Engineering Analysis with Boundary Elements 173 (2025) 106154

Contents lists available at ScienceDirect

Engineering Analysis with Boundary Elements

journal homepage: www.elsevier.com/locate/enganabound

A study on different implementations of Neumann boundary conditions in the meshless RBF-FD method for the phase-field modelling of dendrite growth

Tadej Dobravec [a], Boštjan Mavrič [b,a], Božidar Šarler [a,b,*]

[a] Faculty of Mechanical Engineering, University of Ljubljana, Aškerčeva cesta 6, Ljubljana, SI-1000, Slovenia
[b] Institute of Metals and Technology, Lepi pot 11, Ljubljana, SI-1000, Slovenia

ARTICLE INFO

Keywords:
RBF-FD
Neumann boundary conditions
Ghost nodes
Phase-field method
Dendrite growth

ABSTRACT

This paper studies and assesses different Neumann boundary conditions (BC) implementations in the radial basis function generated finite difference (RBF-FD) method. We analyse four BC implementations by solving a phase-field model for single dendrite growth in supercooled pure melts. In the first BC implementation, the BC are satisfied when constructing interpolation problems in the local support domains near the boundary. In the second one, the BC are satisfied by solving an additional system of linear equations for the field values in the boundary nodes. In the third one, we add a layer of ghost nodes to the boundary nodes; the BC are satisfied by solving an additional system of linear equations for the field values in the ghost nodes. The fourth BC implementation uses the same node distribution as the third one; since we are dealing with the symmetric BC, we set the values in the ghost nodes by direct mirroring. We analyse the influence of the size of a local support domain and the type of node distribution (regular/scattered) on the accuracy. We show that using ghost nodes is recommended to consider Neumann BC in the RBF-FD method accurately when solving phase-field models for dendritic growth.

1. Introduction

The strong-form meshless radial basis function generated finite difference (RBF-FD) method [1,2], also known as the local radial basis function collocation method (LRBFCM) [3], emerged roughly 20 years ago as a powerful tool to solve various problems in science and engineering [4–9]. The RBF-FD method approximates the solution of a partial differential equation (PDE) in the nodes distributed in the computational domain and boundary. A local support domain is constructed for each node, consisting (usually) of the nearest neighbouring nodes. The RBF-FD method uses radial basis function (RBF) collocation in local support domains to obtain finite-difference-like coefficients of an arbitrary linear differential operator. The RBFs in the interpolation are usually augmented with polynomials, which dictate the method's order of convergence. Conversely, the RBFs control the accuracy and possible issues due to locally unusual node distributions.

The RBF-FD method experiences issues when evaluating derivatives at the computational domain boundary where the local support domains are one-sided [2,10], decreasing the method's accuracy when applying the Neumann boundary conditions (BC). Two known solutions exist to mitigate issues with the Neumann BC in the RBF-FD method. The first one is to increase the number of nodes in local support domains [2]; however, a large number of nodes in local support domains increases the method's computational complexity and limits the method's applicability. The second one adds ghost nodes to the boundary nodes, an approach well-known in the mesh-based finite difference method [2]. Additionally, an oversampled RBF-FD method [11] has recently emerged as an up-and-coming candidate to mitigate problems with Neumann BC, especially for complex shapes of computational domain boundary, where the positioning of ghost nodes becomes non-trivial.

In our studies, we apply the strong-form meshless RBF-FD method to solve the phase-field (PF) models describing dendritic solidification [8,12,13]. Dendrites are out-of-equilibrium structures commonly occurring under various casting conditions in different materials [14,15]. Their complex morphology and impact on chemical inhomogeneity are closely linked to the material properties of the solidified material [16]. Needless to emphasise, modelling dendritic solidification is essential for producing high-quality castings for various applications. The PF model represents the method of choice for modelling solidification [17,18] and many other free-boundary problems in material science, like crack

Abbreviations: RBF-FD, radial basis function generated finite difference; LRBFCM, local radial basis function collocation method; RBF, radial basis function; PHS, poly-harmonic spline; PF, phase-field; BC, boundary conditions; PDE, partial differential equation

[*] Corresponding author at: Faculty of Mechanical Engineering, University of Ljubljana, Aškerčeva cesta 6, Ljubljana, SI-1000, Slovenia.

E-mail addresses: tadej.dobravec@fs.uni-lj.si (T. Dobravec), bostjan.mavric@fs.uni-lj.si (B. Mavrič), bozidar.sarler@fs.uni-lj.si (B. Šarler).

https://doi.org/10.1016/j.enganabound.2025.106154
Received 11 September 2024; Received in revised form 24 December 2024; Accepted 4 February 2025
Available online 14 February 2025

propagation in brittle materials [19–21].

This paper considers a simplified case of a single dendrite growing into a supercooled pure melt [22]; the convection and interface kinetics are neglected, and the material properties in the solid and liquid phases are equal and constant. We recast the PF model with the non-linear preconditioned PF to ensure numerical stability when solving the model with larger node spacings [23]. The preconditioned PF model can be implemented in two ways. In the first one, the PDEs are expanded so that the spatial operators apply only to the preconditioned PF and supersaturation [24]. In the second implementation, the spatial operators apply to two additional fields, which depend on the preconditioned PF gradient [25]. The first implementation is computationally more efficient than the second one and is especially suitable for implementations utilising GPUs. The second one is easier to implement and more versatile regarding the choice of anisotropy function; these two features have contributed to the selection of the second implementation in the present study.

The main aim of the present paper is to analyse different implementations of Neumann BC in the RBF-FD method for the solution of non-linear parabolic PDEs. This paper extends our previous study on different BC implementations in the RBF-FD method applied to PF modelling [26]. We are solving the same test case and analysing the same four implementations of the BC as in the previous study. Additionally, the current research considers growth in two different growth directions, initialised from two different positions in the computational domain and using two different types of node distribution. The selected test cases are very suitable for investigating the influence of node distribution, the size of the local support domain, and the type of node distribution on the accuracy and spatial convergence for different BC implementations.

2. Governing equations

We consider the growth of a single dendrite from a pure supercooled melt in the 2-D computational domain Ω. The solidification is diffusion-driven; we do not consider melt convection. A case with constant density, specific heat at constant pressure, and thermal conductivity is tackled. The governing equations used are valid for the cases with negligible interface kinetics. The dimensionless PF model for this case consists of PDEs for the PF ϕ and the dimensionless supercooling θ [22]. The spatial coordinates and time are measured in units of PF interface thickness and PF characteristic attachment time, respectively. The PF equals $\phi = 1$ in the solid and $\phi = -1$ in the liquid phase. We recast the original PF model using the preconditioned PF $\psi = \sqrt{2}\tanh^{-1}(\phi)$ to increase numerical stability for larger node spacings [23].

The governing equation for θ reads as

$$\frac{\partial\theta}{\partial t} = D\nabla^2\theta + \frac{1-\phi^2}{2\sqrt{2}}\frac{\partial\psi}{\partial t}, \tag{1}$$

where D stands for the dimensionless thermal diffusivity. We consider the implementation of the governing equation for ψ where the spatial operators apply to two additional fields $a(\boldsymbol{n})$ and $\boldsymbol{a}(\boldsymbol{n})$, i.e., anisotropy functions [25]. In this paper, we use simple cubic anisotropy functions

$$a(\boldsymbol{n}) = 1 - 3\epsilon_4 + 4\epsilon_4\left(n_x^4 + n_y^4\right), \tag{2}$$

and

$$\boldsymbol{a}(\boldsymbol{n}) = 16\epsilon_4|\nabla\psi|a(\boldsymbol{n})\left(n_x(n_x^4 + n_y^4 - n_x^2), n_y(n_x^4 + n_y^4 - n_y^2)\right), \tag{3}$$

where ϵ_4 stands for the anisotropy strength of the interface energy, and $\boldsymbol{n} = (n_x, n_y) = -\nabla\psi/|\nabla\psi|$ for the normal to the interface. The cubic anisotropy functions from Eqs. (2) and (3) are valid when the four preferential growth directions are aligned with the coordinate system axes. The functions must be recast when considering growth in arbitrary preferential growth directions. Suppose α is an angle between the positive x-axis and the angle-wise nearest arbitrary preferential growth direction. We have to recast anisotropy functions as

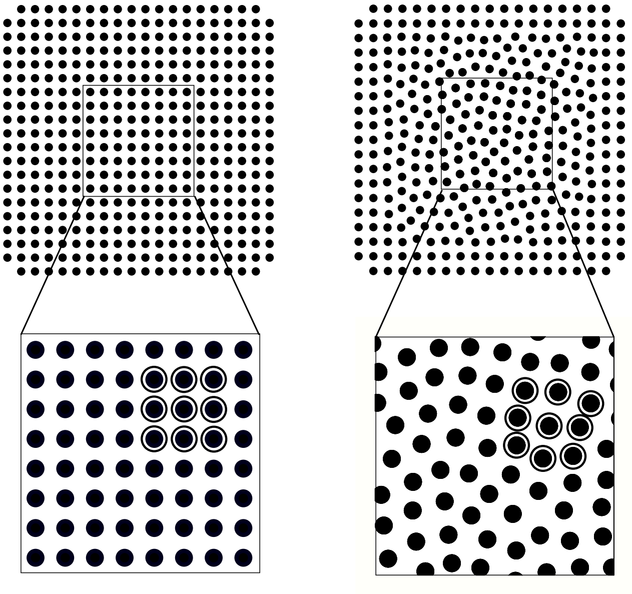

Fig. 1. Regular (left) and scattered (right) node distributions with 396 computational nodes (top). Examples of local support domains with nine computational nodes are also shown by large circles (bottom).

$$a(\boldsymbol{n}) \to a(\boldsymbol{R}^T\boldsymbol{n}), \quad \boldsymbol{a}(\boldsymbol{n}) \to \boldsymbol{R}\boldsymbol{a}(\boldsymbol{R}^T\boldsymbol{n}), \tag{4}$$

where $\boldsymbol{R}$ is the rotation matrix

$$\boldsymbol{R} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}. \tag{5}$$

The governing equation for ψ reads as

$$\begin{aligned} a^2(\boldsymbol{n})\frac{\partial\psi}{\partial t} &= \sqrt{2}\left(\phi - \lambda(1-\phi^2)\theta\right) + 2a(\boldsymbol{n})\nabla a(\boldsymbol{n})\cdot\nabla\psi \\ &\quad - \sqrt{2}\phi\nabla\psi\cdot\boldsymbol{a}(\boldsymbol{n}) + \nabla\cdot\boldsymbol{a}(\boldsymbol{n}) + a^2(\boldsymbol{n})\left(\nabla^2\psi - \sqrt{2}\phi|\nabla\psi|^2\right), \end{aligned} \tag{6}$$

where λ is a free parameter of the PF model. The relation $D = \lambda\alpha_2$ links parameter λ to dimensionless thermal diffusivity D, where $\alpha_2 = 0.6267$ [22].

3. Numerical methods

We employ the RBF-FD method and the forward Euler scheme to discretise the non-linear parabolic PDEs from the previous section. We are solving the PDEs on regular or scattered distributions of computational nodes with the constant node density $\rho = 1/h^2$, where h is the characteristic node spacing. In the case of regular node distribution, h is the exact spacing between a node and its south, east, north, and west neighbour. Fig. 1 shows examples of regular and scattered node distributions. Details of generating a scattered node distribution are given in our previous publication [8].

We set the time step in the forward Euler scheme as

$$\Delta t = \alpha_{\Delta t}\frac{1}{4}\frac{h^2}{\max\left(D, 1/(1-\epsilon_4)\right)}, \tag{7}$$

where $\alpha_{\Delta t}$ stands for the time step stability parameter. The numerical experiments show that a value $\alpha_{\Delta t} = 0.3$ is appropriate for solving the test cases in the present paper [26].

The RBF-FD method is based on interpolation in local support domains using RBFs and monomials. We employ polyharmonic spline (PHS) RBF interpolation; a PHS is defined as

$$\Phi(r) = r^n, \quad n = 1, 3, 5, \dots, \tag{8}$$

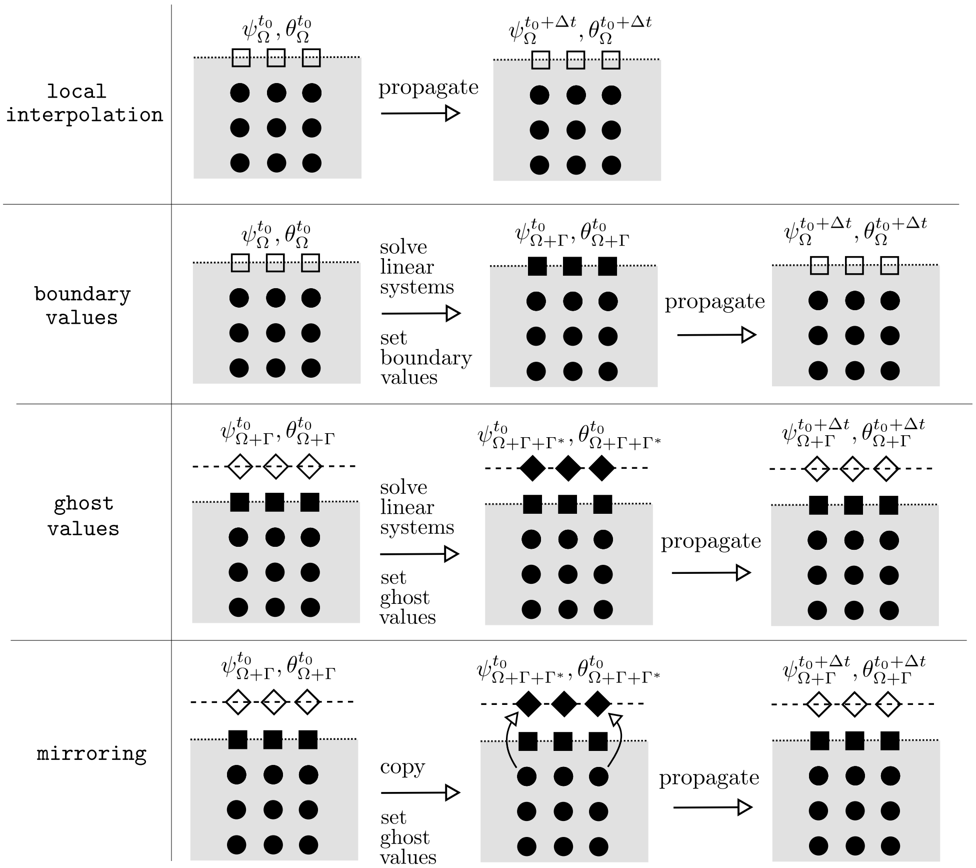
local interpolation
$\psi_\Omega^{t_0}, \theta_\Omega^{t_0}$
propagate
$\psi_\Omega^{t_0+\Delta t}, \theta_\Omega^{t_0+\Delta t}$
boundary values
$\psi_\Omega^{t_0}, \theta_\Omega^{t_0}$
solve linear systems
set boundary values
$\psi_{\Omega+\Gamma}^{t_0}, \theta_{\Omega+\Gamma}^{t_0}$
propagate
$\psi_\Omega^{t_0+\Delta t}, \theta_\Omega^{t_0+\Delta t}$
ghost values
$\psi_{\Omega+\Gamma}^{t_0}, \theta_{\Omega+\Gamma}^{t_0}$
solve linear systems
set ghost values
$\psi_{\Omega+\Gamma+\Gamma^*}^{t_0}, \theta_{\Omega+\Gamma+\Gamma^*}^{t_0}$
propagate
$\psi_{\Omega+\Gamma}^{t_0+\Delta t}, \theta_{\Omega+\Gamma}^{t_0+\Delta t}$
mirroring
$\psi_{\Omega+\Gamma}^{t_0}, \theta_{\Omega+\Gamma}^{t_0}$
copy
set ghost values
$\psi_{\Omega+\Gamma+\Gamma^*}^{t_0}, \theta_{\Omega+\Gamma+\Gamma^*}^{t_0}$
propagate
$\psi_{\Omega+\Gamma}^{t_0+\Delta t}, \theta_{\Omega+\Gamma}^{t_0+\Delta t}$

Fig. 2. Scheme of four BC implementations. The shadowed area represents the computational domain Ω. Boundary Γ and extended boundary Γ^* are represented by dotted and dashed lines. Circles represent inner nodes of Ω. Squares represent boundary nodes of Γ. Diamonds represent ghost nodes of Γ^*. A filled circle, square, or diamond marks a known value in a computational node. An empty circle, square, or diamond marks unknown values in a computational node.

where n is the (odd) PHS degree. The monomials are defined as

$$p_1(\tilde{\boldsymbol{r}}) = 1, \quad p_2(\tilde{\boldsymbol{r}}) = \tilde{x}, \quad p_3(\tilde{\boldsymbol{r}}) = \tilde{y}, \quad p_4(\tilde{\boldsymbol{r}}) = \tilde{x}^2, \quad \dots, \tag{9}$$

where $\tilde{\boldsymbol{r}} = (\tilde{x}, \tilde{y})$. A local support domain consists of a node and $N-1$ nearest neighbours. Fig. 1 shows examples of local support domains for regular and scattered node distributions. The interpolation's approximation of scalar field η with N RBFs and N_{aug} monomials in a local support domain of node ${}_l\boldsymbol{r}$ reads as

$$\eta(\boldsymbol{r}) \approx \sum_{i=1}^{N} {}_l\alpha_i\, \Phi\left(\frac{|\boldsymbol{r} - {}_l\boldsymbol{r}_i|}{{}_l h}\right) + \sum_{i=1}^{N_{aug}} {}_l\alpha_{N+i}\, p_i\left(\frac{\boldsymbol{r} - {}_l\boldsymbol{r}}{{}_l h}\right), \tag{10}$$

where ${}_l\alpha_i$ stands for an interpolation coefficient; ${}_l h$ and ${}_l\boldsymbol{r}_i$ are the characteristic size of and a node from a local support domain. Applying Eq. (10) in N nodes from a local support domain yields an underdetermined system of linear equations

$$\begin{aligned} \eta({}_l\boldsymbol{r}_1) &= \sum_{i=1}^{N} {}_l\alpha_i\, \Phi\left(\frac{|{}_l\boldsymbol{r}_1 - {}_l\boldsymbol{r}_i|}{{}_l h}\right) + \sum_{i=1}^{N_{aug}} {}_l\alpha_{N+i}\, p_i\left(\frac{{}_l\boldsymbol{r}_1 - {}_l\boldsymbol{r}}{{}_l h}\right), \\ \eta({}_l\boldsymbol{r}_2) &= \sum_{i=1}^{N} {}_l\alpha_i\, \Phi\left(\frac{|{}_l\boldsymbol{r}_2 - {}_l\boldsymbol{r}_i|}{{}_l h}\right) + \sum_{i=1}^{N_{aug}} {}_l\alpha_{N+i}\, p_i\left(\frac{{}_l\boldsymbol{r}_2 - {}_l\boldsymbol{r}}{{}_l h}\right), \dots \end{aligned} \tag{11}$$

We augment the underdetermined system with additional N_{aug} conditions

$$\sum_{i=1}^{N} p_1\left(\frac{{}_l\boldsymbol{r}_i - {}_l\boldsymbol{r}}{{}_l h}\right) {}_l\alpha_i = 0, \quad \dots \quad \sum_{i=1}^{N} p_{N_{aug}}\left(\frac{{}_l\boldsymbol{r}_i - {}_l\boldsymbol{r}}{{}_l h}\right) {}_l\alpha_i = 0, \tag{12}$$

to ensure a square system of linear equations. Applying a linear differential operator $\mathcal{D}$ to Eq. (10) applies on RBFs and monomials only; this allows to approximate $\mathcal{D}$ applied on scalar field η in ${}_l\boldsymbol{r}$ as

$$\mathcal{D}\eta({}_l\boldsymbol{r}) \approx \sum_{i=1}^{N} {}_l w_i\, \eta({}_l\boldsymbol{r}_i), \tag{13}$$

where ${}_l w_i$ is a finite-difference-like coefficient of $\mathcal{D}$. A coefficient ${}_l w_i$ depends on the elements of the interpolation matrix inverse and the values of $\mathcal{D}$ applied on RBFs and monomials in ${}_l\boldsymbol{r}$. The details of the RBF-FD method can be found in our former publications [8,12]. In the present study, we use fifth-degree PHS ($n = 5$), augmented by monomials up to the second-order ($N_{aug} = 6$).

3.1. Implementations of BC in the RBF-FD method

Suppose a linear boundary condition $\mathcal{B}$ applies to a scalar field η on the boundary Γ

$$\mathcal{B}(\boldsymbol{r})\eta(\boldsymbol{r}) = b(\boldsymbol{r}), \tag{14}$$

where b represents the right-hand side of a boundary condition. We can employ different approaches to implement such BC in the RBF-FD method. In this paper, we test four implementations of the BC:

- `local interpolation`,
- `boundary values`,
- `ghost values`, and
- `mirroring`.

Fig. 2 shows the schemes of all four BC implementations.

In the implementation `local interpolation`, we apply Eq. (14) to Eq. (11) during the construction of the local interpolation problem if node ${}_l\boldsymbol{r}_j$ lies on Γ; in that way, the finite-difference-like coefficients ${}_l w_i$ from Eq. (13) satisfy BC directly. We propagate the solution of

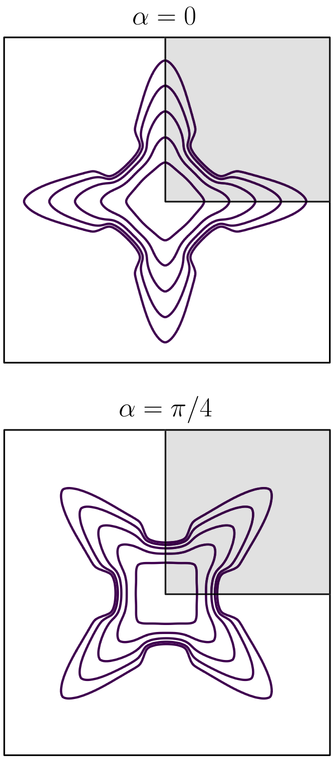

Fig. 3. Evolution of the solid–liquid interface, i.e., $\psi = 0$ contour, for $\alpha = 0$ (top) and $\alpha = \pi/4$ (bottom). The test case is solved in domain $\Omega = [-115.2, 115.2] \times [-115.2, 115.2]$. We also use domain $\Omega = [0, 115.2] \times [0, 115.2]$, which is marked by shadowed square. The solid–liquid interface is plotted with time increment $\delta t = 40$ as the most inner contour is plotted at $t = 40$.

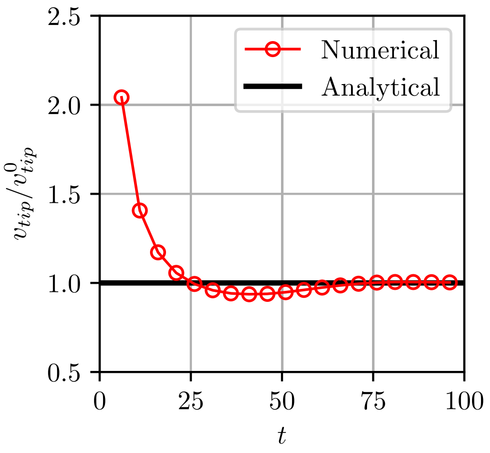

Fig. 4. Dendrite growth speed as a function of time.

a parabolic PDE using the forward Euler scheme in Ω, while the BC are already satisfied when evaluating spatial operators. We use label `local interpolation` for the first implementation since the BC are satisfied by construction of interpolation in local support domains. In the other three BC implementations, the interpolation in local support domains does not consider BC.

In the implementation `boundary values`, values of η in the boundary nodes on Γ are unknown. Before propagating the field values in the inner nodes from Ω, we solve a system of linear equations for the field values in the boundary nodes from Γ. We use label `boundary values` for the second implementation since the BC are satisfied by the appropriate values in boundary nodes. After the BC are satisfied, we propagate η in inner nodes from Ω for one time step using the forward Euler scheme.

In the implementation `ghost values`, we add a ghost node to each boundary node. A ghost node is placed in the direction of outward-facing normal to Γ; the distance between a boundary node and its corresponding ghost node is equal to h. We mark the extended boundary where ghost nodes are placed as Γ^*. Values of η in the ghost nodes are unknown. Before the propagation of the field values in the inner and boundary nodes of $\Omega + \Gamma$, we solve a system of linear equations for field values in the ghost nodes from Γ^*. We use label `ghost values` for the third implementation since the BC are satisfied by the appropriate values in ghost nodes. After the BC are satisfied, we propagate η in inner and boundary nodes from $\Omega + \Gamma$ for one time step using the forward Euler scheme.

The implementation `mirroring` uses the same node distribution with the ghost nodes as implementation `ghost values`. Values of η in the ghost nodes are unknown. Before the propagation of the field values in the inner and boundary nodes from $\Omega + \Gamma$, we set the values in ghost nodes from Γ^* using direct mirroring. We use label `mirroring` for the fourth implementation since the BC are satisfied by direct mirroring to the ghost nodes. This implementation is an alternative one; we can use it since our test case has symmetric BC.

4. Results

4.1. Problem definition

We consider the growth of a single dendrite from a supercooled pure melt in a square computational domain Ω. The initial condition for ψ is a circular nucleus with the centre $\boldsymbol{r}_0$ and radius r_0

$$\psi(\boldsymbol{r}, t = 0) = r_0 - |\boldsymbol{r} - \boldsymbol{r}_0|. \tag{15}$$

The initial condition for θ reads as

$$\theta(\boldsymbol{r}, t = 0) = -\Delta, \tag{16}$$

where Δ stands for the initial constant supercooling. A test case runs from $t = 0$ till $t = t_{end}$.

The symmetric BC are applied for fields ψ, θ, a, and $\boldsymbol{a}$ when solving Eqs. (1) and (6). The symmetric BC for a scalar field is the zero-flux Neumann BC

$$\nabla\psi\Big|_\Gamma \cdot \boldsymbol{n}_\Gamma = 0, \quad \nabla\theta\Big|_\Gamma \cdot \boldsymbol{n}_\Gamma = 0, \quad \nabla a\Big|_\Gamma \cdot \boldsymbol{n}_\Gamma = 0, \tag{17}$$

where $\boldsymbol{n}_\Gamma$ is the outward-facing normal to Γ. The same BC as in Eq. (17) also apply to a component of a vector field parallel to the boundary, while Dirichlet BC apply to the component of a vector perpendicular to the boundary

$$\nabla(\boldsymbol{a}\Big|_\Gamma \cdot \boldsymbol{t}_\Gamma) \cdot \boldsymbol{n}_\Gamma = 0, \quad \boldsymbol{a}\Big|_\Gamma \cdot \boldsymbol{n}_\Gamma = 0, \tag{18}$$

where $\boldsymbol{t}_\Gamma$ is the tangent to Γ.

We solve one of the 2-D test cases from [22] with $\Delta = 0.55$, $\epsilon_4 = 0.05$, and $D = 4$. The test case considers growth into a supercooled infinitely-large computational domain. In the considered test case, the dimensionless analytical steady-state growth speed equals $v_{tip}^0 d_0/D_T = 0.017$, where D_T and d_0 are the dimensional thermal conductivity and capillary length, respectively. The nucleation parameters are $r_0 = 3$ and $\boldsymbol{r}_0 = (0, 0)$. We use two preferential growth directions, defined by angles $\alpha = 0$ and $\alpha = \pi/4$. We solve the test case in two computational domains $\Omega = [0, 115.2] \times [0, 115.2]$ and $\Omega = [-115.2, 115.2] \times [-115.2, 115.2]$ till $t_{end} = 100$. The selected simulation parameters effectively mimic growth into an infinitely-large supercooled melt. Consequently, the results should be α-independent. Figs. 3 and 4 show the evolution of the solid–liquid interface for two preferential growth directions and growth speed as a function of time. Fig. 4 indicates a transient in the growth speed initially, followed by the steady-state growth after $t > 75$. The speed is rescaled by the analytical steady-state growth speed v_{tip}^0.

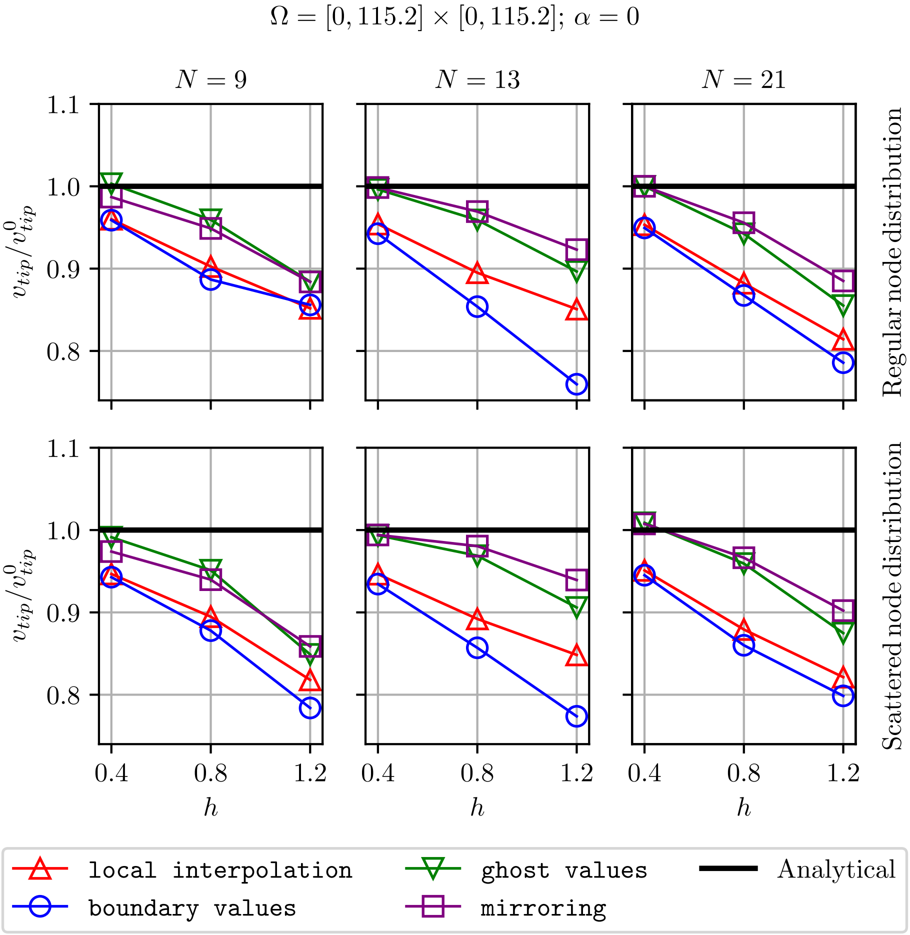

Fig. 5. Steady-state growth speed for different numerical parameters (h and N), BC implementations (`local interpolation`, `boundary values`, `ghost values`, and `mirroring`), and node distributions (regular and scattered) for simulation in domain $\Omega = [0, 115.2] \times [0, 115.2]$ with $\alpha = 0$.

4.2. Numerical experiments

The numerical methods are implemented in the programming language Fortran 2018 and compiled with Intel Fortran Compiler Classic 2021. We use compilers and libraries from Intel oneAPI Base and HPC Toolkits. We run the numerical experiments on an HP ZBook laptop with the hexa-core Intel Core i7-9750H 2.6–4.5 GHz processor. In the numerical experiments, we analyse the spatial convergence of the RBF-FD method for

- four BC implementations (`local interpolation`, `boundary values`, `ghost values`, and `mirroring`),
- two types of node distributions (regular and scattered), and
- three numbers of nodes in local support domains ($N = 9$, $N = 13$, and $N = 21$),

utilising

- two computational domains ($\Omega = [0, 115.2] \times [0, 115.2]$ and $\Omega = [-115.2, 115.2] \times [-115.2, 115.2]$) and
- two preferential growth direction angles ($\alpha = 0$ and $\alpha = \pi/4$).

In $\Omega = [-115.2, 115.2] \times [-115.2, 115.2]$, the dendrite grows in the middle of the computational domain as seen in Fig. 3; in this case, the influence of BC on the growth should be negligible. The reason for running numerical experiments in this computational domain is to obtain reference BC-independent results. In $\Omega = [0, 115.2] \times [0, 115.2]$, dendrite grows from the bottom-left corner of the computational domain, as seen in Fig. 3. In this case, the results should be strongly BC-dependent, especially for $\alpha = 0$. The latter case is ideal for analysing different implementations of Neumann BC since the detail in the solution, which is numerically hardest to resolve, lies directly at the computational domain's boundary.

4.2.1. Growth in $\Omega = [0, 115.2] \times [0, 115.2]$ - observations

Figs. 5 and 6 show steady-state growth speed for $\alpha = 0$ and $\alpha = \pi/4$ when using computational domain $\Omega = [0, 115.2] \times [0, 115.2]$. We observe the spatial convergence towards the analytical steady-state speed in both figures. The results for $\alpha = 0$, obtained using regular node distribution, are almost identical to the ones obtained using scattered node distribution; we see a large difference for $N = 9$ and $h = 1.2$ only, where the scattered node distribution yields a higher error. Implementation `boundary values` yields the highest error; one can also observe that N has almost negligible impact on the results in this implementation. Implementation `local interpolation` experiences lower error than implementation `boundary values`; the largest difference between the implementations is observed for $N = 13$. We observe the lowest error for implementations `ghost values` and `mirroring`, which yield almost identical results. For implementations `local interpolation`, `ghost values`, and `mirroring` the error decreases when increasing N from $N = 9$ to $N = 13$; when increasing N from $N = 13$ to $N = 21$, the error increases.

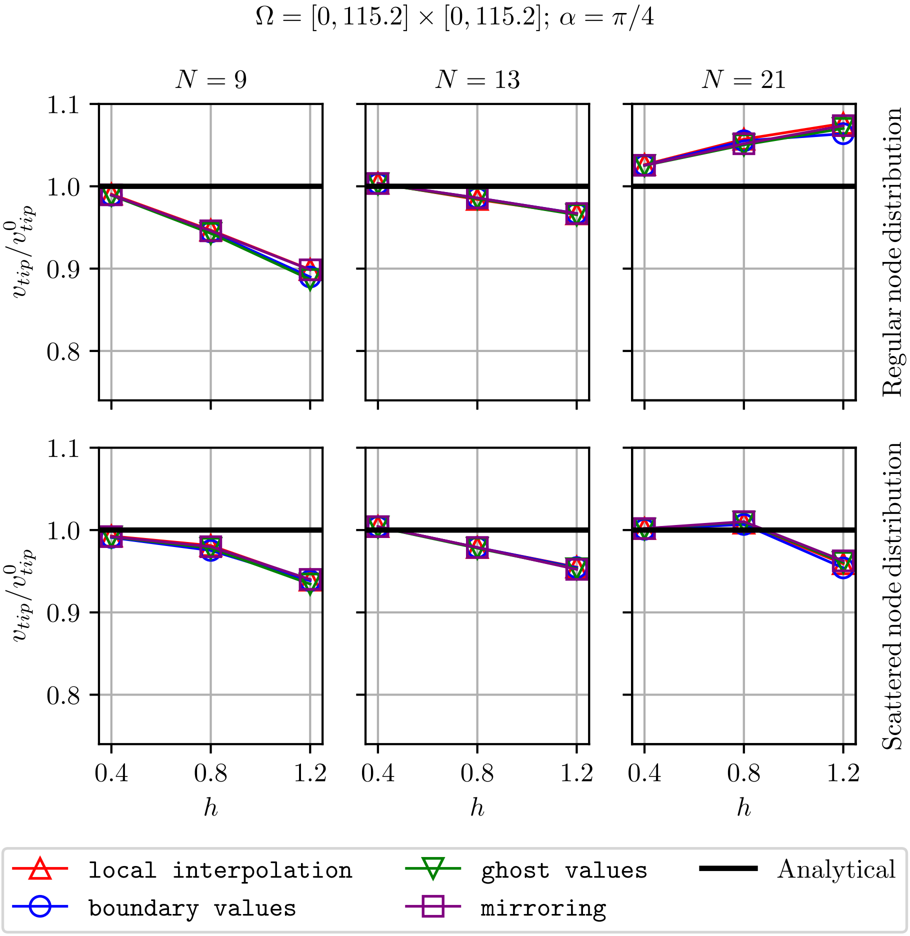

Fig. 6. Steady-state growth speed for different numerical parameters (h and N), BC implementations (`local interpolation`, `boundary values`, `ghost values`, and `mirroring`), and node distributions (regular and scattered) for simulation in domain $\Omega = [0, 115.2] \times [0, 115.2]$ with $\alpha = \pi/4$.

As seen in Fig. 6, all the BC implementations experience the same accuracy when $\alpha = \pi/4$. In the case of regular node distribution, $N = 13$ yields the lowest error, while the other two values of N yield similar errors. In the case of scattered node distribution, the accuracy slightly increases with N. Utilising scattered node distribution yields lower error for $N = 9$ and $N = 21$ compared to regular node distribution and slightly higher error for $N = 13$. The results for $\alpha = 0$ from Fig. 5 are quite different from those for $\alpha = \pi/4$ from Fig. 6. We observe higher accuracy for $\alpha = \pi/4$, where the dendrite tip does not move along the boundary of the computational domain; consequently, different BC implementations for $\alpha = \pi/4$ have negligible impact on the accuracy compared to $\alpha = 0$.

4.2.2. *Growth in $\Omega = [-115.2, 115.2] \times [-115.2, 115.2]$ - observations*

Figs. 7 and 8 show steady-state growth speed for $\alpha = 0$ and $\alpha = \pi/4$ when using computational domain $\Omega = [-115.2, 115.2] \times [-115.2, 115.2]$. We observe the spatial convergence towards the analytical steady-state speed in both figures. The boundary should not significantly impact the results since the dendrite grows from the middle of the computational domain, as seen in Fig. 3. Figs. 7 and 8 clearly show that different implementations of BC do not affect the growth speed. When using the regular node distribution, we observe the lowest error for $N = 13$; value $N = 21$ yields the highest error for $\alpha = 0$, while values $N = 9$ and $N = 21$ yield similar accuracies for $\alpha = \pi/4$. In the case of scattered node distribution, the error decreases with N for $\alpha = 0$ and $\alpha = \pi/4$. When comparing regular and scattered node distributions, the errors are very similar for $N = 9$ and $N = 13$; using scattered node distribution yields much higher accuracy for $N = 21$, especially for $\alpha = 0$. The steady-state growth speed is very sensitive to parameter α when utilising regular node distribution. We observe the sensitivity to change in α when utilising scattered node distribution too; however, to a much lower degree compared to using regular node distribution.

4.2.3. *Discussion*

Results for $\alpha = 0$ using $\Omega = [0, 115.2] \times [0, 115.2]$ (Fig. 5), where the dendrite tip is placed on the boundary of the computational domain directly, are the most important to analyse different implementations of BC. The regular and scattered node distributions are identical at the boundary of the computational domain, as seen in Fig. 1; this explains the resemblance between the results on regular and scattered node distributions. We see that the implementations utilising ghost nodes (`ghost values` and `mirroring`) outperform those without ghost nodes (`local interpolation` and `boundary values`). In the implementations without ghost nodes, the RBF-FD method experiences reduced accuracy since the derivatives have to be evaluated at the computational domain boundary, where the local support domains are one-sided. In the implementations with ghost nodes, the local support domains at the boundary are not one-sided since we add one layer of ghost nodes outside of the computational domain, as seen in Fig. 2. Adding ghost nodes ensures a much more accurate

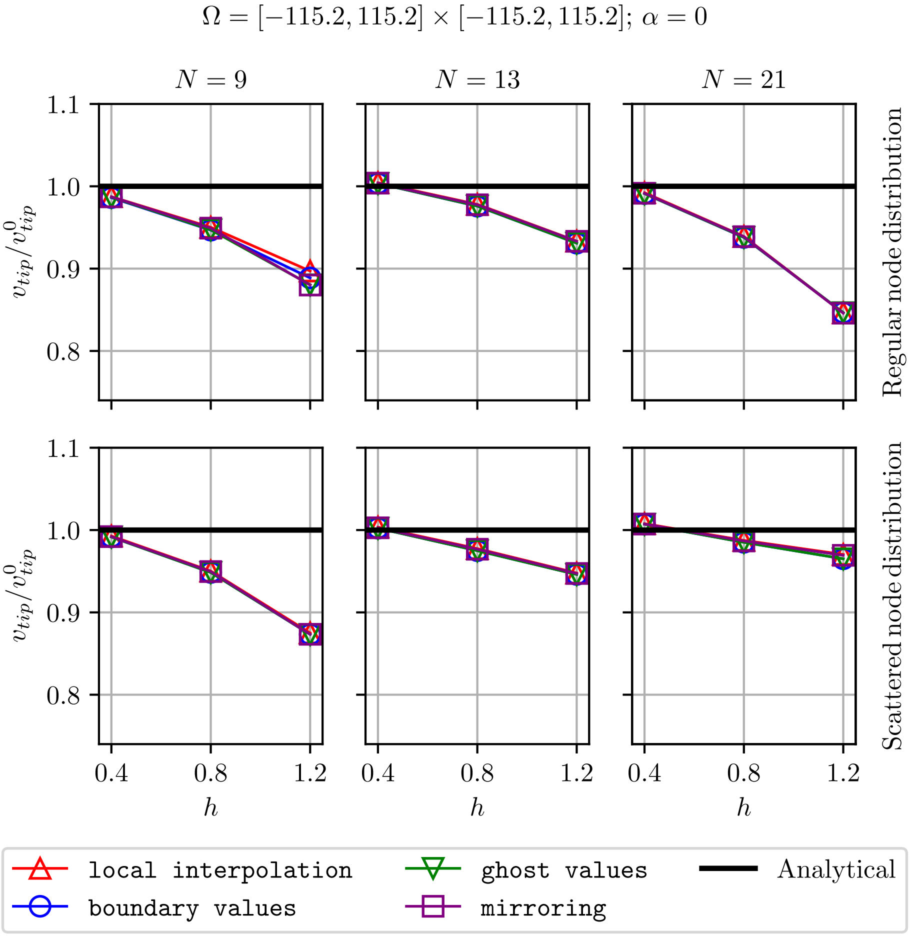

Fig. 7. Steady-state growth speed for different numerical parameters (h and N), BC implementations (`local interpolation`, `boundary values`, `ghost values`, and `mirroring`), and node distributions (regular and scattered) for simulation in domain $\Omega = [-115.2, 115.2] \times [-115.2, 115.2]$ with $\alpha = 0$.

evaluation of derivatives at the boundary and, consequently, increases the overall accuracy of the method. We see that the results for $\alpha = 0$ obtained in $\Omega = [0, 115.2] \times [0, 115.2]$ (Fig. 5) by utilising ghost nodes are almost identical to the results obtained using regular node distribution in $\Omega = [-115.2, 115.2] \times [115.2, 115.2]$ (Fig. 7); this is an important observation, which proves, that the implementations of BC, which utilise ghost nodes, do not introduce any large additional error to the solution. Interestingly, the error for $N = 21$ obtained using $\Omega = [0, 115.2] \times [0, 115.2]$ is lower compared to the error obtained using $\Omega = [-115.2, 115.2] \times [115.2, 115.2]$. The results clearly suggest, that the configuration with $N = 21$ using regular node distribution is unsuitable for solving such problems, especially at larger values of h.

Results for $\alpha = \pi/4$ obtained using $\Omega = [0, 115.2] \times [0, 115.2]$ (Fig. 6) are almost identical to the ones obtained using $\Omega = [-115.2, 115.2] \times [-115.2, 115.2]$ (Fig. 8). In both computational domains, the BC have a negligible impact on growth since the dendrite grows in the direction of diagonals of square computational domains. Similarly, the BC have a negligible effect on the growth for $\alpha = 0$ using $\Omega = [-115.2, 115.2] \times [-115.2, 115.2]$ (Fig. 7); in this case, all implementations of BC yield the same results.

The value $N = 13$ yields the highest accuracy when using regular node distribution or when the node distribution is regular in the region where the dendrite tip grows. The regularity in the node distribution breaks the convergence when increasing N due to the discretisation-induced anisotropy. On the other hand, we do not observe this break when utilising scattered node distribution. The results should be α-independent under the selected simulation parameters. Regularity in the node distribution yields the numerical preferential growth directions and introduces α-dependent results. Utilising scattered node distribution reduces this phenomenon. We have already discussed discretisation-induced anisotropy in the PF modelling of dendrite growth in our previous studies [8,12,13].

5. Conclusions

In the paper, we analyse different implementations of Neumann BC in the RBF-FD method to solve a PF model of dendritic solidification. The considered PF model consists of two coupled non-linear parabolic PDEs. Our solution procedure does not require solving systems of equations in the whole computational domain to propagate the solution in time since we employ an explicit time marching scheme. The performed numerical experiments consider the growth of a cubic dendrite from a supercooled melt. We utilise two square computational domains of different sizes and employ symmetric BC. We analyse how the number of nodes in local support domains and the type of node distribution affect the accuracy of each BC implementation. In the smaller domain, the nucleus is put in the corner of the computational domain; in this setup, the results are highly BC-dependent if the preferential growth directions are parallel to the computational domain boundary. On the

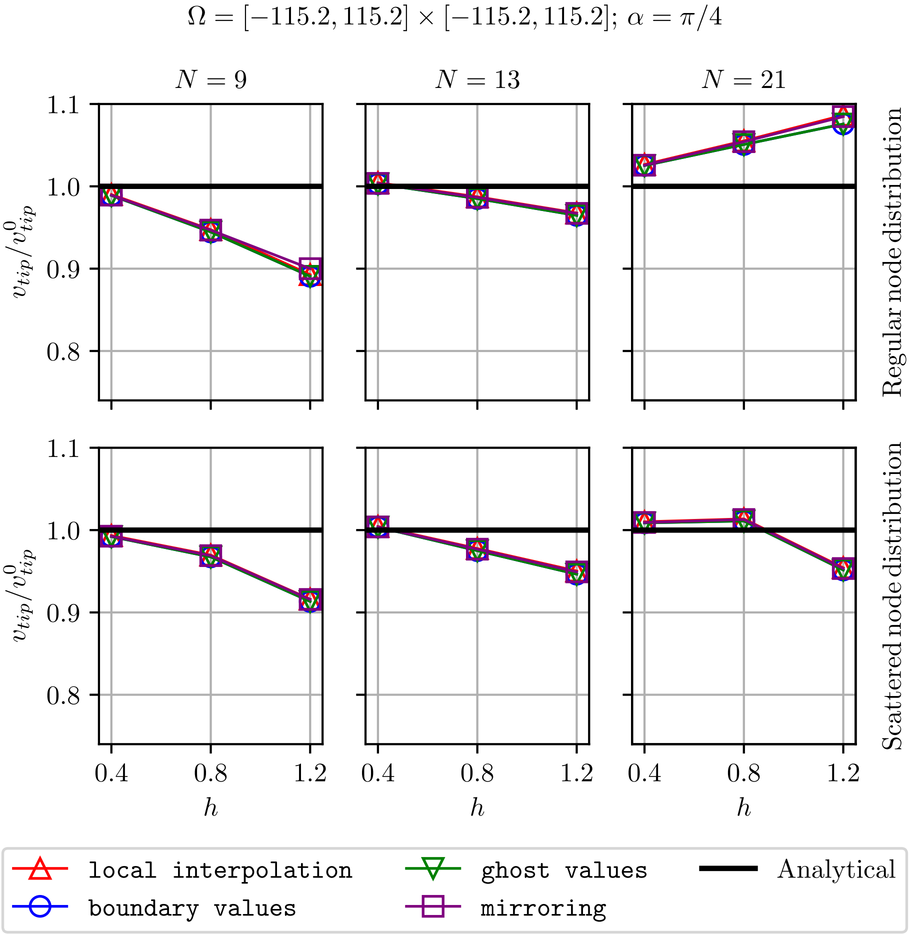

Fig. 8. Steady-state growth speed for different numerical parameters (h and N), BC implementations (`local interpolation`, `boundary values`, `ghost values`, and `mirroring`), and node distributions (regular and scattered) for simulation in domain $\Omega = [-115.2, 115.2] \times [-115.2, 115.2]$ with $\alpha = \pi/4$.

contrary, the results are BC-independent if the preferential growth directions are parallel to the diagonals of the square computational domain. Results obtained in the smaller domain provide important insights for simulating very large cases of single dendrite growth. In such cases, considering symmetry becomes crucial to reduce computational times. The second computational domain is twice as large; in this domain, the nucleus is put in the middle, making the results BC-independent for both preferential growth directions. This scenario resembles cases in which we analyse dynamics of multiple differently-oriented dendrites. In these cases, the activity at the boundary of the computational domain is not of primary focus.

We observe spatial convergence towards the analytical solution for all tested configurations. As expected, all BC implementations yield the same accuracy when the results are BC-independent. In the BC-independent cases, the accuracy increases with the number of nodes in local support domains when using scattered node distribution. On the contrary, the accuracy can even decrease for a larger number of nodes in local support domains when using regular node distribution. The results are much more sensitive to the change of preferential growth directions when regular node distribution is used. In other words, regular node distribution is much more prone to discretisation-induced anisotropy in the PF modelling of dendritic growth than scattered node distribution, as we have observed in our previous similar studies [12].

In the present numerical model, the scattered node distribution is regular at the boundary of the computational domain; therefore, the advantages of using this distribution in BC-independent cases are not translated to BC-dependent cases. One of the tasks of our future research is to investigate the RBF-FD method's performance when the node distribution is also scattered at the boundary. Using ghost nodes in the RBF-FD method essentially increases accuracy when the results are BC-dependent.

In the paper, we utilise square computational domains where the ghost nodes' positioning is trivial. Our approaches utilising ghost nodes are limited to moderately convex or concave boundaries, e.g., a circle. However, positioning ghost nodes in a computational domain with complex convex or concave boundaries can become non-trivial or even impossible. Researchers have recently discovered that using the RBF-FD method in an oversampled setup effectively reduces inaccuracies during evaluating derivatives at the computational domain boundary, even without ghost nodes utilisation [11]. Continuing the present study, we intend to implement the oversampled RBF-FD method and analyse its performance when solving PF models for dendritic growth.

CRediT authorship contribution statement

Tadej Dobravec: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Boštjan Mavrič:** Writing – review & editing, Software, Resources, Methodology, Conceptualization. **Božidar Šarler:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the Slovenian Research and Innovation Agency (ARIS) in the framework of post-doctoral basic projects Z2-4479 and Z2-2640, programme group P2-0162, and project J2-4477.

Data availability

Data will be made available on request.

References

[1] Flyer N, Fornberg B, Bayona V, Barnett GA. On the role of polynomials in RBF-FD approximations: I. Interpolation and accuracy. J Comput Phys 2016;321:21–38. http://dx.doi.org/10.1016/j.jcp.2016.05.026.

[2] Bayona V, Flyer N, Fornberg B, Barnett GA. On the role of polynomials in RBF-FD approximations: II. Numerical solution of elliptic PDEs. J Comput Phys 2017;332:257–73. http://dx.doi.org/10.1016/j.jcp.2016.12.008.

[3] Šarler B, Vertnik R. Meshfree explicit local radial basis function collocation method for diffusion problems. Comput Math Appl 2006;51(8):1269–82. http://dx.doi.org/10.1016/j.camwa.2006.04.013.

[4] Kosec G, Šarler B. H-adaptive local radial basis function collocation meshless method. Comput Mater Contin 2011;26(3):227–53.

[5] Vertnik R, Šarler B. Solution of a continuous casting of steel benchmark test by a meshless method. Eng Anal Bound Elem 2014;45:45–61. http://dx.doi.org/10.1016/j.enganabound.2014.01.017.

[6] Mavrič B, Šarler B. Local radial basis function collocation method for linear thermoelasticity in two dimensions. Int J Numer Method H 2015;25(6):1488–510. http://dx.doi.org/10.1108/HFF-11-2014-0359.

[7] Hanoglu U, Šarler B. Multi-pass hot-rolling simulation using a meshless method. Comput Struct 2018;194:1–14. http://dx.doi.org/10.1016/j.compstruc.2017.08.012.

[8] Dobravec T, Mavrič B, Šarler B. Acceleration of RBF-FD meshless phase-field modelling of dendritic solidification by space-time adaptive approach. Comput Math Appl 2022;126:77–99. http://dx.doi.org/10.1016/j.camwa.2022.09.008.

[9] Vuga G, Mavrič B, Šarler B. An improved local radial basis function method for solving small-strain elasto-plasticity. Comput Methods Appl Mech Engrg 2024;418:116501. http://dx.doi.org/10.1016/j.cma.2023.116501.

[10] Bayona V, Flyer N, Fornberg B. On the role of polynomials in RBF-FD approximations: III. Behavior near domain boundaries. J Comput Phys 2019;380:378–99. http://dx.doi.org/10.1016/j.jcp.2018.12.013.

[11] Larsson E, Mavrič B, Michael A, Pooladi F, Tominec I. Meshfree RBF–FD methods for numerical simulation of PDE problems. J Phys: Conf Ser 2024;2766(1):012158. http://dx.doi.org/10.1088/1742-6596/2766/1/012158, Publisher: IOP Publishing.

[12] Dobravec T, Mavrič B, Šarler B. Reduction of discretisation-induced anisotropy in the phase-field modelling of dendritic growth by meshless approach. Comput Mater Sci 2020;172:109166. http://dx.doi.org/10.1016/j.commatsci.2019.109166.

[13] Dobravec T, Mavrič B, Zahoor R, Šarler B. A coupled domain–boundary type meshless method for phase-field modelling of dendritic solidification with the fluid flow. Internat J Numer Methods Heat Fluid Flow 2023;33(8):2963–81. http://dx.doi.org/10.1108/HFF-03-2023-0131, Publisher: Emerald Publishing Limited.

[14] Kurz W, Fisher DJ, Trivedi R. Progress in modelling solidification microstructures in metals and alloys: dendrites and cells from 1700 to 2000. Int Mater Rev 2019;64(6):311–54. http://dx.doi.org/10.1080/09506608.2018.1537090.

[15] Kurz W, Rappaz M, Trivedi R. Progress in modelling solidification microstructures in metals and alloys. Part II: dendrites from 2001 to 2018. Int Mater Rev 2021;66(1):30–76. http://dx.doi.org/10.1080/09506608.2020.1757894.

[16] Dantzig J, Rappaz M. Solidification. 2nd edition.. Lausanne, Switzerland: EPFL Press; 2017.

[17] Chen L-Q. Phase-field models for microstructure evolution. Annu Rev Mater Res 2002;32(1):113–40. http://dx.doi.org/10.1146/annurev.matsci.32.112001.132041.

[18] Boettinger WJ, Warren JA, Beckermann C, Karma A. Phase-field simulation of solidification. Annu Rev Mater Res 2002;32(1):163–94. http://dx.doi.org/10.1146/annurev.matsci.32.101901.155803 .

[19] Lu X, Li C, Tie Y, Hou Y, Zhang C. Crack propagation simulation in brittle elastic materials by a phase field method. Theor Appl Mech Lett 2019;9(6):339–52. http://dx.doi.org/10.1016/j.taml.2019.06.001.

[20] Borden MJ, Hughes TJR, Landis CM, Verhoosel CV. A higher-order phase-field model for brittle fracture: Formulation and analysis within the isogeometric analysis framework. Comput Methods. Appl Mech Eng 2014;273:100–18. http://dx.doi.org/10.1016/j.cma.2014.01.016.

[21] Ali I, Vuga G, Mavrič B, Hanoglu U, Šarler B. Fourth-order phase-field simulation of cracks using strong form meshless method. J Phys: Conf Ser 2024;2766(1):012167. http://dx.doi.org/10.1088/1742-6596/2766/1/012167, Publisher: IOP Publishing.

[22] Karma A, Rappel W-J. Quantitative phase-field modeling of dendritic growth in two and three dimensions. Phys Rev E 1998;57(4):4323.

[23] Glasner K. Nonlinear preconditioning for diffuse interfaces. J Comput Phys 2001;174(2):695–711. http://dx.doi.org/10.1006/jcph.2001.6933.

[24] Tourret D, Karma A. Growth competition of columnar dendritic grains: A phase-field study. Acta Mater 2015;82:64–83. http://dx.doi.org/10.1016/j.actamat.2014.08.049.

[25] Boukellal AK, Rouby M, Debierre J-M. Tip dynamics for equiaxed Al-Cu dendrites in thin samples : Phase-field study of thermodynamic effects. Comput Mater Sci 2021;186:110051. http://dx.doi.org/10.1016/j.commatsci.2020.110051.

[26] Dobravec T, Mavrič B, Šarler B. On different implementations of boundary conditions in the meshless RBF-FD method for phase-field modelling of dendritic solidification. J Phys: Conf Ser 2024;2766(1):012162. http://dx.doi.org/10.1088/1742-6596/2766/1/012162, Publisher: IOP Publishing.