } \Omega \\ u = \bar{u} & \text{on } \Gamma_D \\ \frac{\partial u}{\partial \mathbf{n}} = \bar{f} & \text{on } \Gamma_N \end{array} \right. \quad (1)$$

where  $\Gamma_D \cup \Gamma_N = \partial\Omega$  is the domain boundary,  $\mathbf{n}$  is the exterior normal to the boundary,  $q$ ,  $\bar{u}$  and  $\bar{u}_n$  are known functions. A negative Laplacian is considered in order to deal with positive-definite matrices in the following. The numerical solution of Problem (1) is of primary importance in many physics/engineering problems such as fluid mechanics, solid mechanics and electromagnetics.

### 2.2. Node distributions

The isotropic node distributions required by the RBF-FD meshless discretizations have been obtained through the modified quadtree algorithm followed by repel refinement recently proposed by Zamolo and Nobile [37] for 2D cases, while 3D distributions are obtained through an analogous otree-based algorithm [38]. The algorithms are briefly summarized as follows.

Given a prescribed spacing function  $s(\mathbf{x})$ , which defines the local linear spacing between the nodes, an initial node distribution is obtained through the

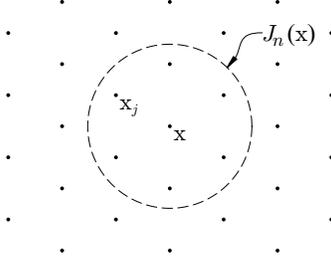


Figure 1: Example of a local support for the RBF expansion around  $\mathbf{x}$ . The nodes  $\mathbf{x}_j$  inside the circle are the 7 nearest nodes from  $\mathbf{x}$ .

quadtrees/octrees algorithms modified with a dithering correction in order to reduce the “nodal quantization error”, i.e., the difference between the integer  
 130 number of the nodes that can be contained in any subset  $\omega \subseteq \Omega$  and the prescribed (non-integer) number  $\int_{\omega} s(\mathbf{x})^{-D} d\omega$ . The resulting node distribution is then improved by the application of a few refinement iterations based on radial repel forces between the nodes.

### 2.3. RBF-FD discretization

135 The basis for RBF-FD method [11], used for the spatial discretization of problem expressed by Eq. (1), are shortly presented as follows.

#### 2.3.1. RBF interpolation

Given a distribution of  $N$  distinct nodes  $\mathbf{x}_i \in \Omega$ , the unknown field  $u$  around  $\mathbf{x} \in \Omega$  is approximated through the following local expansion  $v$

$$v(\mathbf{x}) = \sum_{j \in J_n(\mathbf{x})} a_j \varphi(\|\mathbf{x} - \mathbf{x}_j\|) + \mathbf{b}^T(\mathbf{x} - \bar{\mathbf{x}}) + c \quad (2)$$

140 where  $J_n(\mathbf{x})$  represents the set of indices  $j$  of the  $n$  nodes  $\mathbf{x}_j$  closest to  $\mathbf{x}$  (local support),  $\bar{\mathbf{x}}$  is the mean position of these local nodes  $\mathbf{x}_j$  and  $a_j$  are the RBF expansion coefficients.  $\mathbf{b}$  and  $c$  are the coefficients for the linear polynomial part and  $\mathbf{b}$  is a  $D$ -component column vector. A 2D example of local support in the  
 145 case of  $n = 7$  is depicted in Figure 1.  $v(\mathbf{x})$  is therefore a linear combination of  $n$  radial functions  $\varphi$  centered at  $n$  local nodes  $\mathbf{x}_j$  plus a linear polynomial in  $\mathbf{x}$ .

Hardy’s Multiquadric (MQ) [39–41] are chosen as RBFs [11, 42, 43]

$$\varphi(r) = \sqrt{1 + (\varepsilon r)^2} \quad (3)$$

where the shape factor  $\varepsilon$  is rescaled as  $\varepsilon = s_M \bar{\varepsilon} / d_n(\bar{\mathbf{x}})$ ;  $\bar{\varepsilon}$  is the rescaled shape

150 factor,  $s_M$  is the maximum prescribed spacing function on  $\Omega$  and  $d_n(\bar{\mathbf{x}})$  is the local support size

$$d_n(\bar{\mathbf{x}}) = \sqrt{\frac{1}{n} \sum_{j \in J_n(\bar{\mathbf{x}})} \|\mathbf{x}_j - \bar{\mathbf{x}}\|^2} \quad (4)$$

The coefficients  $a_j$ ,  $\mathbf{b}$  and  $c$  in Eq. (2) can be obtained by writing the same Eq. (2) for the  $m \leq n$  local nodes  $\mathbf{x}_i$  which do not lie on Neumann boundary  $\Gamma_N$

$$\sum_{j \in J_n(\mathbf{x})} a_j \varphi(\|\mathbf{x}_i - \mathbf{x}_j\|) + \mathbf{b}^T(\mathbf{x}_i - \bar{\mathbf{x}}) + c = v_i \quad (5)$$

155 where  $v_i$  is the unknown value for  $v$  in  $\mathbf{x}_i$ . A natural constraint is to require exactness of Eq. (2) for any linear function  $v$ : Eq. (5) with  $v_i = \mathbf{b}^T \mathbf{x}_i + c$  gives the unique obvious solution  $a_j = 0$  since MQ-RBF interpolation matrix is always invertible [44]. Therefore any homogeneous linear system in  $a_j$  with  
160 rank  $D + 1$  could be employed. It is then convenient to impose the following homogeneous conditions in order to get a symmetric interpolation matrix

$$\sum_{j \in J_n(\mathbf{x})} a_j (\mathbf{x}_j - \bar{\mathbf{x}}) = \mathbf{0}, \quad \sum_{j \in J_n(\mathbf{x})} a_j = 0 \quad (6)$$

where  $\mathbf{x}_j$  and  $\bar{\mathbf{x}}$  are  $D$ -component column vectors. If any of the  $\mathbf{x}_j$  local nodes lie on Neumann boundary  $\Gamma_N$ , for each of these  $n_N = n - m$  Neumann nodes  
165  $\hat{\mathbf{x}}_l \in \Gamma_N$  the corresponding boundary condition has to be satisfied

$$\frac{\partial v}{\partial \mathbf{n}} \Big|_{\mathbf{x}=\hat{\mathbf{x}}_l} = \sum_{j \in J_n(\mathbf{x})} a_j \Psi_j(\hat{\mathbf{x}}_l) + \mathbf{b}^T \mathbf{n}_l = \bar{f}_l \quad (7)$$

where  $\bar{f}_l$  is the value of Neumann BC term  $\bar{f}$  at the Neumann node  $\hat{\mathbf{x}}_l$ ,  $\mathbf{n}_l$  is the exterior normal to the boundary at the node  $\hat{\mathbf{x}}_l$  and  $\Psi_j(\mathbf{x}) = \frac{\partial}{\partial \mathbf{n}} [\varphi(\|\mathbf{x} - \mathbf{x}_j\|)]$ .

The local interpolation system obtained from Eqs. (5)-(7) is

$$\left[ \begin{array}{ccc|ccc} \varphi(\|\mathbf{x}_1 - \mathbf{x}_1\|) & \cdots & \varphi(\|\mathbf{x}_1 - \mathbf{x}_n\|) & \mathbf{x}_1^T - \bar{\mathbf{x}}^T & 1 & \\ \vdots & \ddots & \vdots & \vdots & \vdots & \\ \varphi(\|\mathbf{x}_m - \mathbf{x}_1\|) & \cdots & \varphi(\|\mathbf{x}_m - \mathbf{x}_n\|) & \mathbf{x}_m^T - \bar{\mathbf{x}}^T & 1 & \\ \hline \mathbf{x}_1 - \bar{\mathbf{x}} & \cdots & \mathbf{x}_n - \bar{\mathbf{x}} & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1 & \cdots & 1 & 0 & \cdots & 0 \\ \hline \Psi_1(\hat{\mathbf{x}}_1) & \cdots & \Psi_n(\hat{\mathbf{x}}_1) & \mathbf{n}_1^T & 0 & \\ \vdots & \ddots & \vdots & \vdots & \vdots & \\ \Psi_1(\hat{\mathbf{x}}_{n_N}) & \cdots & \Psi_n(\hat{\mathbf{x}}_{n_N}) & \mathbf{n}_{n_N}^T & 0 & \end{array} \right] \left\{ \begin{array}{c} a_1 \\ \vdots \\ a_n \\ b_1 \\ \vdots \\ b_D \\ c \end{array} \right\} = \left\{ \begin{array}{c} v_1 \\ \vdots \\ v_m \\ 0 \\ \vdots \\ 0 \\ \bar{f}_1 \\ \vdots \\ \bar{f}_{n_N} \end{array} \right\} \quad (8)$$

170 By collecting  $n$  local coefficients  $a_j$ ,  $m$  unknown values  $v_i$  and  $n_N$  Neumann boundary contributions  $\bar{f}_i$  in local column vectors  $\mathbf{a}$ ,  $\mathbf{v}$  and  $\bar{\mathbf{f}}$ , respectively, the local interpolation system (8) becomes

$$\mathbf{G}_{\mathbf{x}} \begin{Bmatrix} \mathbf{a} \\ \mathbf{b} \\ c \end{Bmatrix}_{\mathbf{x}} = \begin{Bmatrix} \mathbf{v} \\ \mathbf{0} \\ \bar{\mathbf{f}} \end{Bmatrix}_{\mathbf{x}} \quad (9)$$

where the zero vector is a  $D$ -component column vector (6).  $\mathbf{G}_{\mathbf{x}}$  is a square  $(n + D + 1) \times (n + D + 1)$  matrix whose first  $(m + D + 1)$  rows represent the contribution of MQ-RBF interpolation with an appended linear polynomial as expressed by Eq. (2), while the remaining  $(n - m)$  rows represent the contribution of Neumann BCs if any local node lies on the Neumann boundary. If  $m = n$ , i.e., no Neumann local nodes, an  $LDL^T$  factorization is performed on  $\mathbf{G}_{\mathbf{x}}$  since it is symmetric but not positive definite, otherwise a Schur complement [45] is first performed for the Neumann contributions, followed again by an  $LDL^T$  factorization on the remaining symmetric part of  $\mathbf{G}_{\mathbf{x}}$ .

### 2.3.2. Collocation

Poisson Eq. (1) with the local RBF expansion of Eq. (2) becomes

$$-\nabla^2 v(\mathbf{x}) = - \sum_{j \in J_n(\mathbf{x})} a_j \nabla^2 \varphi(\|\mathbf{x} - \mathbf{x}_j\|) = q(\mathbf{x}) \quad (10)$$

185 The compact form of the previous equation for a generic node  $\mathbf{x}_k$  is

$$-\mathbf{L}^T(\mathbf{x}_k) \mathbf{a} = q(\mathbf{x}_k) \quad (11)$$

where  $\mathbf{L}(\mathbf{x}_k)$  is the column vector of the Laplacian of RBF  $\varphi$ , evaluated in  $\mathbf{x}_k$  for each of the  $n$  neighbors  $\mathbf{x}_j$ . Solving Eq. (9) for  $\mathbf{a}$  gives

$$-\mathbf{L}^T(\mathbf{x}_k) [\mathbf{G}_{\mathbf{x}_k}^{-1}]_{\mathbf{a}} \begin{Bmatrix} \mathbf{v} \\ \mathbf{0} \\ \bar{\mathbf{f}} \end{Bmatrix}_{\mathbf{x}_k} = q(\mathbf{x}_k) \quad (12)$$

190 where  $[\mathbf{G}_{\mathbf{x}_k}^{-1}]_{\mathbf{a}}$  is the top  $n \times (n + D + 1)$  submatrix of  $\mathbf{G}_{\mathbf{x}_k}^{-1}$ . The stencil for node  $\mathbf{x}_k$  is therefore defined by the first  $m$  components of row vector  $-\mathbf{L}^T(\mathbf{x}_k) [\mathbf{G}_{\mathbf{x}_k}^{-1}]_{\mathbf{a}}$  (stencil coefficients) and by the index set  $J_n(\mathbf{x}_k)$  of the corresponding nodes, excluding possible Neumann nodes. The first component of stencil coefficients refers to the ‘‘central’’ node  $\mathbf{x}_k$ .

195 By writing Eq. (12) for each of the  $M$  nodes  $\mathbf{x}_k$  which do not lie on the boundary  $\Gamma$  (internal nodes) gives the following linear system

$$\mathbf{A}\mathbf{v} + \bar{\mathbf{A}}\bar{\mathbf{f}} = \mathbf{q} \quad (13)$$

where  $\mathbf{v}$  is the column vector of all internal and Dirichlet nodal values for  $v$ ,

200  $\bar{\mathbf{f}}$  is the column vector carrying all the local Neumann contributions  $\bar{\mathbf{f}}$  (if any),  $\mathbf{A}$  and  $\bar{\mathbf{A}}$  are the corresponding coefficient matrices and  $\mathbf{q}$  is the column vector of all the right hand side nodal contributions  $q(\mathbf{x}_k)$ . By rearranging Eq. (13), using a partition of *Internal* and *Dirichlet* nodes for  $\mathbf{v}$ , gives

$$\mathbf{A}_I \mathbf{v}_I = \mathbf{q} - \bar{\mathbf{A}} \bar{\mathbf{f}} - \mathbf{A}_D \mathbf{v}_D \quad (14)$$

205 Eq. (14) represents the final linear system that has to be solved for  $\mathbf{v}_I$ . The coefficient matrix  $\mathbf{A}_I$ , whose size is  $M \times M$ , is sparse and unsymmetric. The sparsity is due to the local nature of the RBF trial function in Eq. (2), while the asymmetry is due to the fact that the local RBF formulation does not ensure the same contribution for any couple of the neighboring nodes.

### 210 3. Domains, node distributions, BCs and analytical solutions

Two 2D and one 3D complex shaped domains are employed in this work in order to demonstrate the application of the proposed techniques to practical problems which are characterized by complex geometries and non-trivial node arrangements.

#### 215 3.1. Case 1

*Geometry.* The geometry of the 2D domain  $\Omega$  is a circle with  $N_g = 8$  periodical angular grooves, as depicted in Figure 2a together with the coordinate system. The ratio  $R_i/R_e$  between the internal and the external radii of the grooves has been chosen to be 0.8.

220 *Node distributions.* The following spacing function  $s(\mathbf{x})$  is employed for the generation of node distributions over this domain:

$$\frac{s_M}{s(\mathbf{x})} = 1 + \bar{r}^2 \frac{2 - \cos(2N_g \vartheta)}{3} \left[ 1 + k_E \exp(-k_S(\bar{r} - \bar{r}_S)^2) \right]^2 \quad (15)$$

where  $s_M$  is the maximum spacing function over the domain,  $\bar{r} = r/R_i$  is the dimensionless radius,  $k_E = 0.75$ ,  $k_S = 50$  and  $\bar{r}_S = 1 - (1 + k_E)/(2k_E k_S)$ ; 225 the ratio between the maximum and the minimum nodal spacing is therefore  $s_M/s_m \approx 4$ . An example of node distribution with  $N = 10^4$  nodes obtained with 50 refinement iterations [37] is displayed in Figure 2c. A snapshot of the particular part  $P$  of the node arrangement in the neighborhood of a corner is displayed in Figure 2d.

230 *Boundary conditions.* Dirichlet BCs on the whole boundary  $\Gamma$  are considered.

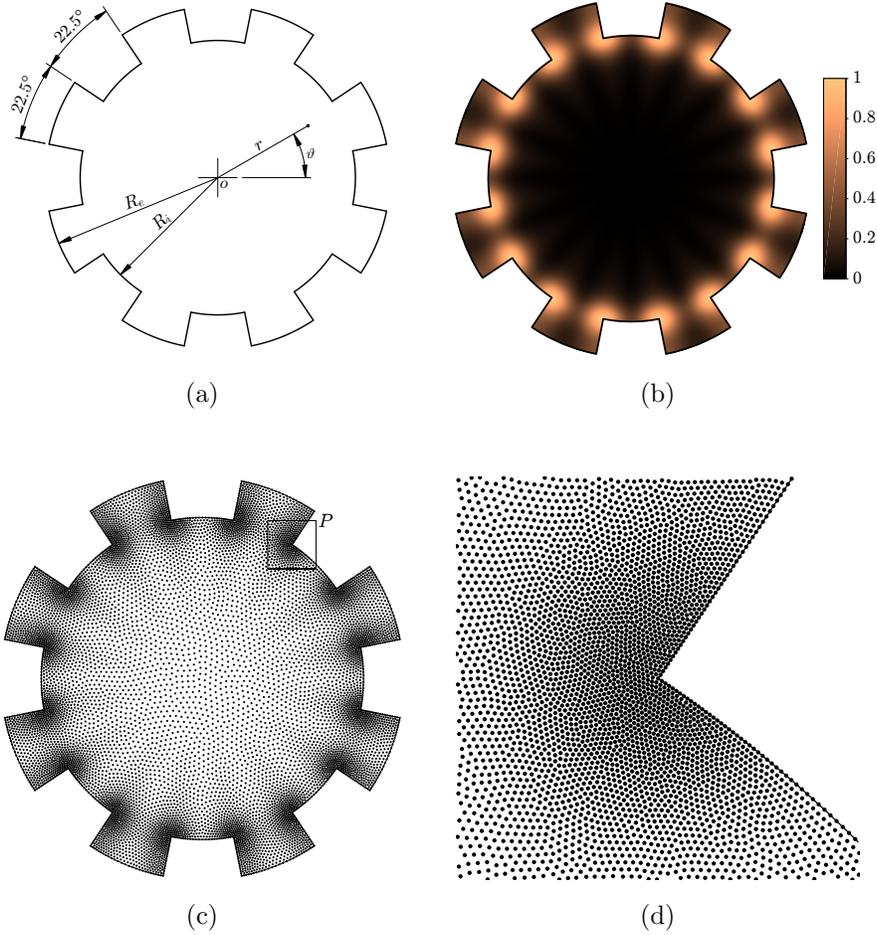


Figure 2: Case 1: geometry (a), analytical solution (b), example of node distribution with  $N = 10^4$  nodes (c) and a snapshot of the particular part  $P$  of the domain (d).

*Analytical solution.* The following analytical solution is used for BCs and for the definition of the error norms:

$$u(\mathbf{x}) = \bar{r}^3 \left[ 2 - \cos(2N_g \vartheta) \right] \left[ 1 + k_E \exp(-k_S(\bar{r} - \bar{r}_S)^2) \right]^3 \quad (16)$$

A graphical representation of this analytical solution is reported in Figure 2b, after scaling it to the interval  $[0, 1]$ , i.e.,  $u_G(\mathbf{x}) = u(\mathbf{x})/u_M$ , where  $u_G$  is the scaled analytical solution and  $u_M$  is the maximum value for  $u$  on  $\Omega$ .

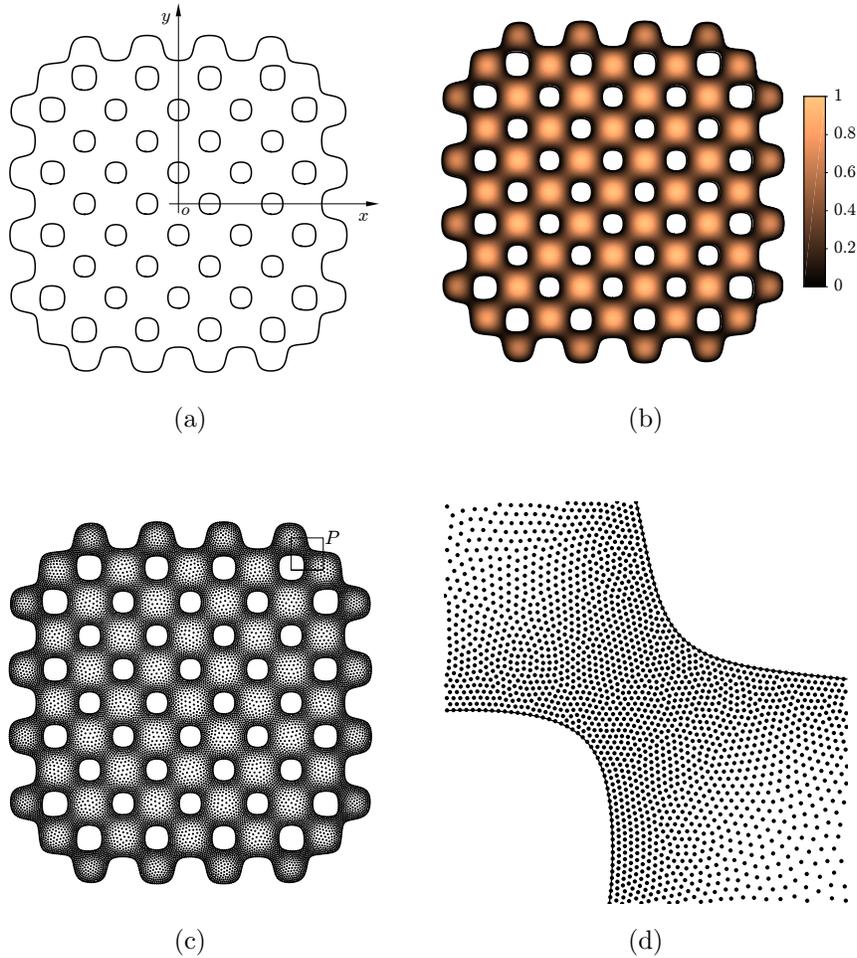


Figure 3: Case 2: geometry (a), analytical solution (b), example of node distribution with  $N = 15000$  nodes (c) and a snapshot of the particular part  $P$  of the domain (d).

### 3.2. Case 2

*Geometry.* The 2D domain  $\Omega$  is implicitly defined as the set  $\{\mathbf{x} \in \mathcal{R}^2 : G(\mathbf{x}) \geq 0\}$  where  $G$  is defined as

$$G(\mathbf{x}) = \frac{2}{3} \left[ \cos(6\pi x) \cos(6\pi y) - 2(x^4 + y^4)^2 + \frac{1}{2} \right] \quad (17)$$

240

The geometry of such complex shaped domain with multiple holes is depicted in Figure 3a together with the coordinate system. The boundary  $\Gamma$  is simply given by the level set  $G(\mathbf{x}) = 0$ .

245 *Node distributions.* The spacing function  $s(\mathbf{x})$  employed for the generation of node distributions over this domain is

$$s(\mathbf{x}) = s_m \left[ 1 + k_G G^2(\mathbf{x}) \right] \quad (18)$$

where  $s_m$  is the minimum spacing function over the domain and  $k_G = 3$ . The ratio between the maximum and the minimum nodal spacing is therefore  $s_M/s_m = 1 + k_G = 4$ . An example of node distribution with  $N = 15000$  nodes 250 obtained with 50 refinement iterations [37] is displayed in Figure 3c. A snapshot of the particular part  $P$  of the domain is displayed in Figure 3d, showing the node arrangement in the neighborhood of a double boundary.

*Boundary conditions.* Two types of BCs have been considered for this domain:

- Case 2Dir: Dirichlet BCs on the whole boundary;
- 255 • Case 2Mix: Mixed BCs: Dirichlet BCs on the “external” boundary, Neumann BCs on the boundary of the 40 “internal” holes.

*Analytical solution.* The analytical solution that has been used for BCs and error norms is simply defined as  $u(\mathbf{x}) = G(\mathbf{x})$ . A graphical representation of this analytical solution is shown in Figure 3b, after scaling it to the interval 260  $[0, 1]$ , i.e.,  $u_G(\mathbf{x}) = u(\mathbf{x})/u_M$ , where  $u_G$  is the scaled analytical solution and  $u_M$  is the maximum value for  $u$  on  $\Omega$ .

### 3.3. Case 3

*Geometry.* The 3D domain  $\Omega$  is obtained by extruding the 2D geometry of Case 2, Figure 3, along the  $z$  direction from  $z = 0$  to  $z = H_z = 1$ . The geometry of 265 such complex-shaped domain is depicted in Figure 4a.

*Node distributions.* The spacing function  $s(\mathbf{x})$  employed for the generation of node distributions over this 3D domain is the same defined by Eq. (18), for which there is no dependence upon the  $z$  coordinate. An example of node distribution with  $N = 4 \cdot 10^5$  nodes obtained with 50 refinement iterations [38] 270 is displayed in Figure 4b and 4c.

*Boundary conditions.* Dirichlet BCs on the whole boundary are considered. Since the geometry is obtained by extruding the 2D profile of Case 2, which is characterized by an external curve and 40 internal holes, the boundary  $\Gamma$  is then composed by two planar surfaces at  $z = 0$  and  $z = H_z$ , an external curved 275 surface and 40 cylindrical holes.

*Analytical solution.* The analytical solution used for BCs is the same as employed for Case 2:  $u(\mathbf{x}) = G(\mathbf{x})$ , where  $G$  is defined by Eq. (17), i.e., there is no dependence upon the  $z$  coordinate.

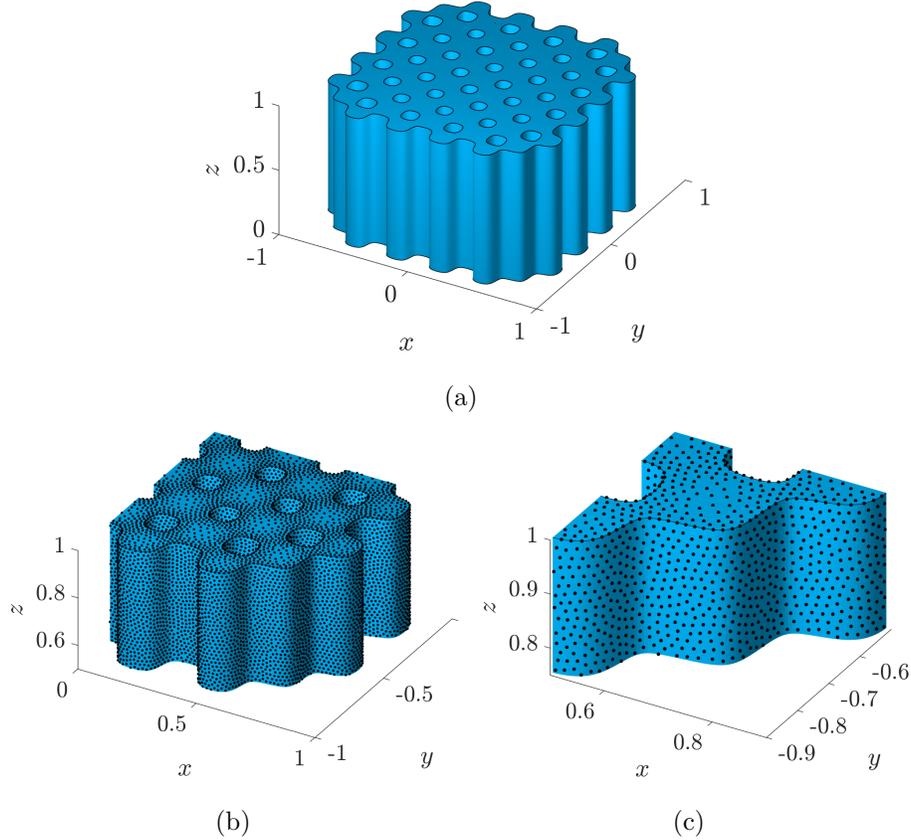


Figure 4: Case 3: geometry (a), enlarged views of the node distribution over a portion of the domain using  $N = 4 \cdot 10^5$  nodes (b,c).

#### 4. Multicloud techniques

280 In this section the basic concepts of MG methods are introduced and extended to the RBF-FD meshless discretization defined in Section 2. Analyses of MG methods are elaborated in [15, 17, 20].

##### 4.1. Multigrid

###### 4.1.1. Mathematical formulation

285 Let us consider the following sparse linear system arising from local discretization of the Poisson Eq. (1) (which is the case of the linear system (14))

$$\mathbf{M}\mathbf{u} = \mathbf{b} \quad (19)$$

290 The solution of system (19) using classical iterative methods, e.g., Jacobi, Gauss-Seidel, successive over-relaxation (SOR), leads to an efficient reduction of the high-frequency error components in space, while the low-frequency error

components are slowly reduced [24]. This property can be seen by considering the continuous-space approximation of the Jacobi iteration defined by  $u_{k+1} = u_k + \alpha(q + \nabla^2 u_k)$ , where  $\alpha = \mathcal{O}(h^2) > 0$  and  $h$  is the local reference size of the discretization.

295 Then, by considering a Fourier component of the error  $e_k = u - u_k = E_k e^{i\boldsymbol{\omega} \cdot \mathbf{x}}$  with  $\boldsymbol{\omega}$  denoting the vector of spatial frequencies, the convergence factor  $\mu(\boldsymbol{\omega})$  is

$$\mu(\boldsymbol{\omega}) = \left| \frac{E_{k+1}}{E_k} \right| = 1 - \mathcal{O}(\|\boldsymbol{\omega}\|_2^2 h^2) \quad (20)$$

300 where  $\mathcal{O}(1) = \omega_m \leq \|\boldsymbol{\omega}\|_2 \leq \omega_M = \mathcal{O}(h^{-1})$ . Therefore, rapidly fluctuating errors, i.e.,  $\|\boldsymbol{\omega}\|_2 \approx \omega_M$ , have favorable convergence factors  $\mu \ll 1$ , while low-frequency components, i.e.,  $\|\boldsymbol{\omega}\|_2 \approx \omega_m$ , are very slowly decaying because  $\mu \approx 1$ . Jacobi iteration acts as an error smoother and thus cannot be used as a practical single-grid solver for discretizations with a large number of unknowns.

305 Nevertheless, this smoothing property can be somehow exploited to improve this iterative method: the application of  $p$  smoothing iterations leaves only a smooth error  $\mathbf{e}_p = \mathbf{u} - \mathbf{u}_p$ . Then, the linear system (19) in terms of error becomes

$$\mathbf{M}\mathbf{e}_p = \mathbf{b} - \mathbf{M}\mathbf{u}_p =: \mathbf{r}_p \quad (21)$$

310 which states that the error  $\mathbf{e}_p$  can alternatively be computed from the residual  $\mathbf{r}_p$ . Since  $\mathbf{e}_p$  is smooth, it can therefore be well approximated using a smaller number of meaningful unknowns  $\tilde{\mathbf{e}}_p$  through a suitable interpolation operator  $\mathbf{I}_H^h$

$$\mathbf{e}_p = \mathbf{I}_H^h \tilde{\mathbf{e}}_p \quad (22)$$

315 The application of a restriction operator  $\mathbf{I}_h^H$  to Eq. (21) with the smooth approximation (22) gives the coarse grid correction equation

$$\underbrace{(\mathbf{I}_h^H \mathbf{M} \mathbf{I}_H^h)}_{\tilde{\mathbf{M}}} \tilde{\mathbf{e}}_p = \underbrace{\mathbf{I}_h^H \mathbf{r}_p}_{\tilde{\mathbf{r}}_p} \quad (23)$$

where  $\tilde{\mathbf{M}}$  and  $\tilde{\mathbf{r}}_p$  are the coarse correction matrix and the coarse residual, respectively.

Eq. (23), solved for  $\tilde{\mathbf{e}}_p$ , allows the final coarse grid correction

$$\mathbf{u}'_p = \mathbf{u}_p + \mathbf{I}_H^h \tilde{\mathbf{e}}_p \quad (24)$$

320 This correction is then followed by further  $p'$  smoothing iterations needed to smooth the high-frequency error components due to interpolation in the final

correction step. If linear system (23) is still too large to be solved directly or by simple iterative methods, this two-level approach can then be recursively  
 325 applied to the solution of the coarse grid correction system (23) itself, giving rise to a hierarchy of coefficient matrices  $\mathbf{M}_l$  where  $l$  is the MG level. This iterative approach yields the classical V-cycle MG algorithm [20].

#### 4.1.2. Transfer operators

The performances of MG methods depends upon the choice for transfer  
 330 operators between MG levels, i.e., interpolation  $\mathbf{I}_H^h$  and restriction  $\mathbf{I}_h^H$ , and their interaction with the smoothing operator. A recommended choice is

$$\mathbf{I}_h^H = (\mathbf{I}_H^h)^T \quad (25)$$

which states that the restriction is obtained from interpolation in order to fulfill  
 335 variational principles [17]. This condition is not always respected in the present work in favor of simpler numerical implementation.

For second order PDEs, such as the Poisson Eq. (1), the following condition has to be satisfied [20] in order to get the real MG convergence, i.e., convergence rate independent from the size of the problem

$$r_R + r_I > 2 \quad (26)$$

340 where  $r_R$  and  $r_I$  are the order of restriction and interpolation operators, respectively. For this reason piecewise linear interpolation ( $r_I = 2$ ), with its transpose piecewise linear restriction ( $r_R = 2$ ), is very common in geometric MG [15].

We briefly present two AMG approaches which are particularly attractive for meshless applications because of their simple implementation, independence  
 345 upon geometric discretization, low memory requirements and their need of the fine grid discretization only, i.e., no coarse-level discretizations are needed. Therefore boundary conditions are needed only at the finest level and do not need any additional treatment at the coarse levels.

*Aggregation-type MG.* The simplest AMG approach, also known as aggregation-type MG or additive correction MG, is given by piecewise constant transfer  
 350 operators and it can be implemented very easily. Nonetheless, a straight additive correction implementation does not produce a true MG convergence for diffusion problems because of an incomplete reduction of the smooth error components [17, 27]. This lack of MG convergence is also confirmed by Eq. (26) which is  
 355 not satisfied since  $r_R = r_I = 1$  for piecewise constant transfer operators.

A partial solution to this problem is overcorrection [27], which amplifies the coarse grid correction (24) with an overcorrection factor  $t > 1$  as follows

$$\mathbf{u}'_p = \mathbf{u}_p + t\mathbf{I}_H^h \tilde{\mathbf{e}}_p \quad (27)$$

where  $\mathbf{I}_H^h$  is piecewise constant. For a Poisson equation the optimal value of

360  $t$  depends upon the size of the aggregates and therefore it depends upon the specific geometric discretization.

*MG with smoothed transfer operators.* It is possible to overcome the problem of the incomplete reduction of the smooth error components encountered with the aggregation-type MG by smoothing transfer operators [31–34]. An under-relaxed homogeneous Jacobi iteration, i.e., with a relaxation smoothing factor [24]  $\omega_S < 1$ , is applied to the piecewise constant interpolation using the coefficient matrix  $\mathbf{M}_l$  at each MG level  $l$  [18], while the restriction operator is given by Eq. (25).

## 4.2. Multicloud

### 370 4.2.1. $C/F$ -splitting

A coarsening strategy is required to define a set of coarse-level variables required by interpolation, Eq. (22), resulting in a *Coarse/Fine*-splitting ( $C/F$ -splitting). In the RBF-FD meshless context, variables and equations are associated to nodes, therefore a  $C/F$ -splitting is given by a set of coarse nodes 375 obtained from a set of fine nodes by using a nodal coarsening strategy. We assume that the coarse-level node set is a subset of the fine-level node set.

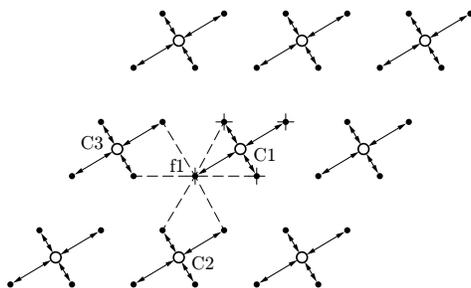
In this work we employ the nodal coarsening strategy proposed by Katz and Jameson [16] which is similar to the one proposed by Chan et al [46]: starting from a fine-level set where each node has a null flag  $F = 0$ , i.e. it is a free 380 node, each free node is visited sequentially, its flag is set to  $F = 2$  and its  $n_C$  nearest neighboring nodes with  $F = 0$  are set to have a flag  $F = 1$  ( $n_C = 6$  in 2D,  $n_C = 13$  in 3D). Then the coarse-level set is given by the nodes with  $F = 2$ . This choice leads to coarse sets whose number of nodes is approximately  $2^D$  times less than the number of fine-level nodes. Such isotropic coarsening strategy is 385 compatible with the isotropic node generation algorithm [38] employed in this work. The total number of nodes at MC level  $l \geq 0$  is  $M_l \approx M/2^{lD}$  and  $M_0 = M$  is the total number of internal nodes at the finest level  $l = 0$ .

### 4.2.2. Additive correction multicloud technique (ACMC)

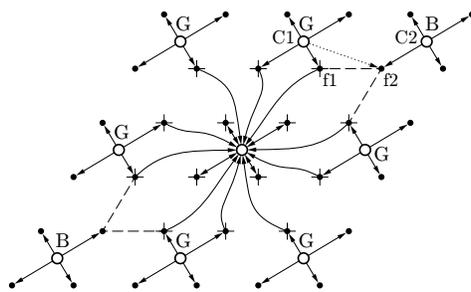
The first MC technique proposed in this work is additive correction multicloud (ACMC), which is based on aggregation-type MG with overcorrection, 390 described in Subsection 4.1.2, where both transfer operators are piecewise constant over their support. Given a  $C/F$ -splitting, the interpolation support is defined by choosing which coarse-level variables/nodes are employed to interpolate a fine-level variable, Eq. (22). Similarly, the restriction support is defined 395 by choosing which fine-level equations are employed to obtain a coarse-level correction equation by restriction, Eq. (23).

One nearest neighboring coarse-level node is used as the support for the constant interpolation. Therefore the interpolation operator  $\mathbf{I}_H^h$  is a  $M_l \times M_{l+1}$  matrix whose entries  $a_{ij}$  are non-zero only if coarse-level node  $j$  (at level  $l+1$ ) 400 is the nearest coarse-level node to the fine-level node  $i$  (at level  $l$ ), and the corresponding entry is  $a_{ij} = 1$ . The restriction is then given by Eq. (25), i.e., the constant restriction support is given by the fine-level nodes employing the

- fine-level nodes
  - coarse-level nodes
  - ◆ fine-level support nodes for restriction to central coarse-level node
- interpolation/restriction
  - equation links
  - > modified interpolation support



(a)



(b)

Figure 5: 2D interpolation and restriction support nodes for ACMC (a) and SRMC (b).

same coarse-level node as interpolation support. This strategy is depicted in Figure 5a for a 2D case where the arrows indicate interpolation (coarse to fine nodes) or restriction (fine to coarse nodes) with uniform and isotropic coarse/fine node distributions. For the sake of graphic clarity, the coarse-level nodes have been moved in order not to coincide with the fine/coarse-level nodes. The coincidence of some fine/coarse-level nodes is actually assumed in the nodal coarsening strategy described in Subsection 4.2.1 since the coarse-level node set is a subset of the fine-level node set.

By considering a stencil with  $n = 7$  neighboring nodes in 2D and  $n = 14$  neighboring nodes in 3D for the RBF-FD approach (these particular choices are explained in Subsection 4.3.1), the corresponding MC strategy maintains a constant stencil size, on average, at each level. In the 2D case, this property

415 can be deduced from Figure 5a by considering the central coarse-level node C1  
 and its related equation: its stencil is given by the coarse-level nodes employed  
 to interpolate all the fine-level nodes (marked with symbol  $\blacklozenge$  in Figure 5a)  
 which are involved in restriction to C1. For example, the equation for fine-level  
 node f1, which is one of the nodes used for the restriction to C1, depends upon  
 420 its 6 fine-level neighbors whose interpolation support employs only coarse-level  
 nodes C1, C2 and C3. Finally, it is possible to observe that the coarse correction  
 equation for C1 depends upon 6 coarse-level neighbors, and therefore the stencil  
 size  $n = 7$  is maintained at all coarse levels on the average.

#### 4.2.3. Smoothed restriction multcloud technique (SRMC)

425 The second MC technique proposed in this work is smoothed restriction  
 multcloud (SRMC), which is obtained from APMC by smoothing the piecewise  
 constant restriction operator as described in Subsection 4.1.2. More precisely,  
 the rows  $\mathbf{r}_i$  of the SRMC restriction matrix  $\mathbf{I}_h^H$  are obtained by applying one  
 under-relaxed homogeneous Jacobi iteration to the rows  $\bar{\mathbf{r}}_i$  of the APMC res-  
 430 triction matrix defined in Subsection 4.2.2, using the coefficient matrix  $\mathbf{M}_l$ .  
 Let us consider the homogeneous linear system

$$\mathbf{M}_l \mathbf{z} = \mathbf{0} \tag{28}$$

to be solved for  $\mathbf{z}$  by Jacobi iteration where the initial solution is set to be  $\mathbf{z}^{(0)} =$   
 $\bar{\mathbf{r}}_i^T$ , i.e., the transpose of the  $i$ -th row of the piecewise constant APMC restriction  
 435 matrix. Then the corresponding  $i$ -th row of the SRMC restriction matrix is  
 obtained after the first under-relaxed Jacobi iteration as  $\mathbf{r}_i^T = \mathbf{z}^{(1)}$ . The effect  
 of the Jacobi iteration is an increased radius for the restriction support.

The interpolation is still maintained piecewise constant as in APMC using  
 1 nearest neighboring coarse-level node for support, therefore Eq. (25) does not  
 440 hold anymore.

This strategy is depicted in Figure 5b in the 2D case where the solid line  
 arrows indicate interpolation (coarse to fine nodes) or restriction (fine to coarse  
 nodes) with uniform and isotropic coarse/fine node distributions. Again, for the  
 sake of graphic clarity, the coarse-level nodes have been moved in order not to  
 445 have coincident fine/coarse-level nodes.

By considering the central coarse-level node of Figure 5b and an RBF-FD  
 stencil with  $n = 7$  neighboring nodes, we observe that the radius of the res-  
 triction support is increased from 4 fine-level nodes of APMC approach to 14  
 fine-level nodes (marked with symbol  $\blacklozenge$  in Figure 5b) because of the smoothing  
 450 operation on the piecewise constant operator. This “extended” restriction sup-  
 port is given by the fine-level nodes whose stencils include any of the “original”  
 4 fine-level nodes defining the piecewise constant restriction support.

We also observe that this MC strategy does not maintain a constant stencil  
 size, on average, across the levels. This property can be deduced in the 2D case  
 455 from Figure 5b by considering again the central coarse-level node and its related  
 equation: its stencil is given by the coarse-level nodes employed to interpolate

all the fine-level nodes involved in the restriction to the central coarse-level node. It is then possible to observe that the coarse correction equation for the central coarse-level node depends upon 9 coarse-level neighbors, and therefore  
460 the stencil size  $n = 7$  is not maintained across levels on average, but will grow at coarse levels. To avoid a growing stencil size that would imply larger memory and computational requirements, we propose a modified restriction strategy which ensures a constant stencil size. In the 2D case, consider Figure 5b, where a coarse-level node is marked G (“good”) if it is involved in the interpolation of  
465 any of the 14 fine-level nodes defining the central coarse-level node restriction support, otherwise it is marked B (“bad”). Then, the restriction contribution of B coarse-level nodes is replaced by G coarse-level nodes only. For example, when the fine-level equation for node f1 is considered by restriction, the contribution of interpolation of the fine-level node f2, which is included in f1 stencil, is now  
470 given by the coarse-level node C1 and not by C2 anymore, because C1 is the G coarse-level node employed for the interpolation of f1. The same approach can be employed in 3D cases as well. Numerical tests confirmed that this strategy ensures a constant stencil size at all coarse levels, on average.

### 4.3. Multicloud convergence

475 As outlined before, the RBF-FD discretization employed in this work leads to an unsymmetric coefficient matrix, Eq. (14). Therefore, the classic positive definiteness requirement for symmetric matrices cannot be used to guarantee the convergence of the proposed MC techniques. An analogous requirement for the convergence of MG methods in the unsymmetric case is that the matrix be  
480 an  $M$ -matrix [47]. The sufficient conditions leading to a  $M$ -matrix structure in the context of the present meshless approach are presented in Appendix A.

#### 4.3.1. Positive stencils

In Appendix A the RBF-FD coefficient matrix of Eq. (14) is proved to be an  $M$ -matrix under the hypothesis of positive stencils for every internal node.  
485 The validity of such hypothesis depends upon the number of support nodes  $n$  for the MQ-RBF interpolation of Eq. (2), their geometrical arrangement and upon the MQ shape factor  $\varepsilon$  of Eq. (3). In what follows is the investigation of these factors on stencil positivity.

First of all, we investigate the effect of the number of support nodes  $n$  and  
490 MQ shape factor  $\varepsilon$  on stencil positivity for a 2D local node arrangement given by a central node surrounded by  $n - 2$  nodes which are distributed over a circle with angular intervals  $\Delta\alpha = 2\pi/(n - 1)$ , while the position  $\mathbf{y}$  of the remaining node is free. The positions of such a free node allowing a positive stencil are depicted as hatched areas in the diagrams of Figure 6 for  $n = 7, 8, 9$  and  $\varepsilon = 0.1, 1, 10$ ,  
495 while the areas delimited by red-dashed curves represent “safe” areas where the stencil coefficients  $g_1(\mathbf{y}), \dots, g_n(\mathbf{y})$  satisfy the condition  $G(\mathbf{y}) > G_M/2$ , where  $G(\mathbf{y}) = \min\{g_1(\mathbf{y}), -g_2(\mathbf{y}), \dots, -g_n(\mathbf{y})\}$  and  $G_M$  is the maximum value for  $G$  over the whole plane. The larger these areas, the more likely it is to get a positive stencil from general node arrangements with  $n$  support nodes. To the

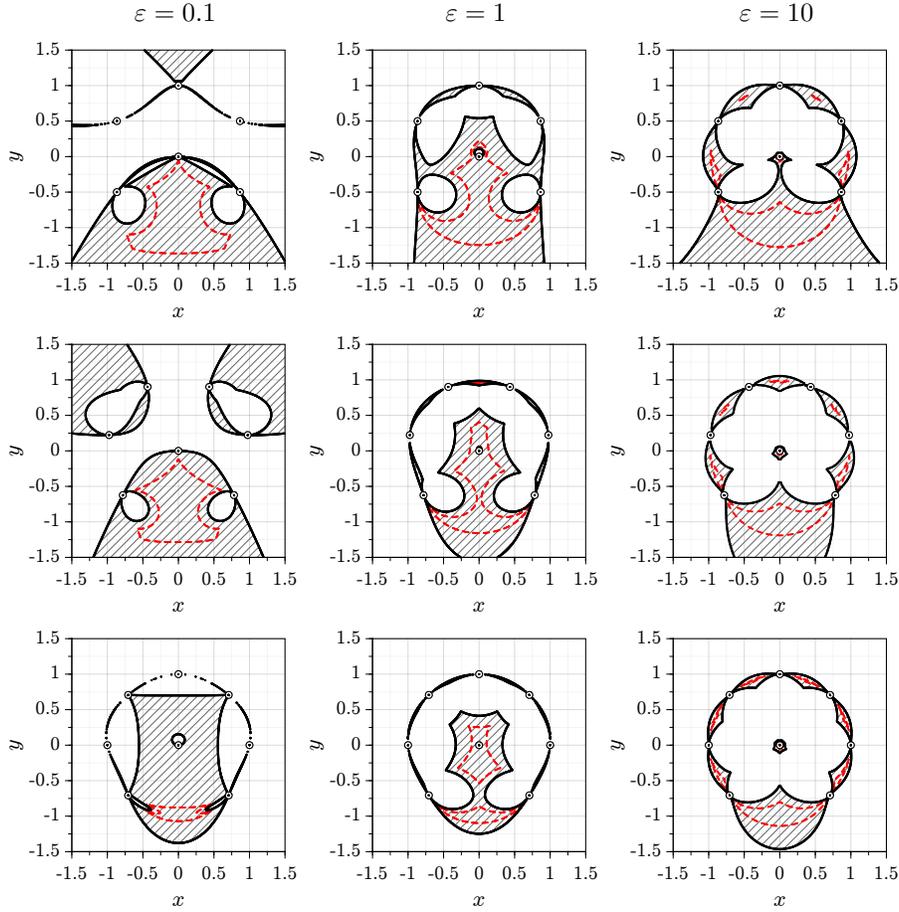


Figure 6: 2D stencil positivity plots for  $n = 7$  (top row),  $n = 8$  (middle row) and  $n = 9$  (bottom row): grey hatched areas represent positive stencil areas, while red dashed curves delimit “safe” areas for stencil positivity (see Subsection 4.3.1).

500 best of author’s knowledge, this is the first time that such analysis is performed in the context of multiquadric RBF-FD.

From Figure 6 it is possible to deduce that increasing  $n$  from 7 to 9 nodes will decrease the positive stencil area, also due to obvious geometric reasons, while the effect of shape factor  $\varepsilon$  is non-monotonic: the positive stencil area decreases and then increases considering an increasing shape factor from  $\varepsilon = 0.1$  to  $\varepsilon = 10$ . This is unwanted because “small”  $\varepsilon$  will produce numerical instabilities when solving interpolation Eq. (9), while “big”  $\varepsilon$  will produce bad interpolants [43, 44]. Nonetheless,  $\varepsilon$  has to be chosen on the basis of discretization properties for a specific problem, and therefore it is not a tunable parameter [48].

510 Lastly, we investigate the effect of the node distribution on stencils positivity for 2D node distributions with  $N = 10^5$  nodes for Case 1 and Case 2Mix. The

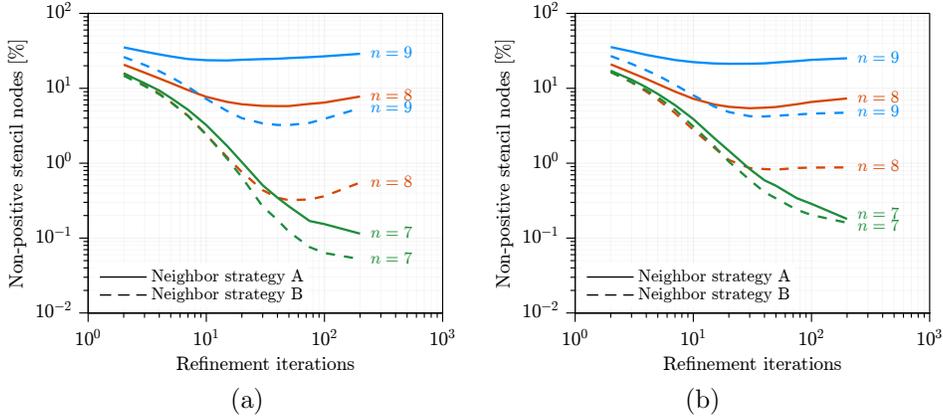


Figure 7: Effect of refinement iterations on stencils positivity for Case 1 (a) and Case 2Mix (b),  $N = 10^5$  nodes and  $\bar{\varepsilon} = 1$ .

fraction of nodes with non-positive stencils is reported in Figure 7 as a function of the number of refinement iterations in the node generation algorithm [37], for  $n = 7, 8, 9$  and two strategies for the choice of RBF neighboring support nodes:

- 515 • Neighbor strategy A:  $n$  nearest neighboring nodes are always chosen;
- Neighbor strategy B: all the nodes within a circle of radius proportional to  $s(\mathbf{x})\sqrt{n}$  are chosen.

The difference between the two previous strategies is that the number of support nodes remains  $n$  with Neighbor strategy A, while with Neighbor strategy B the number of support nodes is  $n$  only on average and a small deviation from  $n$  (1 or 2 nodes) can occur depending upon the local node arrangement.

The curves reported in Figure 7 reveal that the refinement iterations have a positive effect in the reduction of the fraction of non-positive stencils for  $n = 7$  only, while for  $n = 8, 9$  this effect is limited and non-monotonic. Neighbor strategy B is also more effective than Neighbor strategy A in the reduction of the fraction of non-positive stencils, especially with  $n = 8, 9$ . Nonetheless this strategy is found to produce unacceptable discretization errors and therefore only Neighbor strategy A ( $n$  nearest neighbors) is employed.

To conclude,  $n = 7$  is found to be the most appropriate choice in 2D cases, for which 50 refinement iterations are a reasonable balance between computational effort and an adequate reduction of non-positive stencils. For this choice the fraction of non-positive stencils drops significantly below 1% for both cases 1 and 2, and therefore the presence of these few non-positive stencils destroys the  $M$ -matrix property. However, this is not a necessary condition [36] for the convergence of the proposed MC techniques. The choice  $n = 7$  for the 2D case has also a geometrical interpretation since each node is surrounded by 6 nearest neighbours in an isotropic 2D node distribution, i.e., with hexagonal arrange-

Table 1: MC parameters.

Parameter	Symbol	ACMC	SRMC	Range
- MC pre/post smoothing SOR iterations	$p = p'$	1**	1**	$\mathcal{N}^+$
- SOR relaxation factor	$\omega$	1.2*	1.2*	[1, 1.5]
- ACMC overcorrection factor	$t$	variable	–	[1, 3]
- SRMC under-relaxation smoothing factor	$\omega_S$	–	variable	[0, 1]
- SOR iteration multiplier at coarse levels	$r_I$	1*	1*	[1, 2]

\* unless otherwise specified.

\*\* at finest level.

ments. This interpretation allows an extension to 3D isotropic node arrangements where each node is surrounded by 12 nearest neighbours, corresponding to the choice  $n = 13$  [49], which is employed in the 3D case.

#### 4.4. Preconditioning

Similarly to the traditional MG methods, both the proposed MC techniques can be used as standalone solvers and as preconditioners for traditional iterative solvers. As suggested by Stüben [18], the coupling of MG methods as preconditioners with reliable and robust iterative methods is more effective than trying to fine-tune a standalone MG solver, especially in cases where the exact convergence proofs cannot be stated. This is our situation because of the presence of non-positive stencils.

When the proposed MC techniques are used as preconditioners, then the BiCGSTAB method [13] is used as an iterative solver. Such iterative solver is one of the most common choices for the solution of nonsymmetric linear systems. The resulting approaches are denoted by BiCGSTAB/ACMC and BiCGSTAB/SRMC.

#### 4.5. Multicloud parameters

The working parameters that completely define the proposed MC approaches are briefly summarized in Table 1, where  $r_I$  is the SOR iteration multiplier at coarse levels, which defines an amplification factor for the number of MC pre-smoothing SOR iterations from a MC level  $l$  to the successive coarse level  $l + 1$ :  $p_{l+1} = r_I p_l$ . Therefore  $r_I$  defines the amount of smoothing work at coarse levels.

The coarsest level is assumed to be reached when the number of equations at this level is less than 4000, and the corresponding linear system is solved by direct  $LU$  solution.

#### 4.6. Work count and residual norm

565 The comparison of the amount of work, i.e., the number of floating point operations in different algorithms is measured in terms of work units (WU), which is defined as the amount of work needed for one residual evaluation at the finest level. For the MC techniques, the amount of work is automatically evaluated by considering the pre/post smoothing SOR iterations, residual evaluation, 570 restriction and interpolation phases. For BiCGSTAB solver, each full iteration is composed by two semi-iterations, each of which has a cost of 1 WU+10M operations [13], where  $M$  is the number of internal nodes (number of fine-level equations).

The convergence histories are plotted in terms of normalized RMS residual, 575 which is defined as  $\|\mathbf{r}\|_2/\|\mathbf{b}\|_2$ , where  $\mathbf{b}$  is the right hand side term in Eq. (14) and  $\mathbf{r}$  the corresponding residual vector; a null vector is employed as a starting solution vector.

#### 4.7. Implementation details

The implementation of the presented procedures has been done through 580 MATLAB<sup>®</sup> environment using MATLAB linked MEX functions which are compiled from C source code for the computational expensive tasks, on a modern laptop equipped with an Intel<sup>®</sup> i7 2.6GHz processor with 4 cores. Multi-core parallelism is achieved by using OpenMP<sup>®</sup> API for the C source code employed in node generation, MC setup phase, SOR smoothing iterations, residual evaluation, 585 interpolation and restriction. Some of the remaining MATLAB operations, including BiCGSTAB algorithm, are natively parallelized on all available cores by MATLAB.

#### 4.8. Error norm

590 The comparison between the computed solution  $v$  and the corresponding analytical solution  $u$  in 2D cases is done by computing the normalized RMS norm of the error:

$$\begin{aligned} \text{Normalized RMS error} &= \sqrt{\frac{1}{A(\Omega)} \int_{\Omega} \left( \frac{u - v}{u_{max} - u_{min}} \right)^2 d\Omega} \approx \\ &\approx \sqrt{\frac{1}{A(\Omega)} \sum_{i=1}^N \left( \frac{u(\mathbf{x}_i) - v_i}{u_{max} - u_{min}} \right)^2 s^2(\mathbf{x}_i)} \end{aligned} \quad (29)$$

where  $A(\Omega)$  is the area of  $\Omega$ . The integral has been approximated by numerical quadrature using the prescribed spacing function  $s^2$  as quadrature weight.

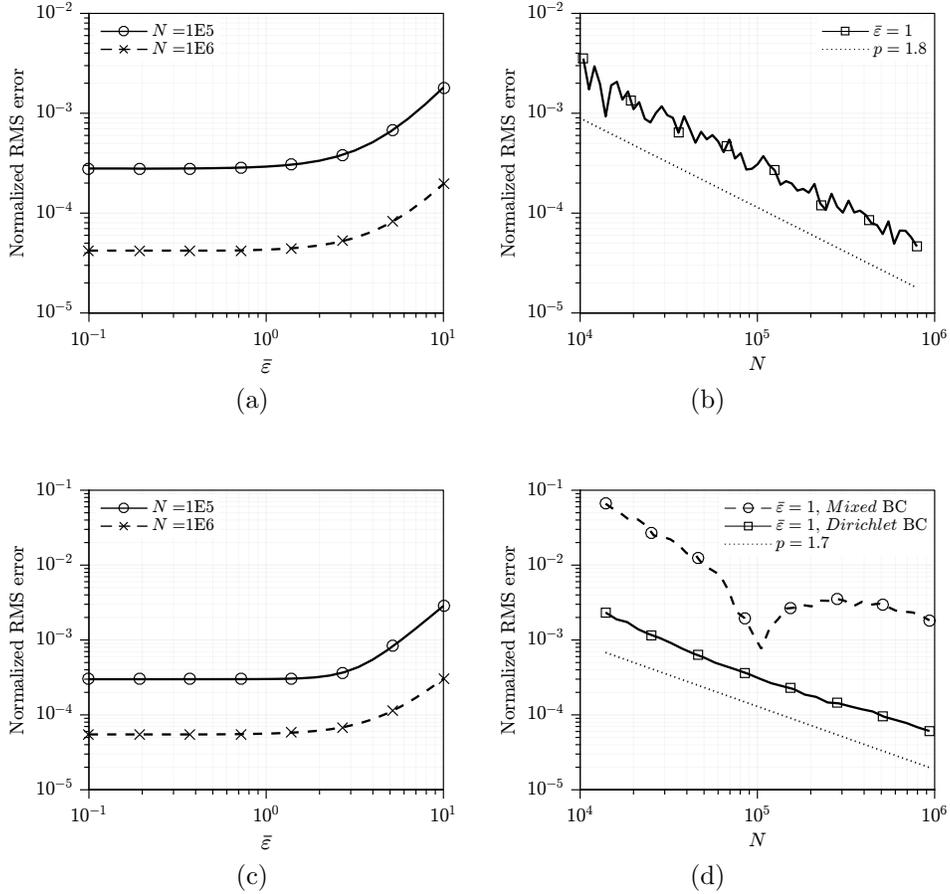


Figure 8: Normalized RMS error as function of rescaled shape factor  $\bar{\epsilon}$ , Case 1 (a) and Case 2Dir (c), and as function of number of the nodes  $N$ , Case 1 (b) and Case 2 (d).

595 **5. Results**

5.1. *Preliminary analyses*

First of all, we investigated the effect of the rescaled shape factor  $\bar{\epsilon} \in [0.1, 10]$  on the normalized RMS error as reported in Figures 8a and 8c for both 2D domains and for  $N = 10^5$  and  $N = 10^6$  nodes. As expected, the error reduces as  $\bar{\epsilon}$  decreases, while below  $\bar{\epsilon} = 1$  the error reaches an asymptotic behaviour and therefore  $\bar{\epsilon} = 1$  has been chosen for the following results, including the 3D case. The choice  $\bar{\epsilon} < 1$  would also imply numerical stability issues for large  $N$  in the local RBF interpolation phase.

600 Figures 8b and 8d show the convergence curves in the case  $\bar{\epsilon} = 1$  for both

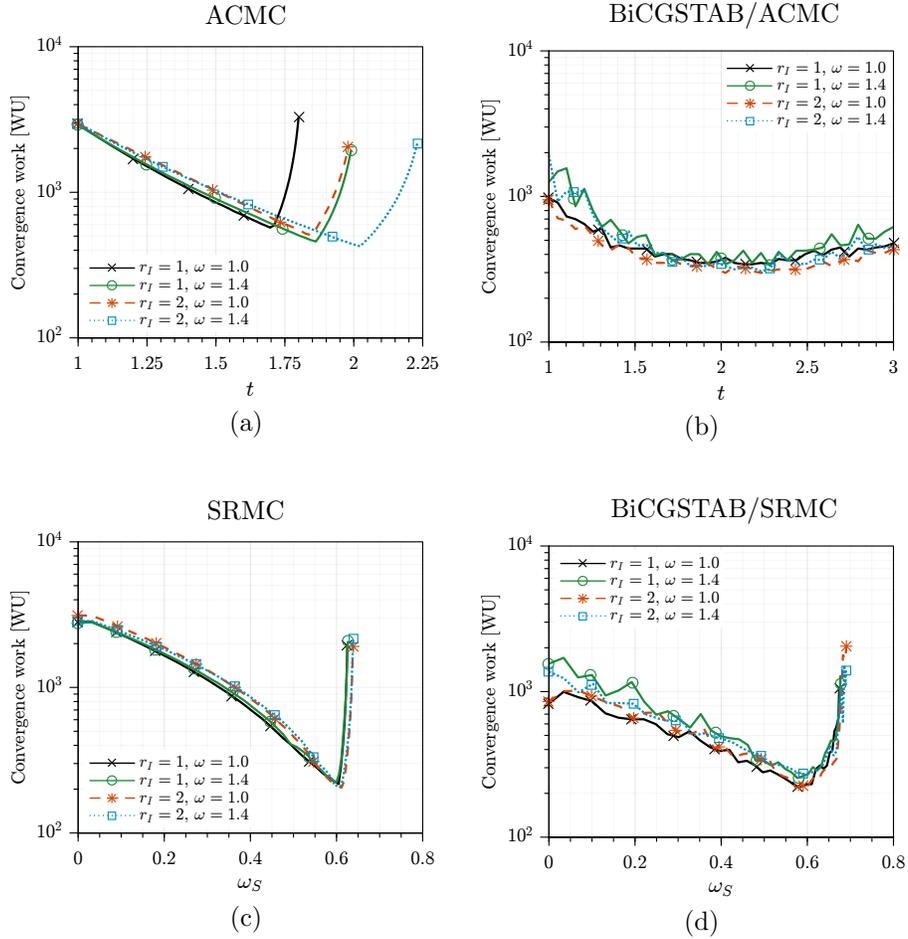


Figure 9: Effect of APMC overcorrection parameter  $t$  (a,b) and SRMC smoothing factor  $\omega_S$  (c,d) on the convergence work,  $N = 10^5$  nodes, Case 1.

605 domains, revealing an order of accuracy<sup>1</sup>  $p \approx 1.8$  for Case 1 and  $p \approx 1.7$  for Case 2Dir. When smoother analytical solutions are employed, as reported in [38], the order of accuracy is always  $p \approx 2.0$ . Convergence curves for Case 2Mix reveal also that the present RBF-FD approach is sensitive to the imposition of Neumann BCs.

610 5.2. Case 1

5.2.1. Multicloud parameters

The influence of MC parameters on the convergence work, i.e., the work needed to reach a normalized RMS residual less than  $10^{-14}$ , is reported in Figure 9 for ACMC and SRMC both employed as standalone solvers and as preconditioners for BiCGSTAB in the case  $N = 10^5$  nodes.

Figure 9a shows that for the standalone ACMC the convergence work decreases with the overcorrection factor  $t$  as expected, until a minimum is reached for an optimal value of  $t$  depending upon the choice for SOR iteration multiplier  $r_I$  and SOR relaxation factor  $\omega$ . Beyond this optimal value, which lies in the range  $[1.7, 2.0]$  for the MC parameters considered here, the convergence work quickly increases and therefore an accurate choice for  $t$  is crucial. The convergence work for ACMC used as preconditioner for BiCGSTAB is reported in Figure 9b, where the almost flat curves reveal that this strategy is more effective and robust than the standalone ACMC strategy because its sensitivity to MC parameters is very low, especially for an overcorrection factor  $t \in [1.5, 3.0]$ .

Convergence work for standalone SRMC is reported in Figure 9c where it can be observed that the effect of  $r_I$  and  $\omega$  is very limited and therefore there is a “single” optimal smoothing factor  $\omega_S \approx 0.6$ , beyond which the growth of the convergence work is very steep. Once again, the effect of using SRMC as preconditioner for BiCGSTAB, shown in Figure 9d, is beneficial since the curves show a smoothed behaviour near the optimal  $\omega_S \approx 0.6$ , but the steep growth beyond this optimal value is present as well.

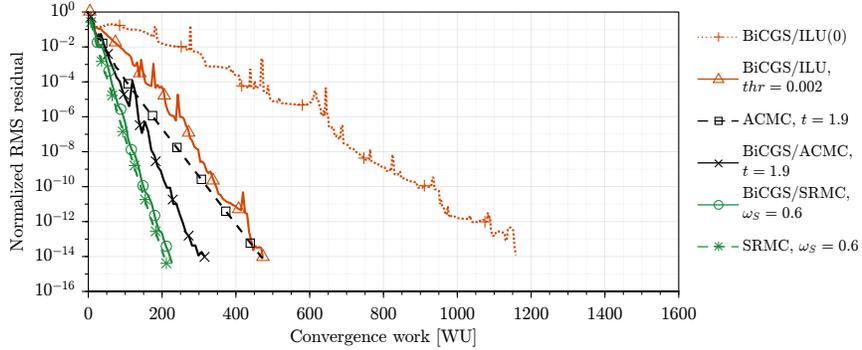
An initial comparison between the previous strategies reveals that the preconditioned versions are always convenient. BiCGSTAB/ACMC strategy is less sensitive than BiCGSTAB/SRMC to the choice of MC parameters, while BiCGSTAB/SRMC strategy can offer a slightly smaller optimal convergence work than BiCGSTAB/ACMC.

5.2.2. Multicloud results

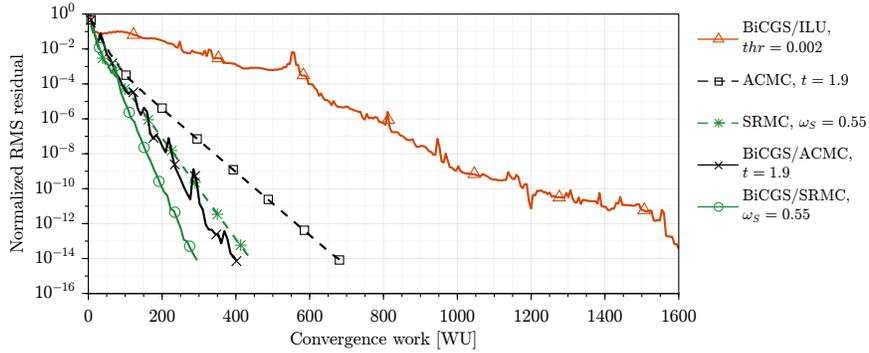
Figure 10 shows the convergence histories in the case of  $N = 10^5$  and  $N = 10^6$  nodes for all the MC strategies previously proposed and also for BiCGSTAB with incomplete LU factorization (ILU) preconditioning and reverse Cuthill-McKee ordering [50]. In the case of  $N = 10^5$  nodes, Figure 10a, each MC strategy has shown to be much more effective in the reduction of the residual than BiCGSTAB with ILU(0) preconditioning, i.e., 0 level of fill in. Comparable performance between ACMC and BiCGSTAB/ILU can be obtained employing a small ILU factorization drop tolerance  $thr = 0.002$ , which implies non-negligible memory requirements and time consumptions in the initial factorization phase. In the case of  $N = 10^6$  nodes, Figure 10b, shows that BiCGSTAB/ILU(0) is practically unfeasible (its convergence history is not even reported since it converged in much more than 1600 WU), the convergence work

---

<sup>1</sup>For 2D cases the order of accuracy  $p$  is defined by a normalized RMS error proportional to  $N^{-p/2}$



(a)



(b)

Figure 10: Comparison of convergence histories for  $N = 10^5$  nodes (a) and  $N = 10^6$  nodes (b), Case 1.

for BiCGSTAB/ILU with the same drop tolerance  $thr = 0.002$  grows rapidly over 1600 WU while the convergence work for the MC strategies show a moderate growth which is subsequently analyzed in a greater detail.

The convergence work for each MC strategy and also for two BiCGSTAB/ILU cases ( $thr = 0.002$  and  $thr = 0.0005$ , the latter requiring twice the memory required by the former) is reported in Figure 11 for  $N$  ranging from  $2 \cdot 10^4$  to  $8 \cdot 10^6$ . The most evident fact is that the growth of convergence work for both BiCGSTAB/ILU cases is considerably larger than the MC ones, and such strategies become uncompetitive with MC strategies for  $N > 10^5$  nodes, taking also account of the fact that drop tolerances below  $thr = 0.0005$  are practically unfeasible due to high memory requirements for such large problem sizes. For  $N < 10^5$  nodes both BiCGSTAB/ILU cases are comparable or even better than the proposed MC approaches.

BiCGSTAB/SRMC with  $\omega_S = 0.55$  turns out to be the most efficient strategy, whose convergence work grows very slowly from 300 WU for  $N = 2 \cdot 10^4$

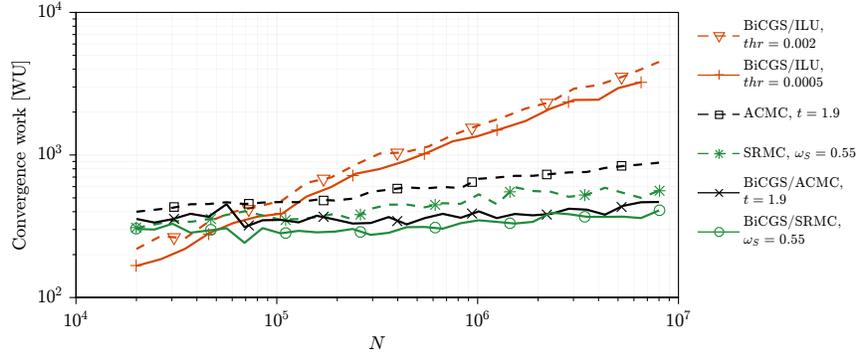


Figure 11: Convergence work vs number of nodes  $N$ , Case 1.

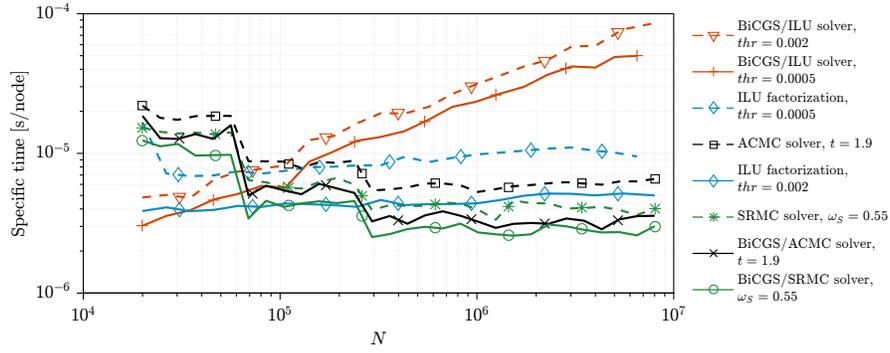


Figure 12: Specific time vs number of nodes  $N$ , Case 1.

nodes to 400 WU for  $N = 8 \cdot 10^6$  nodes. BiCGSTAB/ACMC with  $t = 1.9$  performs almost identically. The less efficient strategy, as expected, is standalone ACMC with  $t = 1.9$ , whose convergence work grows from 400 WU for  $N = 2 \cdot 10^4$  nodes to 900 WU for  $N = 8 \cdot 10^6$  nodes.

670 The comparison between the specific times, i.e., time per node, is reported  
in Figure 12 for each of the previous MC strategies by considering the time  
required to reach a normalized RMS residual less than  $10^{-14}$ , and also for two  
675 ILU factorization cases ( $thr = 0.002$  and  $thr = 0.0005$ ) considering the initial  
factorization time. 8 OpenMP threads have been employed for the parallelized  
C code. Similarly to the behaviour of convergence work, specific times for  
both BiCGSTAB/ILU cases show a larger growth than MC ones and become  
uncompetitive compared to each MC strategy beyond  $N = 2 \cdot 10^5$  nodes,  
while the specific times for MC strategies turn out to be almost constant for  
 $N > 3 \cdot 10^5$ . Again, BiCGSTAB/SRMC with  $\omega_S = 0.55$  turns out to be the  
680 best (faster) strategy, requiring approximately  $3 \cdot 10^{-6}$  s/node for  $N > 3 \cdot 10^5$ .  
BiCGSTAB/ACMC with  $t = 1.9$  performs almost identically, while standalone

ACMC with  $t = 1.9$  is the less performing and shows a very moderate growth in specific time for high  $N$ . The speedup between a traditional approach like BiCGSTAB/ILU with drop tolerance  $thr = 0.0005$  and BiCGSTAB/SRMC with  $\omega_S = 0.55$  ranges from 4 for  $N = 3 \cdot 10^5$  nodes, to almost 20 for  $N = 8 \cdot 10^6$  nodes.

The specific times for the remaining main tasks are almost constant and are as follows:  $5 \cdot 10^{-6}$  s/node for node generation (50 refinement iterations),  $3.5 \cdot 10^{-7}$  s/node for matrix coefficients calculation (discretization) and  $1.5 \cdot 10^{-7}$  s/node for the MC setup phase (coarsening and coarse-level coefficients calculation).

### 5.3. Case 2

#### 5.3.1. Multicloud results

The convergence work for each MC strategy and also for two BiCGSTAB/ILU cases ( $thr = 0.002$  and  $thr = 0.0005$ ) is reported in Figure 13 for  $N$  ranging from  $2 \cdot 10^4$  to  $8 \cdot 10^6$ , for both Case 2Dir and Case 2Mix. Again, as in Case 1, the most evident fact is that the growth rate of convergence work for BiCGSTAB/ILU cases is considerably larger than the MC ones for both types of BCs. In Case 2Dir, Figure 13a, MC strategies outperform BiCGSTAB/ILU strategies only for  $N > 4 \cdot 10^5$  nodes, while in Case 2Mix, Figure 13b, MC strategies outperform BiCGSTAB/ILU strategies from  $N > 6 \cdot 10^4$  nodes already. Both BiCGSTAB/ILU strategies turned out to be unfeasible due to both high time consumption and large memory requirements for  $N > 2 \cdot 10^6$  nodes in Case 2Mix.

BiCGSTAB/SRMC with  $\omega_S = 0.55$  is again the most efficient strategy for both BCs and BiCGSTAB/ACMC with  $t = 1.9$  perform almost identically. In Case 2Dir the convergence work for these strategies is almost constant ( $\approx 350$  WU), while in Case 2Mix the convergence work shows a moderate growth from 300 WU for  $N = 2 \cdot 10^4$  nodes to 550 WU for  $N = 8 \cdot 10^6$  nodes. Therefore the less efficient strategies are again standalone ACMC and SRMC, especially in Case 2Dir where their convergence work shows a moderate but constant growth from  $N = 3 \cdot 10^5$  nodes.

The comparison between the specific times is reported in Figures 14 and 15 for each of the previous MC strategies (normalized RMS residual less than  $10^{-14}$ ), and also for two ILU factorizations ( $thr = 0.002$  and  $thr = 0.0005$ ) for both cases 2Dir and 2Mix. Also in this case 8 OpenMP threads have been employed for the parallelized C code. Similarly to the behaviour of the convergence work, specific times for BiCGSTAB/ILU cases show a larger growth than MC ones and become uncompetitive compared to each MC strategy beyond  $N = 2 \cdot 10^5$  nodes in Case 2Dir, (Figure 14), while in Case 2Mix, (Figure 15), the MC strategies outperforms BiCGSTAB/ILU strategies from  $N = 7 \cdot 10^4$  nodes already. Specific times for all MC strategies are practically constant for  $N > 3 \cdot 10^5$  nodes. The specific times for the remaining main tasks (node generation, matrix coefficients calculation and MC setup phase) remain the same as in Case 1.

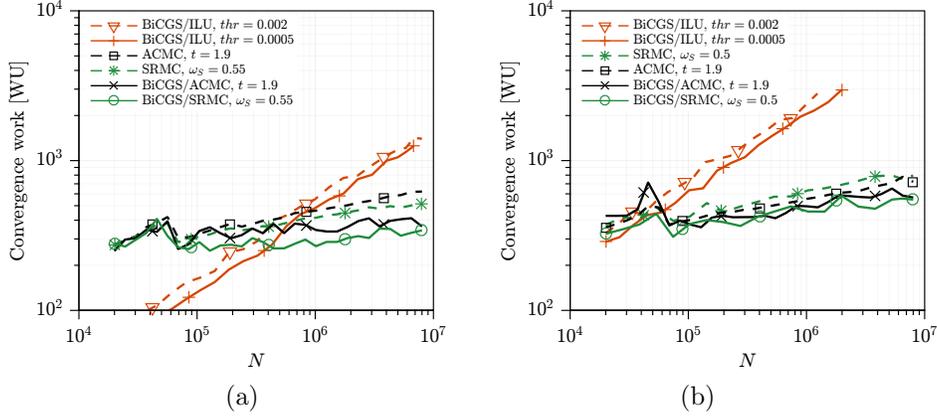


Figure 13: Convergence work vs number of nodes  $N$  for Case 2Dir (a) and Case 2Mix (b).

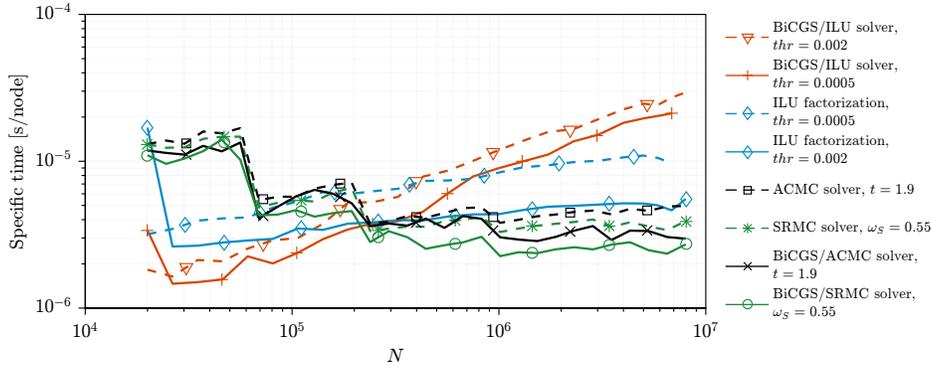


Figure 14: Specific time vs number of nodes  $N$  for Case 2Dir.

The speedup between a traditional approach like BiCGSTAB/ILU with drop tolerance  $thr = 0.0005$  and BiCGSTAB/SRMC with  $\omega_S = 0.55$  ranges from 4 for  $N = 10^6$  nodes, to 8 for  $N = 8 \cdot 10^6$  nodes, in Case 2Dir. In Case 2Mix such speedup ranges from 2 for  $N = 10^5$  nodes, to 10 for  $N = 2 \cdot 10^6$  nodes. Therefore the use of MC strategies can bring great advantages especially in case of Neumann BCs, which is a typical condition encountered in many practical applications such as Poisson problems in computational fluid dynamics. Furthermore, in these cases the proposed MC strategies are able to deal with very large problems which can not be easily tackled by traditional BiCGSTAB/ILU approaches.

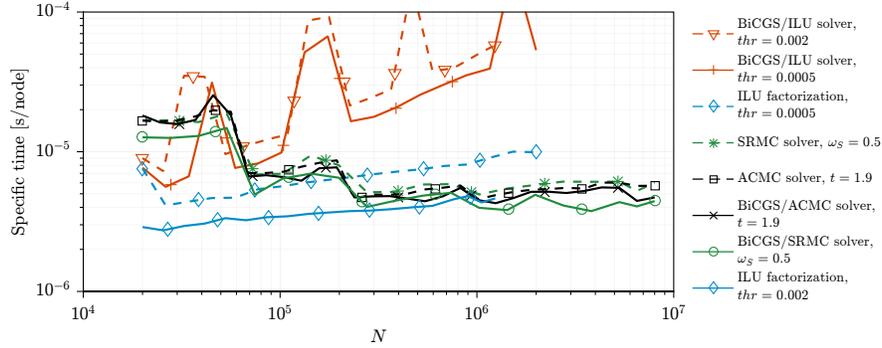


Figure 15: Specific time vs number of nodes  $N$  for Case 2Mix.

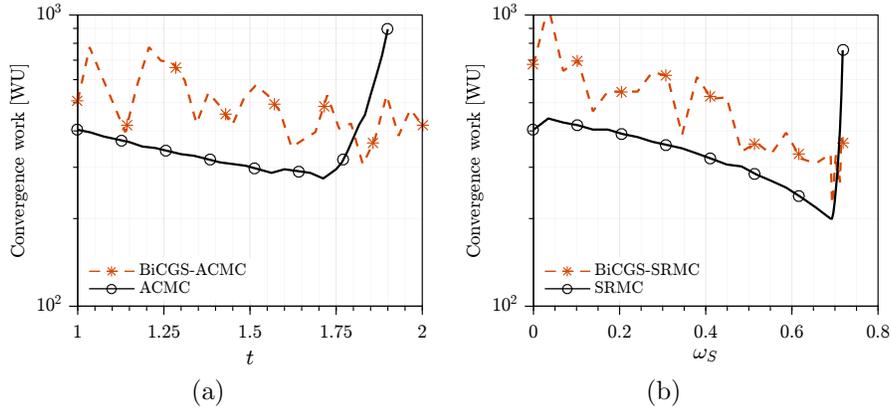


Figure 16: Effect of ACMC overcorrection parameter  $t$  (a) and SRMC smoothing factor  $\omega_S$  (b) on the convergence work,  $N = 2 \cdot 10^6$  nodes, Case 3.,  $\omega = 1.5$ .

#### 5.4. Case 3

##### 5.4.1. Multicloud parameters

The influence of MC parameters on the convergence work is reported in Figure 16 for ACMC and SRMC when both are employed as standalone solvers  
740 and as preconditioners for BiCGSTAB, using  $N = 2 \cdot 10^6$  nodes and a SOR relaxation factor  $\omega = 1.5$ .

Figure 16a reveals a limited influence of the overcorrection parameter  $t$  on the reduction of the convergence work for both standalone and preconditioned ACMC. The minimum convergence work for the standalone ACMC is obtained  
745 for  $t = 1.6 - 1.7$ , while the convergence work for the preconditioned ACMC appears to be almost independent from  $t$ . Nonetheless, the minimum convergence work for both standalone and preconditioned ACMC is approximately 300 WU, which is very similar to the convergence work obtained in the corresponding Case 2Dir for the same number of nodes  $N = 2 \cdot 10^6$ .

750 The effect of the under-relaxation smoothing factor  $\omega_S$  on the convergence work is shown in Figure 16b for both standalone and preconditioned SRMC. Similarly to 2D cases, the increase in  $\omega_S$  allows a reduction of the convergence work whose minimum is found for  $\omega_S = 0.65 - 0.70$  and it is approximately 200 WU for both strategies, with a more regular behaviour in the standalone case. Therefore the  
 755 SRMC technique is less sensitive to the coarsening strategy and more effective than ACMC. This result is in perfect accordance with 2D cases.

#### 5.4.2. Multicloud results

The convergence work for each MC strategy and also for a BiCGSTAB/ILU case ( $thr = 0.03$ ) is reported in Figure 17 for  $N$  ranging from  $10^5$  to  $8 \cdot 10^6$  nodes.  
 760 By considering the analyses of Subsection 5.4.1, the ACMC overcorrection parameter is chosen to be  $t = 1.6$  while the SRMC under-relaxation smoothing factor is chosen to be  $\omega_S = 0.65$ .

In the 3D case the advantages of MC techniques over the traditional BiCGSTAB/ILU solver still hold, although they are less evident than the ones obtained in 2D cases: BiCGSTAB/ILU is more effective than any MC technique for  $N < 10^6$ ,  
 765 while beyond this point both SRMC strategies allow a certain advantage, especially when using the standalone strategy. Conversely, both ACMC techniques are always less efficient than BiCGSTAB/ILU in the range of  $N$  considered here. Despite the unfavorable comparison of the convergence work between  
 770 ACMC and BiCGSTAB/ILU, both preconditioned ACMC and SRMC show a smaller growth rate when increasing  $N$  compared to BiCGSTAB/ILU, as well as for standalone ACMC and SRMC although in a limited extent. This suggests that when a very large number of nodes is employed, e.g.,  $N > 10^7$ , preconditioned ACMC and SRMC could bring greater advantages over the traditional  
 775 BiCGSTAB/ILU solver in terms of computational efficiency, although this claim has not been investigated in the present work.

The comparison between the specific times is reported in Figure 18 for each of the previous MC strategies by considering the time required to reach a normalized RMS residual less than  $10^{-14}$ , and also for a ILU factorization with  
 780  $thr = 0.03$  considering the initial factorization time. Also here 8 OpenMP threads have been employed for the parallelized C code. Similarly to the behaviour of the convergence work, the specific time for the BiCGSTAB/ILU solver shows an overall growth rate which is slightly larger than the growth rate of MC strategies, while for  $N > 3 \cdot 10^6$  nodes the BiCGSTAB/ILU growth rate is similar to the standalone ACMC and SRMC. However, both standalone techniques  
 785 perform better than BiCGSTAB/ILU for  $N > 4 \cdot 10^5$  and the corresponding speedup values for SRMC, which is the fastest, ranges from 2 for  $N = 4 \cdot 10^5$  nodes to 2.7 for  $N = 8 \cdot 10^6$  nodes. Similarly to the behaviour of the convergence work, the preconditioned MC strategies are slower than the corresponding standalone version, but they exhibit a smaller growth rate, especially for the SRMC  
 790 technique. Therefore greater speedup values are reasonably expected for large number of nodes beyond  $N = 10^7$ .

The specific times for the remaining main tasks are almost constant and are as follows:  $1.6 \cdot 10^{-5}$  s/node for node generation (50 refinement iterations),  $4 \cdot 10^{-6}$

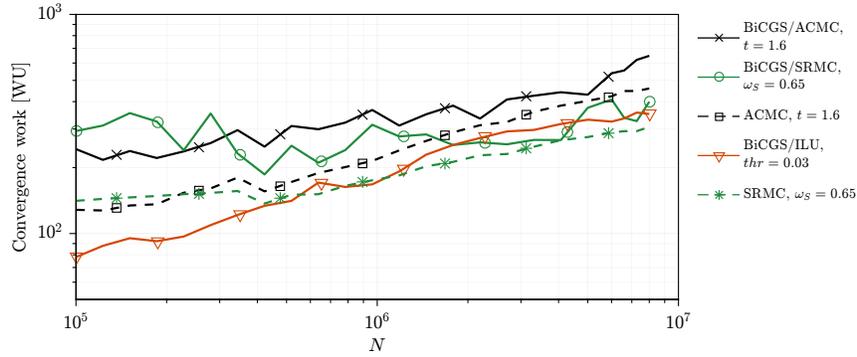


Figure 17: Convergence work vs number of nodes  $N$ , Case 3.

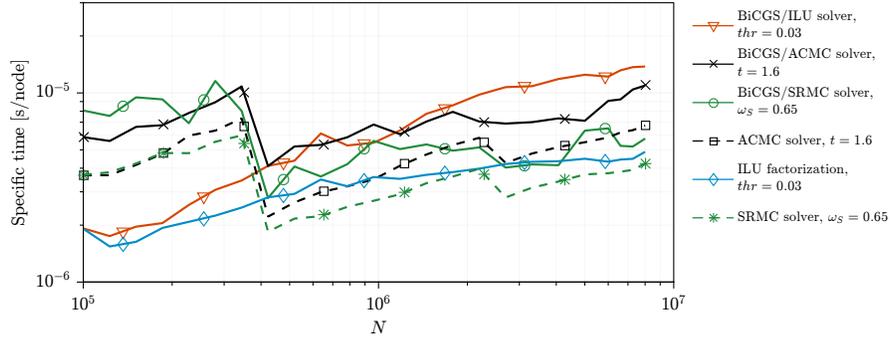


Figure 18: Specific time vs number of nodes  $N$ , Case 3.

795 s/node for matrix coefficients calculation (discretization) and  $1.0 \cdot 10^{-6}$  s/node for the MC setup phase (coarsening and coarse-level coefficients calculation).

### 5.5. OpenMP speedup

The measured speedup values obtained using 2, 4 and 8 OpenMP threads for the parallelized C code for the MC algorithms are reported in Table 2 for Case 1 and cases 2Dir and 2Mix where  $N = 10^6$  nodes. The speedup values are listed for each of the main tasks separately.

805 From the previous tables it can be observed that the refinement phase for the node generation and the matrix coefficients calculation are the tasks that exhibit the largest speedups (4.3 – 4.7 with 8 OpenMP threads), while the achieved speedup for the MC solvers is lower and does not exceed 2.7. In particular, the speedup for BiCGSTAB with MC preconditioning is always lower than the speedup for the corresponding standalone MC solver because BiCGSTAB is always parallelized on all available cores in MATLAB by default, as well as

810 other vector operations (vector sums) that are carried out in MATLAB in the present implementation.

Table 2: Speedup values.

OpenMP threads	Ref	ACMC			SRMC		
		Cff	Slv-A	Bi/A	Cff	Slv-S	Bi/S
Case 1, $N = 10^6$ nodes							
2	1.9	1.9	1.9	1.6	1.9	1.6	1.5
4	2.9	3.7	2.5	2.0	2.6	2.2	2.0
8	4.5	4.7	2.7	2.1	4.4	2.4	2.3
Case 2Dir, $N = 10^6$ nodes							
2	2.0	1.8	1.6	1.5	1.8	1.8	1.5
4	3.0	2.7	2.1	1.9	2.7	2.4	2.0
8	4.7	4.3	2.3	1.9	4.4	2.6	2.0
Case 2Mix, $N = 10^6$ nodes							
2	2.0	1.8	1.8	1.6	1.9	1.7	1.6
4	3.0	3.1	2.3	2.1	2.8	2.3	2.2
8	4.7	4.3	2.5	2.2	4.3	2.4	2.1

**Ref** = node refinement phase (50 iterations)

**Cff** = matrix coefficients phase (meshless discretization)

**Slv-A, Slv-S** = solution phase with A=ACMC, S=SRMC

**Bi-A, Bi-S** = BiCGSTAB preconditioned with A=ACMC, S=SRMC

## 6. Conclusions

815 In this work several multilevel strategies have been proposed as fast techniques for the solution of linear systems arising from RBF-FD meshless discretizations of Poisson problems. Two new multilevel techniques, i.e., additive correction multilevel (ACMC) and smoothed restriction multilevel (SRMC), have been developed and employed for the solution of 2D and 3D Poisson problems over complex-shaped domains with different boundary conditions. Each of these novel techniques can be used as both standalone solver or as a preconditioner for a typical iterative method such as BiCGSTAB.

820 Extensive analyses are carried out in order to highlight the influence of the various MC parameters on the performance. When a large number of nodes is employed, e.g.,  $N \approx 10^6$  nodes, great advantages over traditional approaches (BiCGSTAB with incomplete LU factorization as preconditioner) have been obtained for 2D cases in both reduction of convergence work and reduction of

required computational time. Such reductions can reach a factor 10 for convergence work and even a factor 20 for computational time in the case of extremely large problem sizes, e.g.,  $N = 8 \cdot 10^6$  nodes. These advantages over traditional approaches have proven to be particularly evident in the case of Neumann boundary conditions, which often characterize many problems of engineering relevance. Furthermore, in cases where Neumann boundary conditions are employed, the proposed MC strategies have proven to be able to deal with very large size problems where traditional solver approaches can not deal with because of excessive memory and time requirements. In the 3D case the advantages over traditional solver approaches are less evident but still present, especially when comparing the computational time. Furthermore, greater advantages are expected when using very large number of nodes in 3D, e.g.,  $N > 10^7$ . Therefore, the proposed MC strategies are extremely attractive for problems where complex-shaped domains and large number of nodes are employed.

Between the proposed strategies, the use of both ACMC and SRMC as preconditioners for BiCGSTAB showed the best performance, requiring an almost constant specific time for the convergence in 2D cases presented here, meaning that the required computational time grows almost linearly with the size of the problem  $N$ . SRMC allows slightly better performance than ACMC but it is more sensitive to the choice of MC parameters and its implementation is also a little more complex. The use of both MC techniques as preconditioners also allows a reduction of sensitivity to MC parameters in 2D cases.

Possible extensions of this work might be considered, for example, by studying the application of the presented MC techniques to higher order RBF-FD discretizations, i.e., larger stencil sizes  $n > 7$  in 2D and  $n > 13$  in 3D, and by employing different radial basis functions such as polyharmonic splines [6], as well as the extension to different PDEs.

## Acknowledgments

Božidar Šarler was financed by Slovenian Grant Agency (ARRS) under program group P2-0162 and project J2-7384.

## Appendix A. M-matrices

We briefly present the sufficient conditions leading to a  $M$ -matrix structure (see Seibold [36] and Hackbusch [47] for further details) in the context of the present meshless approach, for which the original contribution is given by Theorem 1.

**Definition 1** (*Z-matrices and L-matrices*). *A square matrix  $\mathbf{M} = (m_{ij}) \in \mathcal{R}^{M \times M}$  is called Z-matrix if  $m_{ij} \leq 0 \ \forall i \neq j$ . A Z-matrix is called L-matrix if  $m_{ii} > 0 \ \forall i$ .*

The notation  $\mathbf{M} \geq 0$  indicates that its matrix entries satisfy  $m_{ij} \geq 0 \ \forall i, j$ .

865 **Definition 2** (*M*-matrices). A *Z*-matrix is called *M*-matrix if it is nonsingular and  $\mathbf{M}^{-1} \geq 0$ .

**Definition 3** (Essentially diagonally dominance). A square matrix  $\mathbf{M}$  is called essentially diagonally dominant if it is weakly diagonally dominant ( $|m_{ii}| \geq \sum_{j \neq i} |m_{ij}| \quad \forall i$ ), and every node is connected (directly or indirectly) through stencil entries to a node  $k$  which satisfies the strict diagonal dominance relation  
870  $|m_{kk}| > \sum_{j \neq k} |m_{kj}|$ .

**Definition 4** (Positive stencil). A stencil is called positive if its coefficients  $g_1, \dots, g_m$  satisfy  $g_1 > 0$  and  $g_2, \dots, g_m \leq 0$ .

Benefits on the use of positive stencils in the context of generalized-FD  
875 methods can be found in Demkowicz et al [10].

**Theorem 1.** Matrix  $\mathbf{A}_I$  of Eq. (14) is essentially diagonally dominant if at least one Dirichlet node is employed and if positive stencils are employed for every internal node.

*Proof.* Consider the interpolation system of Eq. (9) with constant data, i.e.,  
880  $\mathbf{v}^T = (v_1, \dots, v_m) = (1, \dots, 1)$  and  $\mathbf{f} = \mathbf{0}$ , where Neumann/Dirichlet nodes can be included. Since such interpolation is exact for constant data, we get  $\mathbf{a} = \mathbf{0}$ ,  $\mathbf{b} = \mathbf{0}$ ,  $c = 1$  and therefore  $v(\mathbf{x}) = 1$  from Eq. (2). Then, Eq. (12) with  $q(\mathbf{x}) = -\nabla^2 v(\mathbf{x}) = 0$  becomes  $\sum_{i=1}^m g_i = 0$ , where  $g_1, \dots, g_m$  are the stencil coefficients. Since the stencil is positive, the previous relation can be written as  $|g_1| =$   
885  $\sum_{i=2}^m |g_i|$ . If the stencil has no Dirichlet nodes, the last relation implies a weakly diagonally dominant row in terms of matrix coefficients, otherwise a strictly diagonally dominant row is obtained since  $|g_1| = \sum_{i=2}^m |g_i| > \sum_{i \in I} |g_i|$  where  $I$  is a subset of  $\{2, \dots, m\}$  identifying the non-Dirichlet stencil contributions, i.e., the ones corresponding to internal nodes. Finally, in the present RBF-FD  
890 approach every node is always directly or indirectly connected to a Dirichlet node through stencil entries because of the employed node distributions and strategy for the choice of stencil support nodes (nearest neighbors).  $\square$

**Theorem 2.** An essentially diagonally dominant *L*-matrix is an *M*-matrix.

*Proof.* See Hackbusch [47], pp. 154-155.  $\square$

895 The coefficient matrix  $\mathbf{A}_I$  of Eq. (14), resulting from the present RBF-FD discretization, is an *M*-matrix if the conditions of Theorem 1 hold.

## References

- [1] P. Frey, P. George, Mesh Generation: Application to Finite Elements, 2nd Edition, ISTE Ltd and John Wiley & Sons, UK and USA, 2010.  
900 doi:10.1002/9780470611166.  
URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470611166>

- 905 [2] H. Li, S. Mulay, Meshless Methods and Their Numerical Properties, CRC Press, Boca Raton, Florida, USA, 2013. doi:10.1201/b14492.  
URL <https://www.taylorfrancis.com/books/9781466517479>
- [3] B. Fornberg, N. Flyer, Solving PDEs with radial basis functions, Acta Numer. 24 (2015) 215–258. doi:10.1017/S0962492914000130.  
URL <http://www.journals.cambridge.org>
- 910 [4] E. Bollig, N. Flyer, G. Erlebacher, Solution to PDEs using radial basis function finite-differences (RBF-FD) on multiple GPUs, J. Comput. Phys. 231 (21) (2012) 7133–7151. doi:<https://doi.org/10.1016/j.jcp.2012.06.030>.  
URL <http://www.sciencedirect.com/science/article/pii/S0021999112003452>
- 915 [5] M. Tilenius, E. Larsson, E. Lehto, N. Flyer, A scalable rbf-fd method for atmospheric flow, J. Comput. Phys. 298 (2015) 406–422. doi:<https://doi.org/10.1016/j.jcp.2015.06.003>.  
URL <http://www.sciencedirect.com/science/article/pii/S0021999115003824>
- 920 [6] V. Bayona, N. Flyer, B. Fornberg, G. Barnett, On the role of polynomials in RBF-FD approximations: II. Numerical solution of elliptic PDEs, J. Comput. Phys. 332 (2017) 257–273. doi:<https://doi.org/10.1016/j.jcp.2016.12.008>.  
URL <http://www.sciencedirect.com/science/article/pii/S0021999116306490>
- 925 [7] N. Flyer, G. Barnett, L. Wicker, Enhancing finite differences with radial basis functions: Experiments on the Navier-Stokes equations, J. Comput. Phys. 316 (2016) 39–62. doi:<https://doi.org/10.1016/j.jcp.2016.02.078>.  
URL <http://www.sciencedirect.com/science/article/pii/S0021999116300195>
- 930 [8] N. Flyer, B. Fornberg, V. Bayona, G. Barnett, On the role of polynomials in RBF-FD approximations: I. Interpolation and accuracy, J. Comput. Phys. 321 (2016) 21–38. doi:<https://doi.org/10.1016/j.jcp.2016.05.026>.  
URL <http://www.sciencedirect.com/science/article/pii/S0021999116301632>
- 935 [9] M. Dehghan, M. Abbaszadeh, The use of proper orthogonal decomposition (POD) meshless RBF-FD technique to simulate the shallow water equations, J. Comput. Phys. 351 (2017) 478–510. doi:<https://doi.org/10.1016/j.jcp.2017.09.007>.  
URL <http://www.sciencedirect.com/science/article/pii/S0021999117306575>
- 940

- [10] L. Demkowicz, A. Karafiat, T. Liszka, On some convergence results for FDM with irregular mesh, *Comput. Method. Appl. M.* 42 (3) (1984) 343–355. doi:10.1016/0045-7825(84)90013-6.  
 945 URL <http://linkinghub.elsevier.com/retrieve/pii/S0045782584900136>
- [11] B. Šarler, R. Vertnik, Meshfree explicit local radial basis function collocation method for diffusion problems, *Comput. Math. Appl.* 51 (8) (2006) 1269–1282. doi:10.1016/j.camwa.2006.04.013.  
 950 URL <http://linkinghub.elsevier.com/retrieve/pii/S0898122106000836>
- [12] V. Bulgakov, B. Šarler, G. Kuhn, Iterative solution of systems of equations in the dual reciprocity boundary element method for the diffusion equation, *Int. J. Numer. Meth. Eng.* 43 (4) (1998) 713–732. doi:https://doi.org/10.1002/(SICI)1097-0207(19981030)43:4<713::AID-NME445>3.0.CO;2-8.  
 955 URL [https://onlinelibrary.wiley.com/doi/abs/10.1002/\(SICI\)1097-0207\(19981030\)43:4%3C713::AID-NME445%3E3.0.CO;2-8](https://onlinelibrary.wiley.com/doi/abs/10.1002/(SICI)1097-0207(19981030)43:4%3C713::AID-NME445%3E3.0.CO;2-8)
- [13] H. van der Vorst, Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, *SIAM J. Sci. Stat. Comp.* 13 (2) (1992) 631–644. doi:10.1137/0913035.  
 960 URL <http://epubs.siam.org/doi/10.1137/0913035>
- [14] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd Edition, SIAM, Philadelphia, Pennsylvania, 2003, Ch. 10: Preconditioning Techniques. doi:10.1137/1.9780898718003.ch10.  
 965 URL <https://epubs.siam.org/doi/abs/10.1137/1.9780898718003.ch10>
- [15] S. McCormick, *Multigrid Methods*, *Frontiers in Applied Mathematics.*, SIAM, Philadelphia, Pennsylvania, 1987. doi:10.1137/1.9781611971057.  
 970 URL <https://doi.org/10.1137/1.9781611971057>
- [16] A. Katz, A. Jameson, Multicloud: multigrid convergence with a meshless operator, *J. Comput. Phys.* 228 (14) (2009) 5237–5250. doi:10.1016/j.jcp.2009.04.023.  
 975 URL <http://dx.doi.org/10.1016/j.jcp.2009.04.023http://linkinghub.elsevier.com/retrieve/pii/S0021999109001995>
- [17] K. Stüben, *Algebraic multigrid (AMG): an introduction with applications* (Technical Report), Tech. Rep. GMD-70, Institute for Algorithms and Scientific Computing (SCAI), German National Research Center for Information Technology (GMD)., Sankt Augustin, Germany (1999).  
 980 URL [https://www.scai.fraunhofer.de/content/dam/scai/de/documents/AllgemeineDokumentensammlung/SchnelleLoeser/SAMG/AMG\\_Introduction.pdf](https://www.scai.fraunhofer.de/content/dam/scai/de/documents/AllgemeineDokumentensammlung/SchnelleLoeser/SAMG/AMG_Introduction.pdf)

- 985 [18] K. Stüben, A review of algebraic multigrid, *J. Comput. Appl. Math.* 128 (1-2) (2001) 281–309. doi:10.1016/S0377-0427(00)00516-1.  
URL <http://linkinghub.elsevier.com/retrieve/pii/S0377042700005161>
- 990 [19] R. Vertnik, K. Mramor, B. Šarler, Solution of three-dimensional temperature and turbulent velocity field in continuously cast steel billets with electromagnetic stirring by a meshless method, *Eng. Anal. Bound. Elem.* 104 (2019) 347–363. doi:https://doi.org/10.1016/j.enganabound.2019.03.026.  
URL <http://www.sciencedirect.com/science/article/pii/S0955799718305010>
- 995 [20] A. Brandt, Multi-level adaptive solutions to boundary-value problems, *Math. Comput.* 31 (138) (1977) 333–390. doi:10.2307/2006422.  
URL <http://www.jstor.org/stable/2006422?origin=crossref>  
<https://www.jstor.org/stable/2006422?origin=crossref>
- 1000 [21] R. Southwell, *Relaxation Methods in Theoretical Physics*, The Oxford engineering science series. , Oxford University Press, London, 1946. doi:10.1017/S0368393100111915.  
URL <https://doi.org/10.1017/S0368393100111915>
- 1005 [22] E. Stiefel, Über einige methoden der relaxationsrechnung, *Z. Angew. Math. Phys.* 3 (1) (1952) 1–33. doi:10.1007/BF02080981.  
URL <http://link.springer.com/10.1007/BF02080981>
- [23] R. Fedorenko, The speed of convergence of one iterative process, *USSR Comp. Math. Math+* 4 (3) (1964) 227–235. doi:10.1016/0041-5553(64)90253-8.  
URL <http://linkinghub.elsevier.com/retrieve/pii/S0041555364902538>
- 1010 [24] R. Fedorenko, Iterative methods for elliptic difference equations, *Russ. Math. Surv+* 28 (2) (1973) 129–195.  
URL [http://iopscience.iop.org/0036-0279/28/2/R05/pdf/RMS\\_28\\_2\\_R05.pdf](http://iopscience.iop.org/0036-0279/28/2/R05/pdf/RMS_28_2_R05.pdf)
- 1015 [25] F. de la Vallee Poussin, W. Timlake, An accelerated relaxation algorithm for iterative solution of elliptic equations, *SIAM J. Numer. Anal.* 5 (2) (1968) 340–351. doi:10.1137/0705029.  
URL <http://epubs.siam.org/doi/10.1137/0705029>
- 1020 [26] A. Settari, K. Aziz, A generalization of the additive correction methods for the iterative solution of matrix equations, *SIAM J. Numer. Anal.* 10 (3) (1973) 506–521. doi:10.1137/0710046.  
URL <http://epubs.siam.org/doi/10.1137/0710046>

- [27] R. Blaheta, A multilevel method with overcorrection by aggregation for solving discrete elliptic problems, *J. Comput. Appl. Math.* 24 (1-2) (1988) 227–239. doi:10.1016/0377-0427(88)90355-X.  
1025 URL <http://linkinghub.elsevier.com/retrieve/pii/037704278890355X>
- [28] R. Blaheta, A multilevel method with correction by aggregation for solving discrete elliptic problems, *Appl. Math-Czech* 31 (5) (1986) 365–378. doi:10.1016/0377-0427(88)90355-X.  
1030 URL <http://eudml.org/doc/15461>
- [29] B. Hutchinson, G. Raithby, A multigrid method based on the additive correction strategy, *Numer. Heat Tr. A-Appl.* 9 (5) (1986) 511–537. doi:10.1080/10407788608913491.  
1035 URL <http://www.tandfonline.com/doi/abs/10.1080/10407788608913491>
- [30] D. Braess, Towards algebraic multigrid for elliptic problems of second order, *Computing* 55 (4) (1995) 379–393. doi:10.1007/BF02238488.  
URL <http://link.springer.com/10.1007/BF02238488>
- [31] P. Vaněk, Acceleration of convergence of a two-level algorithm by smoothing transfer operators, *Appl. Math-Czech* 37 (4) (1992) 265–274.  
1040 URL <https://eudml.org/doc/15715{#}citations>
- [32] P. Vaněk, J. Mandel, M. Brezina, Algebraic multigrid on unstructured meshes, Tech. rep., University of Colorado at Denver, Denver, CO, USA (1994).  
1045 URL <https://dl.acm.org/citation.cfm?id=865628>
- [33] P. Vaněk, J. Mandel, M. Brezina, Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems, *Computing* 56 (3) (1996) 179–196. doi:10.1007/BF02238511.  
1050 URL <http://link.springer.com/10.1007/BF02238511>
- [34] P. Vaněk, M. Brezina, J. Mandel, Convergence of algebraic multigrid based on smoothed aggregation, *Numer. Math.* 88 (3) (2001) 559–579. doi:10.1007/s211-001-8015-y.  
URL <http://link.springer.com/10.1007/s211-001-8015-y>
- [35] K. Leem, S. Oliveira, D. Stewart, Algebraic multigrid (AMG) for saddle point systems from meshfree discretizations, *Numer. Linear Algebr.* 11 (23) (2004) 293–308. doi:10.1002/nla.383.  
1055 URL <http://doi.wiley.com/10.1002/nla.383>
- [36] B. Seibold, Performance of algebraic multigrid methods for non-symmetric matrices arising in particle methods, *Numer. Linear Algebr.* 17 (2-3) (2009) 433–451. doi:10.1002/nla.710.  
1060 URL <http://arxiv.org/abs/0905.3005>

- [37] R. Zamolo, E. Nobile, Two algorithms for fast 2d node generation: Application to rbf meshless discretization of diffusion problems and image halftoning, *Comput. Math. Appl.* 75 (12) (2018) 4305–4321. doi:<https://doi.org/10.1016/j.camwa.2018.03.031>.  
URL <http://www.sciencedirect.com/science/article/pii/S0898122118301676>
- [38] R. Zamolo, Radial Basis Function-Finite Difference Meshless Methods for CFD Problems, Ph.D. thesis, Trieste, Italy (2019).
- [39] R. Hardy, Multiquadric equations of topography and other irregular surfaces, *J. Geophys. Res.* 76 (8) (1971) 1905–1915. doi:10.1029/JB076i008p01905.  
URL <http://doi.wiley.com/10.1029/JB076i008p01905>
- [40] R. Franke, A critical comparison of some methods for interpolation of scattered data (Technical Report), Tech. Rep. NPS-53-79-003, Naval Postgraduate School, Monterey, California (1980).  
URL <http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA081688http://hdl.handle.net/10945/35052>
- [41] R. Franke, Scattered data interpolation: tests of some methods, *Math. Comput.* 38 (157) (1982) 181–200. doi:10.1090/S0025-5718-1982-0637296-4.  
URL <http://www.ams.org/jourcgi/jour-getitem?pii=S0025-5718-1982-0637296-4>
- [42] C. Lee, X. Liu, S. Fan, Local multiquadric approximation for solving boundary value problems, *Computat. Mech.* 30 (5-6) (2003) 396–409. doi:10.1007/s00466-003-0416-5.  
URL <http://link.springer.com/10.1007/s00466-003-0416-5>
- [43] S. Sarra, E. Kansa, Multiquadric radial basis function approximation methods for the numerical solution of partial differential equations, *Advances in Computational Mechanics* 2 (2009) 1–206.  
URL <http://www.scottsarra.org/math/papers/mqMonographSarraKansa.pdf>
- [44] G. Fasshauer, Meshfree Approximation Methods with Matlab., Vol. 6 of *Interdisciplinary Mathematical Sciences*, World Scientific, Singapore, 2007, Ch. 9: Conditionally Positive Definite Radial Functions.  
URL <https://www.worldscientific.com/worldscibooks/10.1142/6437>
- [45] R. Horn, C. Johnson, *Matrix Analysis*, Cambridge University Press, 1990.  
URL <https://books.google.it/books?id=PLYQN0ypTwEC>

- [46] T. Chan, B. Smith, Domain decomposition and multigrid algorithms for elliptic problems on unstructured meshes, *Electron T. Numer. Ana.* 2 (1994) 171–182. doi:10.1090/conm/180/1970.  
1105 URL <http://etna.mcs.kent.edu/volumes/1993-2000/vol2>
- [47] W. Hackbusch, *Iterative Solution of Large Sparse Systems of Equations.*, Vol. 95 of Applied Mathematical Sciences, Springer-Verlag, Berlin, Germany, 1994, Ch. 6: Analysis for M-Matrices. doi:10.1007/978-1-4612-4288-8.  
1110 URL <https://www.springer.com/br/book/9781461287247>
- [48] E. Larsson, B. Fornberg, Theoretical and computational aspects of multivariate interpolation with increasingly flat radial basis functions, *Comput. Math. Appl.* 49 (1) (2005) 103–130. doi:<https://doi.org/10.1016/j.camwa.2005.01.010>.  
1115 URL <https://www.sciencedirect.com/science/article/pii/S0898122105000118>
- [49] R. Zamolo, E. Nobile, Numerical analysis of heat conduction problems on 3d general-shaped domains by means of a RBF collocation meshless method, *J. Phys. Conf. Ser.* 923 (2017) 012034. doi:10.1088/1742-6596/923/1/012034.  
1120 URL <https://doi.org/10.1088%2F1742-6596%2F923%2F1%2F012034>
- [50] E. Cuthill, J. McKee, Reducing the bandwidth of sparse symmetric matrices, in: *Proceedings of the 1969 24th National Conference, ACM '69*, ACM, New York, NY, USA, 1969, pp. 157–172. doi:10.1145/800195.805928.  
1125 URL <http://doi.acm.org/10.1145/800195.805928>