Mathematics and Computers in Simulation 218 (2024) 49–78

Contents lists available at ScienceDirect

Mathematics and Computers in Simulation

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

Original articles

Stencil and kernel optimisation for mesh-free very high-order generalised finite difference method

S. Clain [a,*], J. Figueiredo [b,c]

[a] Center of mathematics of the FCTUC, Largo D. Dinis, 3000-143 Coimbra, Portugal
[b] Centre of Physics UM - UP, Campus de Gualtar, 4710-057 Braga, Portugal
[c] Department of Mathematics, Campus de Azurém, University of Minho, 4080-058 Guimarães, Portugal

ARTICLE INFO

Keywords:
General finite difference
Mesh-free
Stencil optimisation
Very high-order

ABSTRACT

Generalised Finite Difference Methods and similar mesh-free methods (Pointset method, Multipoint method) are based on three main ingredients: a stencil around the reference node, a polynomial reconstruction and a weighted functional to provide the relations between the derivatives at the reference node and the nodes of the stencil. Very few studies were dedicated to the optimal choice of the stencil together with the other parameters that could reduce the global conditioning of the system and bring more stability and better accuracy. We propose a detailed construction of the very high-order polynomial representation and define a functional that assesses the quality of the reconstruction. We propose and implement several techniques of optimisation and demonstrate the advantages in terms of accuracy and stability.

1. Introduction

The Generalised Finite Difference Method (GFDM) dates back to the 70s' with the articles of Jensen [14], Perrone and Kao [24], Liszka and Orkisz [18] on irregular grids. Such a technique has attracted a lot of researchers and schemes' designers for several reasons: (1) the simplicity to handle boundary conditions while providing very high-order approximations; (2) the ability to avoid the complex meshing machinery to achieve refinement; (3) the ability to handle the Lagrangian/ALE formulation by moving the nodes. There exists a recent literature on the GFDM method for problem using an Eulerian formulation to take advantage of the ability to handle complex geometries [29,32,33].

Under the umbrella of GFDM, a plethora of mesh-free methods has been proposed and implemented.

- The Generalised Finite Difference Method proposed by Liszka & Orkisz [18] and Benito, Gavete, Ureña et al. [1,2,6,30,31], is based on the Taylor expansion to link the partial derivatives with the function at the neighbouring nodes. The least squares minimisation then provides an explicit relation for each derivative with the neighbour values. Such relations are then plug into the physical equations (inner equations or boundary equations) and one obtains a (non)-linear system that only involves the unknown values at the nodes.
- The Finite Pointset Method (FPM) proposed by Kuhnert et al. [15,16,28], also uses the Taylor expansion terms, *i.e.* the partial derivatives, at a given node we connect to the function values at the neighbouring nodes. The principal difference with the GFDM is the inclusion of the physical equations (inner equations and boundary equations) in the weighted least squares procedure. Consequently, the computed partial derivatives do not exactly satisfy the physical equations. Such a technique has also received important developments by Reséndiz-Flores et al. [25–27].

* Corresponding author.
E-mail address: clain@mat.uc.pt (S. Clain).

https://doi.org/10.1016/j.matcom.2023.11.009
Received 28 May 2023; Received in revised form 15 September 2023; Accepted 7 November 2023
Available online 22 November 2023

- The Finite Point Method of Oñate, Idelsohn et al. [3,4,20–23], derives from a general weighted residual method using a local polynomial representation of the approximation over subsets $\Omega_k \subset \Omega$. Polynomial coefficients are explicitly determined regarding the unknowns located on Ω_k. Notice that the local representation is the same on the whole domain Ω_k, contrarily to the GFDM where each representation is related to a node. For that reason, the authors use a Moving Weighted Least Squares method to create a dependence between the polynomial coefficients and the nodes of Ω_k. The reconstruction is plugged into the weighted residual form of the physical equations that provides the (non)-linear system one has to solve regarding the unknowns at the nodes.
- The Multipoint method developed by Jaworska and Orkisz et al. [10–13] is quite different from the previous methodologies, since the authors are dealing with implicit relations between the values and the derivatives. Given a reference node i and a stencil $\mathcal{V}_i'$, the derivative at node i is given by a linear combination of the neighbouring node values together with the derivatives (at the same order) over nodes of $\mathcal{V}_i'$. Such an approach is close to the compact scheme philosophy, since it combines both the values and the derivatives. As in the Pointset method, the numerical approximations are plugged into the physical equation (inner/core and boundary equation).

All the methods are based on three main ingredients: a stencil around the reference node i, a polynomial reconstruction and a weighted functional to provide the relations between the derivatives at the node i and the nodes of the stencil. Then arises the question of optimisation of the three ingredients: the choice of the nodes to elaborate the stencil, the scaling parameter h_M of the polynomial representation and the scaling parameter h_W of the kernel that produces the weights. At the end of the day, one has to solve a linear system of type

$$MWM^{\mathrm{t}}c = MW\Phi$$

where c represents the coefficients' vector of the local polynomial reconstruction while Φ gathers the values at the neighbouring nodes. The matrix $M = M(h_M)$ gathers the polynomial expressions, while $W = W(h_W)$ is the weights' diagonal matrix. Condition number of MWM^{t} is then a key issue to control the accuracy and stability of the method. There exists several recipes in the literature to improve the computation of the polynomial. Jensen [14] selects the closest nodes from the reference node. Perrone and Kao [24] propose the eight segments criterion, consisting on the selection of a node for each octant in a system of Cartesian axes around the reference node. Liszka and Orkisz [18] introduce the four quadrants technique, which consists in picking the two nearest nodes per quadrant. We also mention the reviews on high-order finite difference method on irregular domains [8] or approximation of non-smooth solutions [9].

We propose an optimisation procedure to achieve the best stencil and coefficients h_M and h_W that minimise the condition number of matrix MWM^{t}. On the one hand, the choice of the stencil is related to a discrete minimisation problem where the nodes are the parameters to optimise. On the other hand, the continuous minimisation regarding parameters h_M and h_W is achieved by different derivative-free methods. Numerical tests with and without optimisation for different geometrical configurations are carried out to assess the condition number reduction and the accuracy. We restrict the study to the steady state case since we focus on the operator reconstructions in space together with the boundary conditions prescribed on non-polygonal physical domain. Moving boundary domain with Lagrangian formulation is out of the scope of the present paper.

The rest of the paper is organised as the following. We first recall some basic facts in Section 2 on the polynomial reconstruction and define the material we shall use all along the paper in Section 3. Section 4 concerns the optimisation procedures, both for the stencil (discrete minimisation) and the matrix parameters (continuous minimisation). We present an extensive series of benchmarks to assess the advantage of the minimisation procedures and prove the interest in strongly reducing the condition number by some magnitude. We present some examples of simulations with the convection diffusion reaction equation by using the derivatives computed with the optimal reconstruction.

2. Notations

The $\mathbb{R}^2$ space is described by the generic point $x = (x_1, x_2)$ and $\phi(x) = \phi(x_1, x_2)$ stands for a real-valued smooth function. We introduce the multi-index notation $\beta = (\beta_1, \beta_2) \in \mathbb{N}^2$ with $|\beta| = \beta_1 + \beta_2$ and set,

$$\partial^\beta \phi(x) = \partial_{x_1}^{\beta_1} \partial_{x_2}^{\beta_2} \phi(x_1, x_2).$$

Let $\mathcal{A}$ be a finite subset of $\mathbb{N}^2$. We define the polynomial of degrees in $\mathcal{A}$, centred at a point $\bar{x}$, by

$$\pi(x; h_M, \bar{x}, c, \mathcal{A}) = \sum_{\alpha \in \mathcal{A}} c_\alpha \left(\frac{x - \bar{x}}{h_M} \right)^\alpha,$$

where c_α are real number coefficients and h_M is a scaling parameter. We extend the factorial for multi-index values setting $\alpha! = (\alpha_1)!\,(\alpha_2)!$ and $\alpha! = 0$ if $\alpha_1 < 0$ or $\alpha_2 < 0$. Thanks to this definition, we deduce the relation

$$\partial^\beta \left(\frac{x - \bar{x}}{h_M} \right)^\alpha = \frac{(\alpha - \beta)!}{h_M^{|\beta|}} \left(\frac{x - \bar{x}}{h_M} \right)^{\alpha - \beta}$$

In particular, one has the property

$$\partial^\beta \left(\frac{x - \bar{x}}{h_M} \right)^\alpha_{x = \bar{x}} = \frac{\beta!}{h_M^{|\beta|}} \delta^{\alpha,\beta}$$

where $\delta^{\alpha,\beta} = 1$ if $\alpha = \beta$, and null otherwise.

Let Ω be an open bounded domain of $\mathbb{R}^2$ with boundary $\partial\Omega$. We denote by $\mathcal{G} = (x^i)_{i=1}^I$, with $x^i = (x^i_1, x^i_2)$, a cloud of I points of $\overline{\Omega}$. $\mathring{\mathcal{G}}$ corresponds to the interior nodes of Ω while $\partial\mathcal{G}$ stands for the nodes on the boundary. For any node i, $\mathcal{V}^i \subset \mathcal{G}$ is a stencil of points, usually chosen in the neighbourhood of the node i. Note that we impose $i \in \mathcal{V}^i$.

We further denote by $\boldsymbol{\Phi} = (\phi^i)_{i\in\mathcal{G}}$ the vector of the unknowns such that $\phi^i \approx \phi(x^i)$, while we introduce the stencil-vector $\boldsymbol{\Phi}_{\mathcal{V}^i} = (\phi^j)_{j\in\mathcal{V}^i}$ associated to the stencil $\mathcal{V}^i$. Similarly, for any $\beta \in \mathcal{A}$, we denote $\boldsymbol{\Phi}^{i,\beta} \approx \partial^\beta\phi(x^i)$ the approximations of the derivatives at node i.

3. Derivatives' discretisation

The goal is to provide explicit relations for the β derivative of ϕ, at node i, regarding the data over the stencil $\mathcal{V}^i$, namely, we seek coefficients $a^{i,\beta}_j \in \mathbb{R}$ such that

$$\partial^\beta\phi(x^i) \approx \sum_{j\in\mathcal{V}^i} a^{i,\beta}_j \phi^j.$$

The structural relations provide the discrete relations where $a^{i,\beta}_j$ only depends on the structure of the cloud. To determine the coefficients, we present two distinct methodologies based on the least square method.

3.1. The polynomial approximation

Let i be a node and $\mathcal{V}^i$ its stencil. We seek a polynomial $\pi(x; h_M, x^i, c, \mathcal{A})$ that minimises the loss function

$$J^i(c) := \frac{1}{2}\sum_{j\in\mathcal{V}^i} \omega^2_{ij}\Big(\pi(x^j; h_M, x^i, c, \mathcal{A}) - \phi^j\Big)^2$$

regarding the vector $c \in \mathbb{R}^{|\mathcal{V}^i|}$, where $\omega_{ij} \geq 0$ stands for the weights between nodes i and j. Minimisation provides a vector c^i solution of the matrix problem

$$\Big[M^i W^i (M^i)^{\mathrm{t}}\Big] c^i = M^i W^i \boldsymbol{\Phi}_{\mathcal{V}^i} \tag{1}$$

with $W^i = \mathrm{diag}(\omega^2_{ij})$ the square diagonal $|\mathcal{V}^i| \times |\mathcal{V}^i|$ matrix of the weights and $M^i[\alpha, j] = m^{i,\alpha}_j$ the $|\mathcal{A}| \times |\mathcal{V}^i|$ matrix given by

$$m^{i,\alpha}_j := \left(\frac{x^j - x^i}{h_M}\right)^\alpha, \quad \alpha \in \mathcal{A},\ j \in \mathcal{V}^i. \tag{2}$$

Remark 1. In practice and for implementation purposes, we use a one-to-one index $\alpha \in \mathcal{A} \to \ell$ and a one-to-one local index mapping $j \in \mathcal{V}^i \to \imath$ to store the information in $M[\ell, \imath]$. We do not refer to the local indexation for simplicity and $M[\alpha, j]$ has to be interpreted as $M[\ell, \imath]$.

The stencil $\mathcal{V}^i$ is said to be resolvent if M^i has the maximum rank equal to $|\mathcal{A}|$ assuming that $|\mathcal{V}^i| \geq |\mathcal{A}|$. Indeed, we have enough information from the neighbouring nodes to completely determine the polynomial coefficients and the square matrix $M^i W^i (M^i)^{\mathrm{t}}$ is non-singular.

Noting that, for $\beta \in \mathcal{A}$, we have,

$$\partial^\beta\pi(x^i; h_M, x^i, c^i, \mathcal{A}) = c^i_\beta \frac{\beta!}{h_M^{|\beta|}}.$$

We then obtain the relation between the partial derivatives and the coefficients of the vector c^i. To deliver an explicit relation regarding the value of ϕ^j with $j \in \mathcal{V}^i$, we introduce the diagonal matrix

$$D = \left(\frac{\beta!}{h_M^{|\beta|}}\delta^{\beta,\alpha}\right)_{\beta,\alpha\in\mathcal{A}} = \mathrm{diag}\left[\left(\frac{\beta!}{h_M^{|\beta|}}\right)_{\beta\in\mathcal{A}}\right]$$

and define, using relation (1), the $|\mathcal{A}| \times |\mathcal{V}^i|$ matrix

$$A^i = D\Big[M^i W^i (M^i)^{\mathrm{t}}\Big]^{-1} M^i W^i.$$

Letting $a^{i,\beta} = M^i[\beta, :] \in \mathbb{R}^{|\mathcal{V}^i|}$ and vector $\boldsymbol{\Phi}_{\mathcal{V}^i}$ the data in the neighbourhood of node i, we obtain the discretisation of the β-derivative at node i given by

$$\phi^{i,\beta} := a^{i,\beta}\boldsymbol{\Phi}_{\mathcal{V}^i} = \sum_{j\in\mathcal{V}^i} a^{i,\beta}_j \phi^j \approx \partial^\beta\phi(x^i), \tag{3}$$

which is exact for all polynomials of degrees in $\mathcal{A}$.

3.2. Lagrangian multipliers formulation

We present an alternative approach to produce the relation (3) but with possible extensions that we shall detail in the next section. The idea consists in introducing the relation between the derivatives and the values over the stencil as constraints. To this end, for $i \in \mathcal{G}$ and $\beta \in \mathcal{A}$, we define the functional

$$E^{i,\beta}(\phi; a) := \partial^\beta \phi(x^i) - \sum_{j \in \mathcal{V}^i} a_j^{i,\beta} \phi(x^j). \tag{4}$$

To determine the coefficients $a_j^{i,\beta}$, we first prescribe the constraints $E^{i,\beta}(\phi; a) = 0$ for all $\phi(x) = \left(\frac{x-x^i}{h_M}\right)^\alpha$, $\alpha \in \mathcal{A}$, that is

$$\left[\partial^\beta \left(\frac{x-x^i}{h_M}\right)^\alpha\right]_{x=x^i} = \sum_{j \in \mathcal{V}^i} a_j^{i,\beta} \left(\frac{x^j-x^i}{h_M}\right)^\alpha, \quad \forall \alpha \in \mathcal{A}.$$

The finite set $\mathcal{A}$ characterises the space of constraints that controls the accuracy of the approximations.

Gathering all the constraints for $\alpha \in \mathcal{A}$, we get the equality in $\mathbb{R}^{|\mathcal{A}|}$,

$$a^{i,\beta} M^{\mathrm{t}} = M (a^{i,\beta})^{\mathrm{t}} = d^\beta$$

where $a^{i,\beta}$ is a row vector of $\mathbb{R}^{|\mathcal{V}^i|}$, matrix M^i is the $|\mathcal{A}| \times |\mathcal{V}^i|$ matrix given by (2) and $d^\beta \in \mathbb{R}^{|\mathcal{A}|}$ the canonical vector, with zero everywhere except $\beta!/h_M^{|\beta|}$ for entry β. Since we have the same relation for every $\beta \in \mathcal{A}$, we rewrite the constraints in matrix form

$$A^i (M^i)^{\mathrm{t}} = M^i (A^i)^{\mathrm{t}} = D = D^{\mathrm{t}},$$

where we collect the row vector $a^{i,\beta}$ in the $|\mathcal{A}| \times |\mathcal{V}^i|$ matrix A^i.

Since $|\mathcal{V}^i| \geq |\mathcal{A}|$, we do not have uniqueness for the coefficients $a_j^{i,\beta}$ (except for the particular case $|\mathcal{V}^i| = |\mathcal{A}|$), and the number of constraints is not enough to determine the vector $a^{i,\beta}$. Consequently, we introduce the weighted energy functional

$$G^i(a) := \frac{1}{2} a \left(W^i\right)^{-1} a^{\mathrm{t}}, \quad a \in \mathbb{R}^{|\mathcal{V}^i|},$$

and, for each $\beta \in \mathcal{A}$, we consider the (constraint) minimisation problem

$$a^{i,\beta} := \arg \min_{a \in \mathbb{R}^{|\mathcal{V}^i|}} G^i(a), \quad \text{with } M^i a^{\mathrm{t}} = d^\beta,$$

where a is a row vector of the coefficients. An important note is: on the contrary to the polynomial approach, we can determine vectors $a^{i,\beta}$, $\beta \in \mathcal{A}$, independently, that is, we do not need to compute the whole matrix A^i but only the derivatives of interest.

Minimising G^i under the constraints $E^{i,\beta}\left(\left(\frac{x-x^i}{h_M}\right)^\alpha; a\right) = 0$ for all $\alpha \in \mathcal{A}$ using the Lagrangian multipliers provides the system

$$a^{i,\beta} (W^i)^{-1} + \sum_{\alpha \in \mathcal{A}} \lambda_\alpha^{i,\beta} (m^{i,\alpha})^{\mathrm{t}} = 0,$$

with $m^{i,\alpha} = M[\alpha, :]$. Setting $\lambda^{i,\beta} = (\lambda_\alpha^{i,\beta})_{\alpha \in \mathcal{A}}$, the Lagrangian reads

$$(W^i)^{-1} (a^{i,\alpha})^{\mathrm{t}} + \lambda^{i,\beta} M^i = 0, \text{ in } \mathbb{R}^{|\mathcal{A}|}.$$

Gathering the line vectors $\lambda^{i,\beta}$ into the matrix Λ^i, we obtain the relation

$$A + \Lambda^i M^i W^i = 0$$

we have to solve together with the constraint $A(M^i)^{\mathrm{t}} = D$. After algebraic operations, we deduce $\Lambda^i = -D(M^i W^i (M^i)^{\mathrm{t}})^{-1}$ and we find, once again,

$$A^i = D(M^i W^i (M^i)^{\mathrm{t}})^{-1} M^i W^i.$$

Remark 2. As we mentioned above, the method enables to compute the vector $a^{i,\alpha}$ corresponding to the derivatives' discretisation of interest, and, in that way, reduce the computational effort. For example, assume that $|\mathcal{A}|$ is large (for example 15 for $|\alpha| \leq 5$) and that we only need the discretisation of $\partial^2_{x_1} = \partial^{(2,0)}$ and $\partial^2_{x_2} = \partial^{(0,2)}$. Therefore, we just determine the two vectors corresponding to these two derivatives.

4. Optimisation

The existence of a discrete representation of the derivatives requires that the matrix $K^i = M^i W^i (M^i)^{\mathrm{t}}$ is non-singular. We claim that accuracy is strongly related to its condition number, $\chi^i = \chi(K^i)$, and reducing the condition number shall improve the quality of the approximation of the derivatives (see Section 5.2.1 for the justification). By analysing the construction of M^i and W^i, we identify three ingredients that could help to lower χ^i: (1) the stencil choice, (2) the polynomial normalisation parameter, and (3) the weights. We shall consider such ingredients as independent degrees of freedom to compute χ^i.

To provide the weights, we consider a decreasing function $s \to \omega(s)$ defined on $[0, +\infty[$ with $\omega(0) = 1$ and $\omega(+\infty) = 0$. We then set

$$\omega_j^i = \omega\left(\frac{|x^j - x^i|}{h_W}\right),$$

where the scaling parameter h_W controls the kernel range.

Consequently, the condition number $\chi^i = \chi(K^i) = \chi(\mathcal{V}^i, h_M, h_W)$ depends on $\mathcal{V}^i$, h_M, h_W, which control, respectively, the stencil, matrix M and the weights W. We intend to minimise χ regarding these three parameters. The optimisation procedure involves two continuous variables and a discrete one, hence requiring a different approach in function of the nature of the parameter addressed.

4.1. Stencil Size optimisation (SSO)

In the first stage of optimisation, we shall deal with parameter $\mathcal{V}^i$. The discrete optimisation consists in starting with a large initial stencil of G, usually twice the number of strictly necessary nodes, around the node i and performing different selections of nodes to reduce both the stencil size (computational effort reduction) and the condition number.

Dropping index i for clarity, we consider the operator $\mathcal{V} \to \chi(\mathcal{V}) = \chi(\mathcal{V}, h_M, h_W)$, where the matrix $M = M(\mathcal{V})$ has been constructed using the nodes of the stencil $\mathcal{V}$, parameters h_M and h_W being frozen. The Stencil Size Optimisation (SSO) is based on a brute force-like technique to produce a sequence $\mathcal{V}(m)$ such that $|\mathcal{V}(m+1)| = |\mathcal{V}(m)| - 1$ while avoiding increasing the condition number $\chi(\mathcal{V}(m))$. Note that we require that the initial stencil $\mathcal{V}(0)$ guarantees that the matrix $M(\mathcal{V})$ enjoys the maximal rank property. The iterative procedure reads:

1. Set the initial stencil $\mathcal{V}(0)$ with maximum rank property;
2. Given the stencil $\mathcal{V}(m)$ at stage m, we seek the stencil $\mathcal{V}(m+1) \subset \mathcal{V}(m)$ that minimises χ by subtracting one node to $\mathcal{V}(m)$. In other words, we define the candidate stencil as
$$\mathcal{W} = \arg\min_{j \in \mathcal{V}(m)} \chi\big(\mathcal{V}(m) \setminus \{j\}\big).$$
3. If $\chi(\mathcal{W}) \le (1+\epsilon)\chi(\mathcal{V}(m))$ then we set $\mathcal{V}(m+1) = \mathcal{W}$ and go back to step (2).
4. Otherwise, the final stencil is given by $\mathcal{V}'_{\text{sso}} = \mathcal{V}(m)$.

A strict reduction of the condition number would be quite restrictive, and one can consider a trade-off between stencil size reduction and conditioning. Therefore, we introduce the tolerance factor ϵ that quantifies the flexibility to let the condition number grow. Notice that $\epsilon > 0$ is a tolerance, whereas $\epsilon < 0$ is a penalisation.

In this way, we define an operator such that, given an initial stencil $\mathcal{V}$, it provides the optimal stencil $\mathcal{V}'_{\text{sso}} \subset \mathcal{V}$ that minimises χ by using the nodes of $\mathcal{V}$ uniquely. To assess the quality of the optimal stencil, we define the Condition Number Ratio (CNR) and the Stencil Size Ratio (SSR) associated to the stencil $\mathcal{V}$ as

$$\text{CNR}(\mathcal{V}) = \frac{\chi(\mathcal{V})}{\chi(\mathcal{V}'_{\text{sso}})}, \quad \text{SSR}(\mathcal{V}) = \frac{|\mathcal{V}'_{\text{sso}}|}{|\mathcal{V}|} \ge 1.$$

These coefficients assess the gains of the optimisation procedure in terms of condition number and stencil size.

4.2. Scaling Parameters Optimisation (SPO)

In addition to the stencil optimisation, parameters h_M and h_W that control the scaling of matrices M and W, respectively, will be optimised to reduce the condition number. We propose to use a greedy algorithm where we successively optimise the stencil and the parameters.

To this end, given an initial stencil $\mathcal{V}(0) = \mathcal{V}$ and initial parameters $h_M(0) = h_M$, $h_W(0) = h_W$, we build a sequence $\big(\mathcal{V}(m), h_M(m), h_W(m)\big)$ such that $|\mathcal{V}(m+1)| = |\mathcal{V}(m)| - 1$ and

$$\chi\big(\mathcal{V}(m+1), h_M(m+1), h_W(m+1)\big) \le (1+\epsilon)\chi\big(\mathcal{V}(m), h_M(m), h_W(m)\big),$$

with $\epsilon > 0$ for a relaxed constraint or $\epsilon < 0$ for a tighter constraint.

The greedy procedure consists in two stages:

1. Given $\mathcal{V}(m)$, $h_M(m)$, $h_W(m)$, we first optimise the discrete problem by determining the new stencil
$$\mathcal{W}(m+1) = \arg\min_{j \in \mathcal{V}(m)} \chi\big(\mathcal{V}(m) \setminus \{j\}, h_M(m), h_W(m)\big).$$
2. With the new stencil frozen, we optimise the continuous parameters under positivity constraints
$$(\bar{h}_M, \bar{h}_W) = \arg\min_{h_M>0,\, h_W>0} \chi\big(\mathcal{W}(m+1), h_M, h_W\big).$$

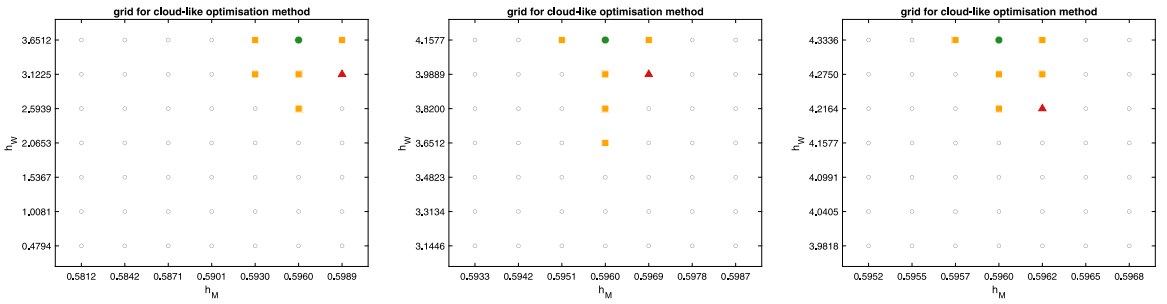

Fig. 1. The cloud-like optimisation procedure. Illustration of the (h_M^r, h_W^r) convergence towards $(\bar{h}_M, \bar{h}_W)$ for a 7×7 points uniform cloud. The panels (from left to right) present the results obtained for three consecutive iterations. The green filled circle corresponds to $(h_M^{r,1}, h_W^{r,1})$, the orange squares to $(h_M^{r,\ell}, h_W^{r,\ell})$, $\ell = 2, \dots, \mathcal{L}-1$, and the red triangle to $(h_M^{r,\mathcal{L}}, h_W^{r,\mathcal{L}})$.

We then update $\mathcal{V}(m+1) = \mathcal{W}(m+1)$, $h_M(m+1) = \bar{h}_M$, $h_W(m+1) = \bar{h}_W$ if

$$\chi\big(\mathcal{W}(m+1), \bar{h}_M, \bar{h}_W\big) \leq (1+\epsilon)\chi\big(\mathcal{V}(m), h_M(m), h_W(m)\big).$$

Otherwise, the procedure ends. The main issue is now the construction and implementation of the minimiser operator

$$\mathcal{O}(\mathcal{V}, h_M, h_W) = \underset{h_M>0,\, h_W>0}{\arg\min}\ \chi\big(\mathcal{W}, h_M, h_W\big).$$

that optimises the two parameters for a given stencil $\mathcal{W}$.

4.2.1. The `fminsearch` optimisation procedure

A first minimisation procedure for $\mathcal{O}(\mathcal{W}, h_M, h_W)$ recurs to the MATLAB® function `fminsearch`. This function finds the minimum of an unconstrained multi-variable function without recurring to numerical or analytic gradients and uses a simplex search method of Lagarias et al. [17]. This kind of approach is adequate for our minimisation problem, since the condition function $\chi(\mathcal{W}, h_M, h_W)$ is only piecewise differentiable.

4.2.2. A cloud-like optimisation procedure

We propose an alternative optimisation procedure to the `fminsearch` method. We take advantage of the piecewise differential character of $\chi(\mathcal{V}, h_M, h_W)$ by designing a derivative-free procedure to compute the optimal parameters $(\bar{h}_M, \bar{h}_W)$ (see Fig. 1).

Given $\mathcal{W}(m+1)$, $h_M(m)$, $h_W(m)$, we shall build a sub-sequence $h_M^r = h_M(m,r)$, $h_W^r = h_W(m,r)$ together with $\Delta_M^r = \Delta h_M(m,r)$, $\Delta_W^r = \Delta h_W(m,r)$ and the rectangle

$$S^r = [h_M^r - \Delta_M^r, h_M^r + \Delta_M^r] \times [h_W^r - \Delta_W^r, h_W^r + \Delta_W^r]$$

such that $(h_M^r, h_W^r) \to (\bar{h}_M, \bar{h}_W)$ with $\Delta_M^r, \Delta_W^r \to 0$. To this end, we use a cloud of $\mathcal{L} \times \mathcal{L}$ points ($\mathcal{L}$ is an odd, user-defined, integer number) uniformly spanned over the rectangle S^r. We order the points of the cloud $(h_M^{r,\ell}, h_W^{r,\ell}) \in S^r$, with index $\ell = 1, \dots, \mathcal{L}^2$, such that $\chi^{r,\ell} = \chi(\mathcal{W}, h_M^{r,\ell}, h_W^{r,\ell})$ is a non-decreasing sequence regarding ℓ.

From the sub-sequence, we extract the discrete minimum $\ell = 1$

$$h_M^{r+1} = h_M^{r,1}, \quad h_W^{r+1} = h_W^{r,1},$$

while we update the size of the rectangle with the intermediate point $\ell = \mathcal{L}$:

$$\Delta_M^{r+1} = |h_M^{r,1} - h_M^{r,\mathcal{L}}|, \quad \Delta_W^{r+1} = |h_W^{r,1} - h_W^{r,\mathcal{L}}|.$$

The procedure ends when both ratios, Δ_M^r / h_M^r and Δ_W^r / h_W^r, are smaller than a prescribed value ϵ_C.

Remark 3. To initialise the procedure, we take $h_M(m,0) = h_M(m)$ and $h_W(m,0) = h_W(m)$ while $\Delta h_M(m,0) = \theta h_M(m)$, $\Delta h_W(m,0) = \theta h_W(m)$ with $\theta \in]0,1[$ a user constant.

Remark 4. To ensure the convergence of the sequence and the positivity of the h values, we add a threshold correction to ensure that $\Delta_M^{r+1} = [(1-\theta)h_M^{r+1}, \theta h_M^{r+1}]$. To this end, we perform a correction stage with

$$\Delta_M^{r+1} := \max\Big((1-\theta)h_M^{r+1}, \min\big(\theta h_M^{r+1}, \Delta_M^{r+1}\big)\Big).$$

We proceed similarly for h_W. Such a correction guarantees the convergence of the procedure, since we automatically reduce the size of the interval of a factor θ. Note that all the benchmarks we present in the numerical section have been carried out with $\theta = 0.99$.

Table 1
Approximation errors and convergence orders for the random distribution case with SSO for $p = 2$. Approximations for the first- and second-order derivatives are considered for different scales (S) leading to an average converge order denoted ACO (for $S = 1 \to 1/16$). The values of parameters h_M and h_W are also presented.

S		1	1/2	1/4	1/8	1/16	1/32	1/64	ACO
∂_x	Error	1.05e−01	2.50e−02	6.17e−03	1.54e−03	3.83e−04	9.58e−05	2.39e−05	
	Order	–	2.1	2.0	2.0	2.0	2.0	2.0	2.0
∂_y	Error	2.55e−01	6.20e−02	1.54e−02	3.85e−03	9.61e−04	2.40e−04	6.01e−05	
	Order	–	2.0	2.0	2.0	2.0	2.0	2.0	2.0
∂_{xx}	Error	6.29e−02	1.45e−02	3.01e−03	4.43e−04	4.62e−04	9.03e−05	6.20e−05	
	Order	–	2.1	2.3	2.8	−0.1	2.4	0.5	1.8
∂_{yy}	Error	2.46e−02	6.74e−02	4.75e−02	2.73e−02	1.45e−02	7.48e−03	3.79e−03	
	Order	–	−1.5	0.5	0.8	0.9	1.0	1.0	0.2
$h_M = h_W$		3.22e−01	1.61e−01	8.04e−02	4.02e−02	2.01e−02	1.01e−02	5.03e−03	

Table 2
Condition number of matrix MWM^t and stencil excess size ratio (SESR) for the random distribution case with SSO, before and at the end of the SSO procedure, for different p values. The values of the condition number and SESR are independent of the scale value S.

p	Condition number			SESR (%)	
	Initial	Final	CNR	Initial	Final
2	1.7e+01	1.2e+01	1.4	100	17
3	2.2e+02	1.8e+02	1.2	100	10
4	1.1e+03	9.6e+02	1.1	100	7
5	7.4e+03	9.0e+03	0.8	100	10
6	3.3e+04	5.9e+04	0.6	100	0

5. Benchmarks

We assess the efficiency of the parameters' optimisation to produce derivatives with better accuracy and using a smaller stencil. We first deal with the optimisation regarding the stencil and then consider the full optimisation involving the three parameters. Since the optimisation procedure only involves a node and its neighbours, we drop the index i and use a local indexation where the reference node is $i = 1$ and the stencil reads $\mathcal{V}$.

Remark. For a better readability of the paper, Tables 28 to 97 are presented in an dedicated appendix.

5.1. Numerical benchmarks (SSO)

In a first set of benchmarks, we address the approximation accuracy, convergence order and condition number of MWM^t provided by the SSO procedure for different geometric distributions of the grid points. In this context, we deal with the approximation of the function e^{x+2y} at the origin of the coordinate system, $(x, y) = (0, 0)$, using a set of points occupying the domain $[-1, 1]\times[-1, 1]$. In order to evaluate the convergence order, the original domain is scaled by a factor $S = 1,\ 1/2,\ 1/4,\ 1/8,\ 1/16,\ 1/32,$ and $1/64$. We recall that set $\mathcal{A}$ characterises the constraints we enforce to provide an expected accuracy. We then use the notation

$$\mathcal{A}(p) = \{\alpha \in \mathbb{N}^2,\ |\alpha| \leq p\},$$

and by extension, we say that the reconstruction is of $p + 1$th order when using $\mathcal{A}(p)$ as the space of constraints.

5.1.1. Random distribution

The first benchmark deals with an initial stencil, $\mathcal{V}$, constituted of points randomly distributed around the origin (reference point). We aim at reducing the stencil size while maintaining the condition number within an interval controlled by the tolerance parameter ϵ.

To check the convergence, we produce a sequence of stencils by re-scaling the initial stencil with the scale factor S, and we report in Table 1 the convergence order for the first- and second-order derivatives for $\mathcal{A}(2)$. Similarly, Tables 28–31 present the convergence results obtained for $\mathcal{A}(p)$, $p = 3, 4, 5, 6$ respectively. Optimal error convergences are generally achieved for scale factors up to $1/64$, with a pth-order of convergence for ∂_x and ∂_y derivatives and $(p-1)$th-order for ∂_{xx} and ∂_{yy} derivatives. The convergence order shows an erratic behaviour for $p = 6$ when $S = 1/64$ due to loss of accuracy associated to truncation errors that contaminate the results. We note, however, that for $p = 6$ the approximation errors for $S = 1/32$ amount to less than 2×10^{-12} and 7×10^{-10} for the first- and second-derivative cases, respectively.

We plot in Fig. 12 the final stencil after applying the reduction of the number of points for $p = 2, \ldots, 6$. We have set $\epsilon = 0.2$ to alleviate the too restrictive condition on the conditioning leading to a poor reduction of the stencil, and will use this value

Table 3
Approximation errors and convergence orders for the uniform distribution case with SSO for $p = 2$. Approximations for the first- and second-order derivatives are considered for different scales (S) leading to an average converge order denoted ACO (for $S = 1 \to 1/16$). The values of parameters h_M and h_W are also presented.

S		1	1/2	1/4	1/8	1/16	1/32	1/64	ACO
∂_x	Error	4.07e−02	1.05e−02	2.67e−03	6.75e−04	1.70e−04	4.25e−05	1.06e−05	
	Order	–	2.0	2.0	2.0	2.0	2.0	2.0	2.0
∂_y	Error	1.00e−01	2.59e−02	6.61e−03	1.67e−03	4.20e−04	1.05e−04	2.64e−05	
	Order	–	1.9	2.0	2.0	2.0	2.0	2.0	2.0
∂_{xx}	Error	1.21e−01	6.86e−02	3.65e−02	1.88e−02	9.56e−03	4.82e−03	2.42e−03	
	Order	–	0.8	0.9	1.0	1.0	1.0	1.0	0.9
∂_{yy}	Error	5.51e−01	3.22e−01	1.74e−01	9.07e−02	4.62e−02	2.33e−02	1.17e−02	
	Order	–	0.8	0.9	0.9	1.0	1.0	1.0	0.9
$h_M = h_W$		3.61e−01	1.80e−01	9.02e−02	4.51e−02	2.26e−02	1.13e−02	5.64e−03	

Table 4
Condition number of matrix MWM^t and stencil excess size ratio (SESR) for the uniform distribution case with SSO, before and at the end of the SSO procedure, for different p values. The values of the condition number and SESR are independent of the scale value S.

p	Condition number			SESR (%)	
	Initial	Final	CNR	Initial	Final
2	2.0e+01	2.9e+01	0.7	100	33
3	5.6e+01	1.3e+02	0.4	100	10
4	3.9e+02	5.0e+02	0.8	100	0
5	3.3e+03	5.5e+03	0.6	100	14
6	9.6e+03	2.7e+04	0.4	100	0

throughout the simulations. The number of points for the initial stencil is the double of the minimal number of points to provide a matrix M of maximal rank, *i.e.*, $|\mathcal{V}| = 12$ for $p = 2$, $|\mathcal{V}| = 20$ for $p = 3$, $|\mathcal{V}| = 30$ for $p = 4$, $|\mathcal{V}| = 42$ for $p = 5$, and $|\mathcal{V}| = 56$ for $p = 6$. This will be the setting for all the simulations performed. We observe a relative uniform distribution of the selected points (red filled circles) regarding the distance and angular configuration.

Remark. Notice that the analysis of the location of the selected points in the different panels of Fig. 12, particularly those corresponding to $p = 3, \dots, 6$, shows that the SSO methodology manages to avoid the selection of points that are very close to each other by discarding at least one of the points involved.

We report in Table 2 the condition number in function of p before and after the stencil reduction. For $p = 2, 3, 4$, we manage to reduce the condition together with the stencil size (CNR up to 1.4) while for higher values of p, the stencil size reduction implies a moderate increase in the condition number (CNR down to 0.6).

To quantify the stencil size reduction, we define the stencil excess size ratio (SESR) as

$$\text{SESR}(\mathcal{V}) = \frac{|\mathcal{V}| - |\mathcal{V}_{min}|}{|\mathcal{V}_{min}|} = 2\ \text{SSR}(\mathcal{V}) - 1,$$

where $|\mathcal{V}_{min}|$ is the minimal number of points to provide maximal rank. For example, for $p = 2$, the minimal number of points is $|\mathcal{V}min| = 6$ and the stencil obtained has 17% more points (*i.e.*, 1).

5.1.2. Uniform distribution

The uniform case concerns a uniform distribution of the points, as depicted in Fig. 13. We proceed similarly as we did for the random case and report in Tables 3 ($p = 2$) and 32–35 ($p = 3, \dots, 6$) the errors and convergence order for the different values of p for the first- and second-order derivatives. The results show evidence of the optimal convergence order of the reconstructions for all the scale factors, except for the $p = 6$ and $S = 1/64$ case, where the approximation error for the first- and second-order derivatives stays below 3×10^{-12} and 10^{-9}, respectively.

We plot in Fig. 13 the final stencil after size reduction for $p = 2, \dots, 6$. We notice the good symmetry of the points' distribution with a regular distribution around the reference node.

Table 4 gives the condition numbers before and after the stencil reduction for the different values of p. The SESR value is in the range $[0\%, 33\%]$, while the conditioning is well-controlled, only moderately larger than the initial value (CNR ranging from 0.4 to 0.8).

Finally, and for the sake of comparison, we present the results obtained for the approximation errors and convergence order, for both first- and second-order derivatives, using the standard (3 points) one-dimension second-order centred differences (*cf.* Table 36) and the standard (5 points) one-dimension fourth-order centred differences (*cf.* Table 37) schemes. The convergence order is the one expected in both cases, while the magnitude of the approximation errors is smaller than the one obtained with the SSO procedure

Table 5
Approximation errors and convergence orders for the half-plane distribution case with SSO for $p = 2$. Approximations for the first- and second-order derivatives are considered for different scales (S) leading to an average converge order denoted ACO (for $S = 1 \to 1/16$). The values of parameters h_M and h_W are also presented.

S		1	1/2	1/4	1/8	1/16	1/32	1/64	ACO
∂_x	Error	6.55e−02	1.55e−02	3.77e−03	9.30e−04	2.31e−04	5.75e−05	1.44e−05	
	Order	–	2.1	2.0	2.0	2.0	2.0	2.0	2.0
∂_y	Error	1.23e−01	2.78e−02	6.63e−03	1.62e−03	4.01e−04	9.98e−05	2.49e−05	
	Order	–	2.1	2.1	2.0	2.0	2.0	2.0	2.1
∂_{xx}	Error	6.32e−02	3.10e−02	1.53e−02	7.59e−03	3.78e−03	1.89e−03	9.43e−04	
	Order	–	1.0	1.0	1.0	1.0	1.0	1.0	1.0
∂_{yy}	Error	1.15e+00	4.97e−01	2.31e−01	1.12e−01	5.48e−02	2.72e−02	1.35e−02	
	Order	–	1.2	1.1	1.0	1.0	1.0	1.0	1.1
$h_M = h_W$		3.51e−01	1.75e−01	8.77e−02	4.39e−02	2.19e−02	1.10e−02	5.48e−03	

Table 6
Condition number of matrix MWM^t and stencil excess size ratio (SESR) for the half-plane distribution case with SSO, before and at the end of the SSO procedure, for different p values. The values of the condition number and SESR are independent of the scale value S.

p	Condition number			SESR (%)	
	Initial	Final	CNR	Initial	Final
2	3.7e+02	3.0e+02	1.2	100	0
3	1.2e+04	8.8e+03	1.4	100	10
4	1.9e+05	1.3e+05	1.5	100	0
5	4.5e+06	5.8e+06	0.8	100	0
6	4.0e+08	2.5e+08	1.6	100	0

having the same order of convergence. We report that the second-order schemes yield very similar results for the first-order derivative with respect to y, whereas within a factor 3 for both the first-order derivative with respect to x and the second-order derivative with respect to y, while within a factor 50 for the second-order derivative with respect to x. The error ratio for the fourth-order schemes is higher, ranging from 4 (first-order derivative with respect to y) to 135 (second-order derivative with respect to x). In any case, it should be noted that the fact that the function whose derivatives are being approximated has a separated form for the independent variables favours the one-dimensional character of the standard schemes, helping to justify the performance gap between the two families of schemes.

5.1.3. The half-plane distribution

To check the method for a node close to the boundary, we now assume that all points of the stencil belong to the right half-plane, randomly distributed. Such a situation arises when the reference node belongs to the boundary and all the information is located inside the right (or left) domain (see Fig. 14). Following the previous examples, we built an initial stencil with $|\mathcal{V}| = 12$, 20, 30, 42 and 56 points for $p = 2$, 3, 4, 5 and 6, respectively, and re-scaled the stencil setting $S = 1, 1/2, \dots, 1/64$ to assess the error and the convergence order. The corresponding results are reported in Tables 5 ($p = 2$) and 38–41 ($p = 3, \dots, 6$) for the different values of p. Optimal effective order is obtained for both the first- and second-order derivatives, except for the $p = 5, 6$ cases when $S = 1/32, 1/64$. Again, we observe very low approximation errors associated with the misbehaviour of the convergence.

Fig. 14 displays the initial and final stencil for the five reconstruction settings. Uniform distribution of the selected points is again noticeable. Table 6 provides the condition number and the stencil excess size ratio. We manage to reduce the stencil cardinal by a factor two with respect to the initial setting, while the condition number is generally reduced (CNR value below 1 only for $p = 5$). On the other hand, we note that the condition number is larger than the one corresponding to the random distribution case on the whole space (roughly, one magnitude higher). Indeed, up-winding naturally results from this one-side configuration and strongly impacts the reconstruction.

5.1.4. The regular convex distribution

Boundary conditions may be sensitive to the curvature of the simulation domain frontier, and therefore the assessment of the first- and second-order derivatives is important to check the accuracy of the method for such situations. This paragraph deals with the regular convex case (displayed in Fig. 15), whereas the more sensitive regular concave situation is presented in the next paragraph. The boundary is represented as a piece of circle of unit radius and all the points are situated inside the circle. We re-scale the initial configuration with parameter S and, therefore, the radius varies with S (larger curvature when S decreases). Indeed, if the radius would not change with S, we would quickly recover the half-plane case since the distance between the points is one or two magnitudes lower than the radius.

We report in Tables 7 ($p = 2$) and 42–45 ($p = 3, \dots, 6$) the error and convergence order obtained for the different values of p with respect to the scale factor S. Similarly to the former cases, the optimal converge is achieved with a pth-order for the first-order derivative and $(p-1)$th-order for the second-order case as long as $S \geq 1/16$. We display in Fig. 15 the location of the initial stencil

Table 7
Approximation errors and convergence orders for the regular convex distribution case with SSO for $p = 2$. Approximations for the first- and second-order derivatives are considered for different scales (S) leading to an average converge order denoted ACO (for $S = 1 \to 1/16$). The values of parameters h_M and h_W are also presented.

S		1	1/2	1/4	1/8	1/16	1/32	1/64	ACO
∂_x	Error	3.37e−02	7.57e−03	1.80e−03	4.38e−04	1.08e−04	2.68e−05	6.69e−06	
	Order	–	2.2	2.1	2.0	2.0	2.0	2.0	2.1
∂_y	Error	5.00e−02	1.04e−02	2.38e−03	5.68e−04	1.39e−04	3.43e−05	8.51e−06	
	Order	–	2.3	2.1	2.1	2.0	2.0	2.0	2.1
∂_{xx}	Error	3.24e−01	1.49e−01	7.17e−02	3.52e−02	1.74e−02	8.66e−03	4.32e−03	
	Order	–	1.1	1.1	1.0	1.1	1.0	1.0	1.1
∂_{yy}	Error	1.02e+00	4.60e−01	2.19e−01	1.07e−01	5.27e−02	2.62e−02	1.31e−02	
	Order	–	1.1	1.1	1.0	1.0	1.0	1.0	1.1
$h_M = h_W$		2.61e−01	1.30e−01	6.52e−02	3.26e−02	1.63e−02	8.15e−03	4.07e−03	

Table 8
Condition number of matrix MWM^t and stencil excess size ratio (SESR) for the regular convex distribution case with stencil size optimisation, before and at the end of the stencil size optimisation procedure, for different p values. The values of the condition number and SESR are independent of the scale value S.

p	Condition number			SESR (%)	
	Initial	Final	CNR	Initial	Final
2	1.5e+03	7.7e+02	1.9	100	0
3	4.7e+04	2.9e+04	1.6	100	0
4	2.2e+06	1.9e+06	1.2	100	7
5	4.9e+07	6.3e+07	0.8	100	5
6	1.1e+09	1.1e+09	1.0	100	4

Table 9
Approximation errors and convergence orders for the regular concave distribution case with SSO for $p = 2$. Approximations for the first- and second-order derivatives are considered for different scales (S) leading to an average converge order denoted ACO (for $S = 1 \to 1/16$). The values of parameters h_M and h_W are also presented.

S		1	1/2	1/4	1/8	1/16	1/32	1/64	ACO
∂_x	Error	1.83e−02	4.48e−03	1.11e−03	2.76e−04	6.88e−05	1.72e−05	4.30e−06	
	Order	–	2.0	2.0	2.0	2.0	2.0	2.0	2.0
∂_y	Error	3.94e−02	9.62e−03	2.38e−03	5.92e−04	1.48e−04	3.69e−05	9.21e−06	
	Order	–	2.0	2.0	2.0	2.0	2.0	2.0	2.0
∂_{xx}	Error	1.21e−01	5.77e−02	2.82e−02	1.39e−02	6.92e−03	3.45e−03	1.72e−03	
	Order	–	1.1	1.0	1.0	1.0	1.0	1.0	1.0
∂_{yy}	Error	2.70e−01	1.24e−01	5.91e−02	2.89e−02	1.43e−02	7.10e−03	3.54e−03	
	Order	–	1.1	1.1	1.0	1.0	1.0	1.0	1.1
$h_M = h_W$		2.32e−01	1.16e−01	5.79e−02	2.90e−02	1.45e−02	7.24e−03	3.62e−03	

points together with the points of the reduced stencil. Again, we notice a good distance and angular distribution with respect to the reference node. As reported in Table 8, the size reduction is of the same order of the half-plane configuration with $\leq 7\%$ of additional points regarding the minimum number of points required to guarantee an eligible reconstruction. On the other hand, the condition number before and after the stencil reduction presents a moderate increase for $p = 5$ (CNR equal to 0.8) and an inverse tendency otherwise (CNR ranging from 1.0 to 1.9). We note the considerable increase of the condition number with p, reaching 1.1×10^9 for $p = 6$.

5.1.5. The regular concave distribution

The regular concave case consists in choosing the stencil points outside the unit circle, as shown in Fig. 16. Such a configuration is more difficult to solve due to the non-convexity of the domain. Nevertheless, the optimal convergence order is achieved, as shown in Tables 9 ($p = 2$) and 46–49 ($p = 3, \ldots, 6$), in a way very similar to the regular convex case.

We plot in Fig. 16 the initial and final stencils. Notice that for $p = 2$ and $p = 3$ the distribution is very close to the half-plane configuration, whereas the stencil for $p \geq 4$ is really impacted by the concave shape with some points located on the two sides of the circle. In Table 10 we notice a considerable increase in the condition number for both the initial and final configurations in comparison with the regular convex case for all the values of p, while the CNR value stays in a narrow range close to 1. On the other hand, we report a reduction of the stencil size of 50% independently of the p value considered.

Table 10
Condition number of matrix MWM^t and stencil excess size ratio (SESR) for the regular concave distribution case with stencil size optimisation, before and at the end of the stencil size optimisation procedure, for different p values. The values of the condition number and SSR are independent of the scale value S.

p	Condition number			SESR (%)	
	Initial	Final	CNR	Initial	Final
2	3.5e+02	3.7e+02	0.9	100	0
3	6.0e+03	5.8e+03	1.0	100	0
4	7.1e+04	7.4e+04	1.0	100	0
5	6.6e+05	7.8e+05	0.8	100	0
6	8.9e+06	6.7e+06	1.3	100	0

Table 11
Approximation errors and convergence orders for the convex corner distribution case with SSO for $p = 2$. Approximations for the first- and second-order derivatives are considered for different scales (S) leading to an average converge rate denoted ACO (for $S = 1 \rightarrow 1/16$). The values of parameters h_M and h_W are also presented.

S		1	1/2	1/4	1/8	1/16	1/32	1/64	ACO
∂_x	Error	2.02e−01	3.39e−02	6.93e−03	1.56e−03	3.72e−04	9.06e−05	2.23e−05	
	Order	–	2.6	2.3	2.2	2.1	2.0	2.0	2.3
∂_y	Error	7.76e−01	1.46e−01	3.19e−02	7.44e−03	1.80e−03	4.43e−04	1.10e−04	
	Order	–	2.4	2.2	2.1	2.0	2.0	2.0	2.2
∂_{xx}	Error	1.11e+00	3.62e−01	1.45e−01	6.46e−02	3.05e−02	1.48e−02	7.29e−03	
	Order	–	1.6	1.3	1.2	1.1	1.0	1.0	1.3
∂_{yy}	Error	4.10e+00	1.65e+00	7.46e−01	3.55e−01	1.73e−01	8.56e−02	4.25e−02	
	Order	–	1.3	1.1	1.1	1.0	1.0	1.0	1.1
$h_M = h_W$		3.35e−01	1.67e−01	8.37e−02	4.19e−02	2.09e−02	1.05e−02	5.23e−03	

Table 12
Condition number of matrix MWM^t and stencil excess size ratio (SESR) for the convex corner distribution case with SSO, before and at the end of the SSO procedure, for different p values. The values of the condition number and SESR are independent of the scale value S.

p	Condition number			SESR (%)	
	Initial	Final	CNR	Initial	Final
2	4.1e+03	2.3e+03	1.8	100	0
3	1.1e+05	8.7e+04	1.9	100	0
4	7.8e+06	7.6e+06	1.0	100	0
5	3.8e+08	2.5e+08	1.5	100	5
6	3.0e+09	2.5e+09	1.2	100	0

5.1.6. The convex corner distribution

General Lipschitz boundaries may involve wedges or corners characterised by a discontinuity of the outward normal leading to an infinite curvature. The solution regularity may be affected around the corner and the accuracy dramatically reduced. On the other hand, although the solution is regular at the corner, the specific geometric configuration of the stencil in the corner represents a challenge to recover the optimal convergence. We consider the convex case with a corner situated at the origin, as depicted in Fig. 17. We follow the same methodology as before and consider for $p = 2, \dots, 6$ initial stencils with $(p+1)(p+2)$ points. We report in Tables 11 ($p = 2$) and 50–53 ($p = 3, \dots, 6$) the error and convergence order for $p = 2, \dots, 6$ after the stencil reduction. We obtain the optimal order for both first- and second-order derivatives for $p = 2, 3$ independently of the scale factor, for $p = 4$ when $S \geq 1/16$, and for $p = 5, 6$ in the $S \geq 1/8$ cases. The high condition number observed for $p = 5, 6$ (see further below) impacts the convergence behaviour for the lowest values of S.

Fig. 17 presents the points that the algorithm has picked-up for the final stencil for the different values of p. Similarly to the previous configuration, a rather uniform distribution of the final stencil points characterises the output of the stencil size reduction algorithm. We present in Table 12 the conditioning of the linear systems associated to the initial and final stencil. We notice a systematic reduction of the condition number as a result of the reduction procedure (CNR between 1.0 and 1.9) and stress the important increase in the condition number with p and in comparison with the half-plane case (ten times larger), reaching values of the order of 3×10^8 and 3×10^9 for $p = 5, 6$, respectively. Finally, we notice that the number of points in the final stencils corresponds to a stencil size reduction equal (or very close) to 50%.

5.1.7. The concave corner distribution

We now turn to the concave corner, as depicted in Fig. 18. All the points of the initial stencil are situated in the three quadrants with a uniform random distribution. We produce a set of stencils by controlling the scaling factor S to assess the convergence order for different values of p. Analysing Fig. 18 we notice that the stencil convex hull is outside the domain and highlights potential reconstruction difficulties if the data are quite different from one side to the other.

Table 13
Approximation errors and convergence orders for the concave corner distribution case with SSO for $p = 2$. Approximations for the first- and second-order derivatives are considered for different scales (S) leading to an average converge rate denoted ACO (for $S = 1 \to 1/16$). The values of parameters h_M and h_W are also presented.

S		1	1/2	1/4	1/8	1/16	1/32	1/64	ACO
∂_x	Error	1.76e−02	4.18e−03	1.02e−03	2.52e−04	6.26e−05	1.56e−05	3.89e−06	
	Order	–	2.1	2.0	2.0	2.0	2.0	2.0	2.0
∂_y	Error	8.90e−02	2.23e−02	5.61e−03	1.41e−03	3.52e−04	8.82e−05	2.21e−05	
	Order	–	2.0	2.0	2.0	2.0	2.0	2.0	2.0
∂_{xx}	Error	4.85e−02	2.62e−02	1.37e−02	7.04e−03	3.57e−03	1.80e−03	9.01e−04	
	Order	–	0.9	0.9	1.0	1.0	1.0	1.0	0.9
∂_{yy}	Error	1.79e−01	1.27e−01	7.34e−02	3.92e−02	2.03e−02	1.03e−02	5.19e−03	
	Order	–	0.5	0.8	0.9	0.9	1.0	1.0	0.8
$h_M = h_W$		2.90e−01	1.45e−01	7.24e−02	3.62e−02	1.81e−02	9.05e−03	4.53e−03	

Table 14
Condition number of matrix MWM^t and stencil excess size ratio (SESR) for the concave corner distribution case with SSO, before and at the end of the SSO procedure, for different p values. The values of the condition number and SESR are independent of the scale value S.

p	Condition number			SESR (%)	
	Initial	Final	CNR	Initial	Final
2	9.8e+01	6.5e+01	1.5	100	17
3	8.5e+02	7.5e+02	1.1	100	0
4	8.4e+03	5.7e+03	1.5	100	7
5	6.5e+04	6.8e+04	1.0	100	5
6	3.9e+06	4.5e+06	0.9	100	4

Table 15
Approximation errors and convergence orders for the random distribution case considering different optimisation procedures and $p = 2$. Approximations for the first- and second-order derivatives with respect to x are considered for different scales (S) leading to an average converge rate denoted ACO (for $S = 1 \to 1/16$). The values of parameters h_M and h_W are also presented.

Optimisation	S		1	1/2	1/4	1/8	1/16	1/32	1/64	ACO
stencil size opt.	∂_x	Error	1.05e−01	2.50e−02	6.17e−03	1.54e−03	3.83e−04	9.58e−05	2.39e−05	
		Order	–	2.1	2.0	2.0	2.0	2.0	2.0	2.0
	∂_{xx}	Error	6.29e−02	1.45e−02	3.01e−03	4.43e−04	4.62e−04	9.03e−05	6.20e−05	
		Order	–	2.1	2.3	2.8	−0.1	2.4	0.5	1.8
cloud opt.	∂_x	Error	5.43e−02	1.36e−02	3.41e−03	8.56e−04	2.14e−04	5.37e−05	1.34e−05	
		Order	–	2.0	2.0	2.0	2.0	2.0	2.0	2.0
	∂_{xx}	Error	1.23e−01	7.44e−02	4.07e−02	2.13e−02	1.09e−02	5.51e−03	2.77e−03	
		Order	–	0.7	0.9	0.9	1.0	1.0	1.0	0.9
fminsearch opt.	∂_x	Error	5.60e−02	1.40e−02	3.53e−03	8.86e−04	2.22e−04	5.56e−05	1.39e−05	
		Order	–	2.0	2.0	2.0	2.0	2.0	2.0	2.0
	∂_{xx}	Error	1.57e−01	9.23e−02	5.00e−02	2.60e−02	1.33e−02	6.71e−03	3.37e−03	
		Order	–	0.8	0.9	0.9	1.0	1.0	1.0	0.9

We present in Tables 13 ($p = 2$) and 54–57 ($p = 3, \dots, 6$) the errors and convergences orders for the first- and second-order derivatives at the origin after the stencil reduction. Optimal orders are effectively achieved for all the reconstruction orders considered, exception made to the smaller values of the scaling factor when $p = 5, 6$. At that stage, the roughness of the boundary does not alter the quality of the derivatives' calculation as long as the solution is smooth enough. Nevertheless, concave edges or corners are well-known to produce singularities with blow-up solutions at the origin. In that case, the singularity has to be identified and subtracted to the solution to provide a smooth residual term we can reconstruct. Table 14 gives the condition numbers before and after reducing the stencil size. The SESR value is generally under 8% while the conditioning in maintained, for each value of p, in a rather narrow interval.

5.2. Numerical benchmarks (SPO)

We now proceed with the full optimisation of the condition number regarding the three parameters. We shall compare the configuration obtained for the random, half-plane and convex corner distributions using the SSO scheme with the ones obtained when the cloud-like and the `fminsearch` optimisation procedures are used. We address, in particular, the accuracy and convergence order for the approximation of the first- and second-order derivatives with respect to x provided by each method

Table 16
Condition number of matrix MWM^{t}, condition number ratio (CNR) and scaling parameters (h_M, h_W) for the random distribution case for different optimisation procedures and scale values (S) when $p = 2$.

Optimisation	S	1	1/2	1/4	1/8	1/16	1/32	1/64
stencil size opt.	h_M	3.22e−01	1.61e−01	8.04e−02	4.02e−02	2.01e−02	1.01e−02	5.03e−03
	h_W	3.22e−01	1.61e−01	8.04e−02	4.02e−02	2.01e−02	1.01e−02	5.03e−03
	cond. nb.	1.2e+01						
	CNR	1.4						
cloud opt.	h_M	1.72e−01	8.60e−02	4.30e−02	2.15e−02	1.07e−02	5.37e−03	2.69e−03
	h_W	2.56e−01	1.28e−01	6.40e−02	3.20e−02	1.60e−02	8.00e−03	4.00e−03
	cond. nb.	6.5e+00						
	CNR	2.6						
fminsearch opt.	h_M	2.03e−01	1.01e−01	5.07e−02	2.53e−02	1.27e−02	6.33e−03	3.16e−03
	h_W	2.88e−01	1.44e−01	7.21e−02	3.60e−02	1.80e−02	9.01e−03	4.50e−03
	cond. nb.	7.1e+00						
	CNR	2.4						

Table 17
Approximation error and convergence order for the half-plane distribution case considering different optimisation procedures and $p = 2$. Approximation for the first- and second-order derivatives with respect to x are considered for different scales (S) leading to an average converge order denoted ACO (for $S = 1 \to 1/16$).

Optimisation	S		1	1/2	1/4	1/8	1/16	1/32	1/64	ACO
stencil size opt.	∂_x	Error	6.55e−02	1.55e−02	3.77e−03	9.30e−04	2.31e−04	5.75e−05	1.44e−05	
		Order	–	2.1	2.0	2.0	2.0	2.0	2.0	2.0
	∂_{xx}	Error	6.32e−02	3.10e−02	1.53e−02	7.59e−03	3.78e−03	1.89e−03	9.43e−04	
		Order	–	1.0	1.0	1.0	1.0	1.0	1.0	1.0
cloud opt.	∂_x	Error	2.61e−02	6.16e−03	1.50e−03	3.71e−04	9.21e−05	2.29e−05	5.73e−06	
		Order	–	2.1	2.0	2.0	2.0	2.0	2.0	2.0
	∂_{xx}	Error	1.41e−01	6.29e−02	2.97e−02	1.45e−02	7.13e−03	3.54e−03	1.76e−03	
		Order	–	1.2	1.1	1.0	1.0	1.0	1.0	1.1
fminsearch opt.	∂_x	Error	2.11e−02	5.04e−03	1.24e−03	3.06e−04	7.62e−05	1.90e−05	4.74e−06	
		Order	–	2.1	2.0	2.0	2.0	2.0	2.0	2.0
	∂_{xx}	Error	3.26e−01	1.45e−02	6.88e−02	3.35e−02	1.65e−02	8.21e−03	4.09e−03	
		Order	–	1.2	1.1	1.0	1.0	1.0	1.0	1.0

as well as the condition number of MWM^{t}. Again, the different domains used to access the convergence behaviour are obtained recurring to a scaling of the simulation domain $[-1,1]\times[-1,1]$, while the function to approximate at $(x,y) = (0,0)$ is e^{x+2y}.

5.2.1. Random distribution

Tables 15 ($p = 2$) and 58–64 ($p = 3,\dots,6$) present the approximation errors and convergence orders obtained for the random distribution with the three optimisation schemes for different combinations of p and S. We observe that optimal convergence order is obtained for all optimisation techniques, except for the $p = 6$ case when $S = 1/32$ and, specially, for $S = 1/64$. We also note that the SSO provides the largest approximation error independently of the values of p and S considered. In other words, the two SPO approaches implemented lead to an improvement in the accuracy of the approximation regarding the SSO.

Fig. 19 presents the final stencils obtained using the `fminsearch` scheme for the different values of p. We observe a rather uniform distribution of the final stencil points as was the case for the SSO counterpart. Tables 16 ($p = 2$) and 59–65 ($p = 3,\dots,6$) present the values of scale parameters h_M and h_W for the SSO and the two SSP schemes for different combinations of p and S, as well as the condition number of MWM^{t} and the corresponding condition number ratio. The simulation results show that the SSO leads to values of h_M, h_W and condition number that are systematically higher than the corresponding SPO values, while the cloud-like algorithm leads to the lowest condition number. In fact, the CNR values obtained for the cloud-like approach range between 2.6 and 7.1, while for the SSO algorithm these values stay below 1.5. The `fminsearch` method leads to intermediate values of the CNR.

We complete the analysis, addressing the relation between the accuracy and the condition number. To this end, given the degree of the polynomial reconstruction, p, we randomly generate a cloud of $p(p+1)$ points (twice the number of unknowns) around the origin and inside the square $[-S,S]\times[-S,S]$, with S the scaling factor we take equal to $1/8$. For each cloud of points, we perform the reconstruction using one of the four algorithms: no optimisation, stencil size optimisation (SSO), cloud optimisation and `fminsearch` optimisation (SPO). We repeat the operation ten times corresponding to ten different random stencils.

For illustration purposes, we plot in Fig. 2 the accuracy (approximation error for the first-order derivative regarding x) versus the condition number for $p = 4$ and $p = 6$. We start by noting that the condition number spans over almost one order of magnitude for $p = 2$ and nearly two orders of magnitude for $p = 6$. On the other hand, we clearly observe that reducing the condition number

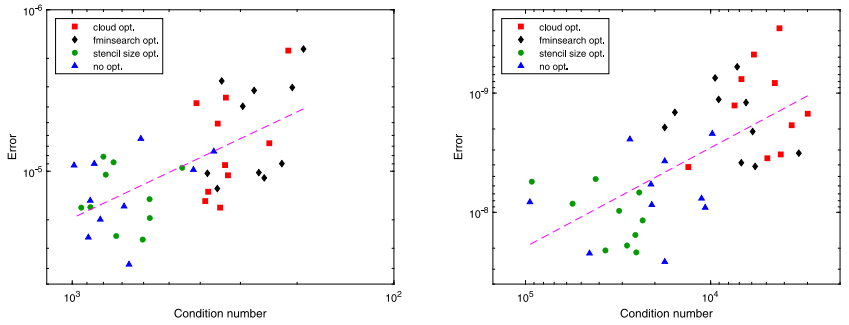

Fig. 2. Accuracy (approximation error for the first derivative with respect to x) versus condition number of MWM^{t} for the random distribution case for four different optimisation algorithms and scale factor $S = 1/8$. The left panel corresponds to $p = 4$, while the right panel stands for $p = 6$. The log–log regression line is also displayed.

greatly improves the accuracy and justifies the interest in diminishing $\chi(MWM^{\mathrm{t}})$. Notice that the points corresponding to the no optimisation and the SSO approaches belong to the left bottom side of the graphics, with high condition number and low accuracy, whereas the points corresponding to the SPO algorithms present a strong reduction of the condition number together with an increased accuracy. We draw the log–log regression line computed with the forty points, *i.e.* ten stencils with four algorithms. The slope is 0.94 for $p = 4$ and 0.83 for $p = 6$, confirming the general tendency of an improvement of the accuracy associated with a reduction of the condition number. Finally, we report that the results obtained for the first-order derivative regarding y, as well as for the second-order derivatives with respect to x and y, are not presented here since they are qualitatively similar to the ones presented for the first-order derivative regarding x.

5.2.2. The half-plane distribution

Tables 17 ($p = 2$) and 66–72 ($p = 3, \dots, 6$) present the results obtained for the approximation errors and convergence orders for the half-plane distribution, considering different values of p. We observe that optimal convergence order is obtained for all the optimisation techniques involved when $p = 2$ and 3 for the first- and second-order derivatives. When $p = 4$, the convergence order for the first-derivative drops for $S = 1/64$ for the SSO and cloud-like approaches, but with a rather low error ($< 3\times10^{-10}$). The $p = 5$ case exhibits a misbehaviour of the convergence for both derivatives when $S = 1/32$ and $1/64$. The approximation error for the first-derivative stands below 4×10^{-10} and 10^{-10} for the SSO and SPO schemes, respectively. Finally, the convergence is not optimal for $S = 1/16,\ 1/32$ and $1/64$ when $p = 6$.

Fig. 20 presents the final stencils obtained for the different values of p when the `fminsearch` scheme is used. We observe that the spatial distribution of the stencil points becomes more uniform as the value of p increases. Tables 18 ($p = 2$) and 67–73 ($p = 3, \dots, 6$) present the values of h_M and h_W obtained for the SSO and the two SSP techniques for different combinations of p and S, as well as the condition number of MWM^{t} and the corresponding CNR value. As in the random distribution case, the SSO leads to values of h_M, h_W and condition number that are systematically higher than the corresponding SPO values, while the cloud-like algorithm leads to the lowest condition number. The CNR values obtained for the cloud-like methodology range between 4.4 and 10.8, while the SSO scheme leads to values in the interval $[0.8, 1.5]$. Again, the `fminsearch` output shows intermediate CNR values in the range 3.0–8.2.

5.2.3. The convex corner distribution

The approximation errors and convergence orders obtained for the convex corner distribution considering different values of p are presented in Tables 19 ($p = 2$) and 74–80 ($p = 3, \dots, 6$). We report that optimal convergence order is obtained for all optimisation algorithms when $p = 2$ and 3 for both derivatives. When $p = 4$, the convergence order misbehaves for $S = 1/64$, independently of the optimisation scheme, and for $S = 1/32$ when `fminsearch` is used. We note, however, that the `fminsearch` strategy is the one that provides the smallest error for $S = 1/32$, while the SSO procedure leads to errors that are one and two orders of magnitude higher, respectively, for the first- and second-order derivatives. When $p = 5$, the convergence order drops below the optimal value from $S = 1/16$ on, exception made to the `fminsearch` technique that presents an erratic convergence for $S = 1/32$ and $1/64$. For $p = 6$ we report an optimal convergence order for $S = 1/8$ and above and similar approximation errors for all the three optimisation schemes being addressed. Fig. 21 presents the final stencils obtained for the different values of p when the `fminsearch` scheme is used. It shows that the final stencil points are distributed in a rather uniformly manner with respect to the reference node.

Lastly, Tables 20 ($p = 2$) and 75–81 ($p = 3, \dots, 6$) present the values of the optimisation parameters for different combinations of p and S, as well as the condition number of MWM^{t} and the corresponding condition number ratio, for the different optimisation settings. We observe that the two SPO algorithms lead to similar values of h_M and H_W, while the SSO technique leads, again, to the highest values of h_M, h_W and condition number. The results also show that the cloud-like methodology delivers the highest CNR values ($[2.1, 4.7]$) and the SSO method the lowest ($[1.0, 1.7]$). It is worth noticing that the condition number of MWM^{t} shows a considerable increase as S decreases independently of the optimisation scheme, reaching values of the order of 10^9. This behaviour of the condition number correlates with the poor convergence order observed for $S \leq 1/16$ when $p = 6$.

Table 18
Condition number of matrix MWM^{t}, condition number ratio (CNR) and scaling parameters (h_M, h_W) for the half-plane distribution case for different optimisation procedures and scale values (S) when $p = 2$.

Optimisation	S	1	1/2	1/4	1/8	1/16	1/32	1/64
stencil size opt.	h_M	3.51e−01	1.75e−01	8.77e−02	4.39e−02	2.19e−02	1.10e−02	5.48e−03
	h_W	3.51e−01	1.75e−01	8.77e−02	4.39e−02	2.19e−02	1.10e−02	5.48e−03
	cond. nb.				3.0e+02			
	CNR				1.2			
cloud opt.	h_M	1.08e−01	5.41e−02	2.71e−02	1.35e−02	6.76e−03	3.38e−03	1.69e−03
	h_W	1.50e−01	7.51e−02	3.76e−02	1.88e−02	9.39e−03	4.70e−03	2.35e−03
	cond. nb.				8.3e+01			
	CNR				4.4			
fminsearch opt.	h_M	1.09e−01	5.45e−02	2.72e−02	1.36e−02	6.81e−03	3.41e−03	1.70e−03
	h_W	1.55e−01	7.75e−02	3.88e−02	1.94e−02	9.69e−03	4.85e−03	2.42e−03
	cond. nb.				9.5e+01			
	CNR				3.9			

Table 19
Approximation error and convergence order for the convex corner distribution case considering different optimisation procedures and $p = 2$. Approximation for the first- and second-order derivatives with respect to x are considered for different scales (S) leading to an average converge rate denoted ACO (for $S = 1 \to 1/16$).

Optimisation	S		1	1/2	1/4	1/8	1/16	1/32	1/64	ACO
stencil size opt.	∂_x	Error	2.02e−01	3.39e−02	6.93e−03	1.56e−03	3.72e−04	9.06e−05	2.23e−05	
		Order	–	2.6	2.3	2.2	2.1	2.0	2.0	2.3
	∂_{xx}	Error	1.11e+00	3.62e−01	1.45e−01	6.46e−02	3.05e−02	1.48e−02	7.29e−03	
		Order	–	1.6	1.3	1.2	1.1	1.0	1.0	1.3
cloud opt.	∂_x	Error	3.34e−01	6.20e−02	1.34e−02	3.11e−03	7.51e−04	1.84e−04	4.57e−05	
		Order	–	2.4	2.2	2.1	2.1	2.0	2.0	2.2
	∂_{xx}	Error	1.28e+00	6.50e−01	2.75e−01	1.27e−01	6.07e−02	2.97e−02	1.47e−02	
		Order	–	1.0	1.2	1.1	1.1	1.0	1.0	1.1
fminsearch opt.	∂_x	Error	2.16e−01	3.80e−02	7.97e−03	1.82e−03	4.37e−04	1.07e−04	2.64e−05	
		Order	–	2.5	2.3	2.1	2.1	2.0	2.0	2.2
	∂_{xx}	Error	1.12e+00	3.79e−01	1.55e−01	6.98e−02	3.31e−02	1.61e−02	7.96e−03	
		Order	–	1.6	1.3	1.2	1.1	1.0	1.0	1.3

5.3. Cloud points preprocessing

From the computational point of view, each node in the cloud is described with a small structure of data, namely the coordinates, the label, and, for boundary points, the outward normal. The label enables to distinguish the different classes of points. For instance, a null label corresponds to an inner point, while a positive label is used to encode the boundary conditions.

A preprocessing routine has been built to compute all the structural information depending only on the geometry and the cloud, namely the stencil, and matrices W and M, the condition number and matrix A that provides the discretisation relations for $\alpha \in \mathcal{A}$. In the present study, we shall deal with five simple cases, namely, $\mathcal{A}(p)$ with $p = 2, \dots, 6$.

Given the grid of points, $\mathcal{G}$, and $\mathcal{A}$ the set of constraints, the preprocessing algorithm is the following.

Algorithm 1 Preprocessing algorithm to provide the stencils and the matrices of discretisation.

Require: grid of points $\mathcal{G}$, $\mathcal{A}$
  for each node in $\mathcal{G}$ **do**
    aggregate the $2 \times |\mathcal{A}|$ closest points to provide an initial stencil
    compute optimised stencil $\mathcal{V}$ and deduce $M(\bar{h}_M)$, $W(\bar{h}_W)$
    compute the condition number of MWM^{t}
    compute the discretisation coefficients and store them in matrix A
    save the data in file
  end for

Notice that we perform the same treatment to all the grid points, whether they belong to the boundary or not.

Table 20
Condition number of matrix MWM^t, condition number ratio (CNR) and scaling parameters (h_M, h_W) for the convex corner distribution case for different optimisation procedures and scale values (S) when $p = 2$.

Optimisation	S	1	1/2	1/4	1/8	1/16	1/32	1/64
stencil size opt.	h_M	3.35e−01	1.67e−01	8.37e−02	4.19e−02	2.09e−02	1.05e−02	5.23e−03
	h_W	3.35e−01	1.67e−01	8.37e−02	4.19e−02	2.09e−02	1.05e−02	5.23e−03
	cond. nb.				2.3e+03			
	CNR				1.7			
cloud opt.	h_M	1.95e−01	9.75e−02	4.87e−02	2.44e−02	1.22e−02	6.09e−03	3.05e−03
	h_W	3.64e−01	1.82e−01	9.11e−02	4.55e−02	2.28e−02	1.14e−02	5.69e−03
	cond. nb.				1.4e+03			
	CNR				3.0			
fminsearch opt.	h_M	2.21e−01	1.11e−01	5.53e−02	2.76e−02	1.38e−02	6.91e−03	3.46e−03
	h_W	3.96e−01	1.98e−01	9.91e−02	4.95e−02	2.48e−02	1.24e−02	6.19e−03
	cond. nb.				1.6e+03			
	CNR				2.6			

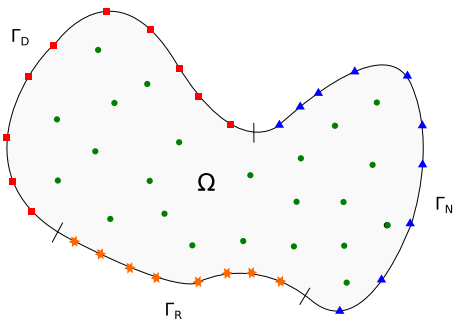

Fig. 3. Generic grid. The open bounded domain Ω and its boundary $\partial\Omega = \Gamma_D \cup \Gamma_N \cup \Gamma_R$, where Γ_D, Γ_N and Γ_R stand, respectively, for the boundary partition where Dirichlet, Neumann and Robin boundary conditions hold. The green filled circles, red squares, blue triangles and orange stars correspond to grid points belonging to Ω, Γ_D, Γ_N and Γ_R, respectively.

6. Generalised finite difference schemes

We consider the partial differential equation

$$\mathfrak{E}(\phi) = -\kappa_1\left(\partial^{(2,0)}\phi + \partial^{(0,2)}\phi\right) + \kappa_2\partial^{(1,0)}\phi + \kappa_3\partial^{(0,1)}\phi = f,$$

supplemented with the (generic) Robin boundary condition

$$\mathfrak{B}(\phi) = \gamma_D\phi + \gamma_N\left(n_x\partial^{(1,0)}\phi + n_y\partial^{(0,1)}\phi\right) = g,$$

where κ_1, κ_2, κ_3 and f are real valued functions defined on Ω, while γ_D, γ_N and g are defined on the boundary $\partial\Omega$.

6.1. Discrete formulation

At the discrete level, we substitute the derivatives by the corresponding discretisation relations, and we get for $i \in \mathring{\mathcal{G}}$

$$\left(-\kappa_1(x^i)(a^{i,(2,0)} + a^{i,(0,2)}) + \kappa_2(x^i)a^{i,(1,0)} + \kappa_3(x^i)a^{i,(0,1)}\right)\Phi_{\eta^i} = f(x^i),$$

where $a^{i,\beta}$ is the coefficient of the β derivative discretisation at node i. On the other hand, the boundary condition at the grid point $i \in \partial\mathcal{G}$ reads

$$\gamma_D(x^i)\phi^i + \gamma_N(x^i)\left(n_x(x^i)a^{i,(1,0)} + n_y(x^i)a^{i,(0,1)}\right)\Phi_{\eta^i} = g(x^i).$$

Fig. 3 illustrates a generic cloud of points used in the simulations.

To compute the weights' matrix W, we adopt the Gaussian kernel [7]

$$\omega(q, h_W) = \frac{1}{\pi h_W^2}e^{-q^2}, \qquad q = \frac{|x^i - x^j|}{h_W}.$$

This function is smooth and considered very stable and accurate [19].

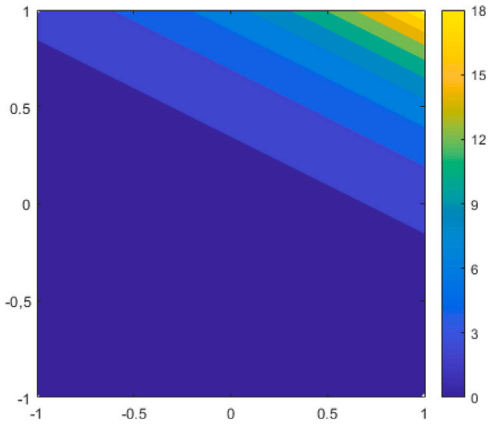

Fig. 4. Exact solution for problem (BVP1).

Assembling all the relations for $i \in \mathcal{G}$ provides the linear system $G\Phi = F$. Additionally, we use the MATLAB® function "`equilibrate`" [5], to reduce considerable the condition number of the global system (see further below). It provides a pre-conditioning linear system $\widetilde{G}\Psi = \widetilde{F}$ we solve with a direct method.

To assess the convergence, we compute the L^1- and L^∞-errors by

$$L^1\text{-error: } \frac{1}{I}\sum_{i\in\mathcal{G}} |\phi_i - \phi_i^{ex}| \quad \text{and} \quad L^\infty\text{-error: } \max_{i\in\mathcal{G}} |\phi_i - \phi_i^{ex}|,$$

where $(\phi_i^{ex})_{i\in\mathcal{G}}$ and $(\phi_i)_{i\in\mathcal{G}}$ are the exact and the approximated solution, respectively.

6.2. Numerical tests

We consider the mixed boundary condition problem

$$\begin{cases} -\left(\partial_{xx}\phi + \partial_{yy}\phi\right) + \partial_x\phi + 2\partial_y\phi = 0, \quad \text{for all } (x,y) \in \Omega, \\ \text{Dirichlet/Neumann conditions, for all } (x,y) \in \partial\Omega \end{cases} \tag{BVP1}$$

for which the exact solution is given by

$$\phi = e^{x+2y}, \text{ for all } (x,y) \in \overline{\Omega}.$$

As an illustration, we plot the solution over the domain $[0,1]\times[0,1]$ in Fig. 4.

We also tackle the problem

$$\begin{cases} -\left(\partial_{xx}\phi + \partial_{yy}\phi\right) + x\partial_x\phi + y\partial_y\phi = f, \quad \text{for all } (x,y) \in \Omega, \\ \text{Dirichlet conditions, for all } (x,y) \in \partial\Omega \end{cases} \tag{BVP2}$$

for which we impose the following exact solution

$$\phi = \cos(2\pi x)\sin(2\pi y), \text{ for all } (x,y) \in \overline{\Omega},$$

by manufacturing the second member, f, accordingly.

6.2.1. Test case 1

Let $\Omega =]-1,1[\times]-1,1[$ and $\Gamma_D = \partial\Omega$. We seek a numerical solution of BVP1 with Dirichlet boundary conditions corresponding to set

$$\phi = e^{x+2y}, \text{ for all } (x,y) \in \Gamma_D.$$

Numerical simulations have been carried out with four clouds of points using uniform grids (GU-1) with 117, 437, 1677, and 7565 points. A second set of clouds (GP15-1) is obtained by performing an up to 15% random perturbation on the previous clouds GU-1. At last, another set of clouds (GP30-1) is obtained as an up to 30% random perturbation of clouds GU-1. On the other hand, a set of clouds (GD-1) is obtained with a Delaunay-like triangulation procedure with 209, 815, 3193, and 12 378 points, using the Gmsh© package. Fig. 5 illustrates the different sets of clouds used in the simulations.

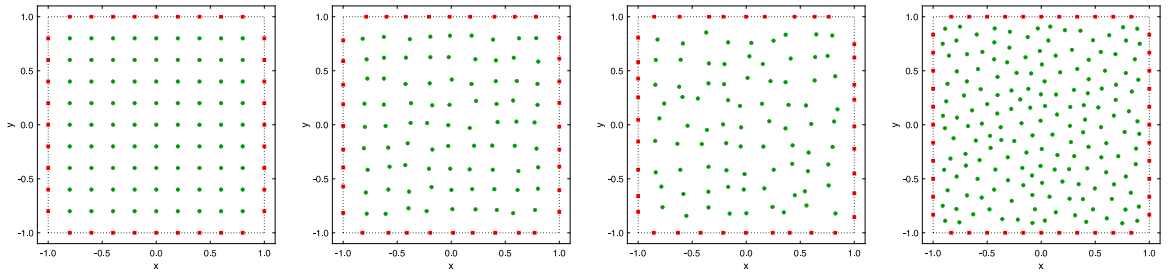

Fig. 5. Grids for the test case 1. For $\overline{\Omega} = \{(x, y) \in \mathbb{R}^2 : x, y \in [-1, 1]\}$ the panels illustrate, from left to right, a uniform grid, two grids with perturbation levels up to 15% and 30% with respect to the uniform case, and a grid obtained with a Delaunay-like triangulation procedure. The green filled circles and the red squares correspond, respectively, to interior points and boundary points where Dirichlet conditions are imposed.

GU-1 tests.

We carry out the simulations using the GU-1 clouds with four optimisation scenarios: no optimisation, stencil size optimisation, cloud-like optimisation and `fminsearch` optimisation. Errors and convergence orders, as well as for the condition number of G and $\widetilde{G}$ are presented in Tables 21 (for $p = 2$) and 82–85 (for $p = 3, \ldots, 6$, respectively).

For $p = 2$ and 3, we report that sticking to the initial stencil, convergence orders are far from optimal, while the condition number of G can reach values as high as 2×10^{10}. On the other hand, for $p = 4,\ 5$ and 6, the convergence order of this scheme is close to optimal, but the L^∞- and L^1-errors are considerably higher than the ones obtained with the three other optimisation procedures. The SSO approach also leads to poor convergence orders for the $p = 2$ and 3 cases, while for $p = 4,\ 5$ and 6, we observe convergence orders that are in line with the (optimal) ones obtained with the cloud and `fminsearch` approaches, but the errors and the condition number are well above the values obtained using its SPO counterparts.

Comparing the results obtained with the cloud-like and the `fminsearch` methods, we observe that the errors for both the numerical solution and the first-order derivatives are generally lower for the `fminsearch` approach. On the other hand, the condition number of G (and $\widetilde{G}$) shows a weak dependency on the optimisation method used. We note that, in both cases, the use of the "equilibrate" procedure allows reducing the condition number of the system matrix up to three orders of magnitude for the finest grid.

Given, on the one hand, the poor quality of the results obtained with the initial stencil approach and (to a less extent) the SSO method for the accuracy and convergence order of the numerical solution when compared with the SPO methods and, on the other hand, the fact that the two SPO algorithms lead to similar results for the numerical solution and first-order derivatives, being the `fminsearch` method much faster than the cloud-like algorithm (see further below), the numerical simulations for the remaining three sets of non-uniform grids are performed exclusively with the `fminsearch` method.

GU15-1, GU30-1 and Delaunay tests.

The results obtained for clouds GP15-1, GP30-1 and GD-1 (*cf.* Tables 22 for $p = 2$ and 86–89) for $p = 3, \ldots, 6$ present an optimal convergence order for both the L^∞- and the L^1-norm, while we report an average value for the ratio between the L^∞-error and the L^1-error is in the 95% confidence interval 7.3 ± 3.0, indicating a good distribution of the error throughout the computational domain. As far as the condition number of G and $\widetilde{G}$ are concerned we observe, as expected, that they increase with the number of degrees of freedom, while using the "equilibrate" formulation results in a reduction of the condition number between one and two orders of magnitude for the `fminsearch` method.

Finally, we address, on the one hand, the computation time associated to the preprocessing computations, that is, to the choice of the stencils and the computation of the corresponding local matrices, and, on the other hand, the time needed to compute the problem solution, a process that involves the assembly of the local matrices as well as the solution of the sparse system of equations. In the former case, we consider, for the set of uniform grids GU-1, the average preprocessing CPU time per degree of freedom needed for the different optimisation approaches considered in the present work for values of p ranging from 2 to 6. The results obtained, presented in Table 23 and depicted in Fig. 6, show that the logarithm of the preprocessing CPU time scales roughly in a linear manner with the approximation degree independently of the optimisation procedure involved. Additionally, and as expected, the absence of optimisation results in the fastest method, while the homemade cloud-like optimisation method is clearly the slowest. It can also be seen that the spread of preprocessing CPU times regarding the optimisation method used tends to increase with p. At last, it is worth mentioning that the optimisation based on the `fminsearch` procedure is, on average, 3 to 5 times slower than the (rather basic) SSO approach. Turning now our attention to the latter case, we consider the relation between the accuracy of the numerical solution and the CPU time needed to determine the problem solution. For that we consider once again the set of uniform grids GU-1 and the two extreme values of the approximation degree, $p = 2$ and $p = 6$, for illustration purposes. The results obtained are presented in Tables 24 and 25 and depicted in Fig. 7. The results show that for a given CPU time the accuracy provided by the cloud-like and `fminsearch` optimisation methods is rather similar and up to two orders of magnitude better than the no optimisation procedure, while up to one order of magnitude better than the accuracy provided by the SSO technique for the finer grids.

Table 21
Results for the test case 1 considering four uniform grids and $p = 2$. The error and convergence order for the numerical solution, as well as the condition number of the system matrix (G) and its regularised counterpart ($\tilde{G}$), are presented for different optimisation procedures. For the cloud and the fminsearch optimisation procedures the error and convergence order for the first-order derivatives are also presented.

Optimisation	DOF		117	437	1677	7565	ACO
no opt.	Solution	L^∞-error	7.40e−01	3.02e−01	7.78e−02	2.66e−01	
		Order	–	1.4	2.0	−1.6	0.5
		L^1-error	1.38e−01	6.01e−02	1.24e−02	1.01e−02	
		Order	–	1.3	2.3	0.3	1.3
	cond. nb. G		1.3e04	4.7e05	1.7e07	2.1e10	
	cond. nb. $\tilde{G}$		3.3e02	2.8e03	2.6e04	6.8e06	
stencil size opt.	Solution	L^∞-error	1.57e−01	5.04e−02	2.26e−02	8.89e−03	
		Order	–	1.7	1.2	1.2	1.4
		L^1-error	2.40e−02	1.04e−02	5.69e−03	2.29e−03	
		Order	–	1.3	0.9	1.2	1.1
	cond. nb. G		1.9e02	1.0e03	5.0e03	3.5e04	
	cond. nb. $\tilde{G}$		1.3e01	5.1e01	2.0e02	9.5e02	
cloud opt.	Solution	L^∞-error	1.16e−01	1.62e−02	3.38e−03	1.15e−03	
		Order	–	3.0	2.3	1.4	2.2
		L^1-error	1.25e−02	9.85e−04	5.62e−04	1.98e−04	
		Order	–	3.9	0.8	1.4	2.1
	∂_x	L^∞-error	8.85e−01	2.45e−01	8.10e−02	2.13e−02	
		Order	–	1.9	1.6	1.8	1.8
		L^1-error	8.80e−02	2.17e−02	5.16e−03	1.21e−03	
		Order	–	2.1	2.1	1.9	2.1
	∂_y	L^∞-error	1.33e00	5.93e−01	1.99e−01	5.15e−02	
		Order	–	1.2	1.5	1.8	1.6
		L^1-error	1.80e−01	4.56e−02	1.07e−02	2.27e−03	
		Order	–	2.1	2.2	2.1	2.1
	cond. nb. G		2.2e02	1.1e03	6.2e03	4.2e04	
	cond. nb. $\tilde{G}$		1.9e01	7.4e01	2.9e02	1.3e03	
fminsearch opt.	Solution	L^∞-error	5.93e−02	9.34e−03	2.11e−03	4.39e−04	
		Order	–	2.8	2.2	2.1	2.4
		L^1-error	1.10e−02	2.60e−03	6.26e−04	1.33e−04	
		Order	–	2.2	2.1	2.1	2.1
	∂_x	L^∞-error	6.94e−01	2.52e−01	7.48e−02	1.75e−02	
		Order	–	1.5	1.8	1.9	1.8
		L^1-error	5.93e−02	1.58e−02	4.02e−03	8.77e−04	
		Order	–	2.0	2.0	2.0	2.0
	∂_y	L^∞-error	9.99e−01	3.62e−01	8.79e−02	2.62e−02	
		Order	–	1.5	2.1	1.6	1.7
		L^1-error	1.28e−01	3.34e−02	8.52e−03	1.86e−03	
		Order	–	2.0	2.0	2.0	2.0
	cond. nb. G		2.0e02	1.1e03	6.1e03	4.1e04	
	cond. nb. $\tilde{G}$		1.9e01	7.5e01	3.0e02	1.4e03	

6.2.2. Test case 2

The computational domain is an annulus having internal and external radius $1/3$ and 1, respectively, that is

$$\Omega = \left\{(x,y) = (\rho\cos\theta, \rho\sin\theta) \in \mathbb{R}^2 \,:\, \rho \in]1/3, 1[,\ \theta \in [0, 2\pi[\right\},$$

where the inner and outer boundaries are given by

$$\Gamma_D = \left\{(x,y) = (\cos\theta, \sin\theta) \in \mathbb{R}^2 \,:\, \theta \in [0, 2\pi[\right\}, \quad \Gamma_N = \left\{(x,y) = (\cos\theta, \sin\theta)/3 \in \mathbb{R}^2 \,:\, \theta \in [0, 2\pi[\right\},$$

We seek the numerical solution of BVP1 with the Dirichlet and Neumann boundary conditions corresponding to set

$$\phi = e^{x+2y}, \text{ for all } (x,y) \in \Gamma_D, \quad n_x\partial_x\phi + n_y\partial_y\phi = e^{x+2y}\left(n_x + 2n_y\right), \text{ for all } (x,y) \in \Gamma_N.$$

The geometrical setting and the exact solution of the BVP, $\phi = e^{x+2y}$, are illustrated in Fig. 8.

To assess the numerical accuracy, we use a set of four uniform clouds (U-2) with 190, 740, 2938, and 11 257 points. A second set of four clouds (P15-2) is obtained by performing an up to 15% random perturbation of a uniform grid, while a set of four grids (P30-2) with an up to 30% random perturbation is also used. The number of points of the four successive clouds (P15-2) (and also (P30-2)) are 168, 711, 2860, and 11 442 points. At last, a set of four grids (GD-2) with 228, 880, 3369, and 11 888 points is obtained thanks to a Delaunay-like triangulation procedure. Fig. 9 illustrates the different sets of grids used in the test case 2. All the simulations are carried out with the SPO `fminsearch` procedure.

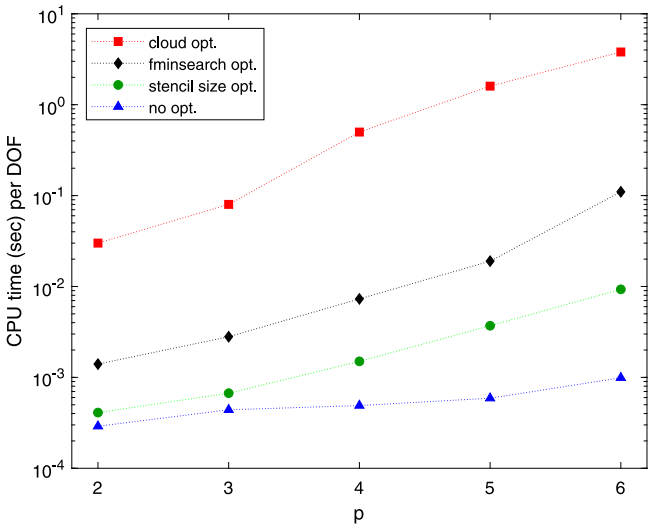

Fig. 6. Test case 1. Average preprocessing time (in CPU seconds) per degree of freedom as a function of p and optimisation procedure (see Table 23).

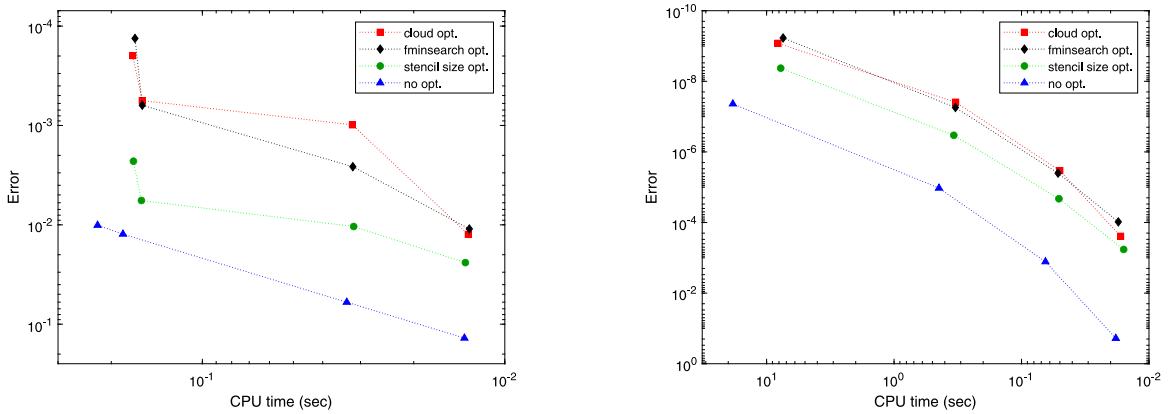

Fig. 7. Test case 1. Relation between the numerical solution accuracy (L^1-error) and the computation time (in CPU seconds) for different optimisation procedures for four uniform grids with 117, 437, 1677 and 7565 points. The left panel stands for $p = 2$ while the right panel corresponds to $p = 6$ (see Tables 24 and 25, respectively).

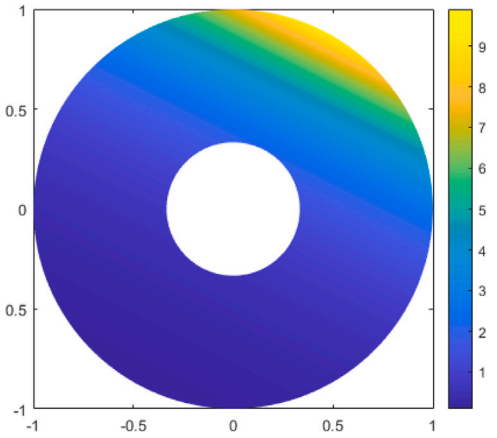

Fig. 8. Test case 2. Geometrical setting and exact solution.

Table 22
Results for the test case 1 for the fminsearch optimisation procedure considering nonuniform grids and $p = 2$. The error and convergence order of the numerical solution, as well as the condition number of the system matrix (G) and its regularised counterpart ($\widetilde{G}$), are presented for three sets of nonuniform grids: two sets of grids with perturbation levels up to 15% and 30% with respect to the uniform case, and a set of grids obtained with a Delaunay triangulation procedure.

Grid type	DOF		117	437	1677	7565	ACO
<15% perturbation	Solution	L^∞-error	1.15e−01	1.87e−02	2.11e−03	7.22e−04	
		Order	–	3.2	1.5	2.1	2.4
		L^1-error	1.21e−02	2.63e−03	6.26e−04	1.76e−04	
		Order	–	2.2	1.6	2.1	2.0
	cond. nb. G		2.3e02	1.3e03	6.1e03	5.4e04	
	cond. nb. $\tilde{G}$		2.1e01	8.0e01	3.0e02	1.6e03	
<30% perturbation	Solution	L^∞-error	6.33e−02	2.15e−02	2.11e−03	9.80e−04	
		Order	–	2.4	2.0	2.1	2.0
		L^1-error	1.14e−02	4.10e−03	6.26e−04	1.85e−04	
		Order	–	2.0	2.3	2.1	2.0
	cond. nb. G		2.7e02	1.9e03	6.1e03	7.7e04	
	cond. nb. $\tilde{G}$		2.2e01	9.0e01	3.0e02	1.9e03	
	DOF		209	815	3193	12 378	ACO
Delaunay	Solution	L^∞-error	6.93e−02	5.48e−03	1.30e−03	3.35e−04	
		Order	–	3.7	2.1	2.0	2.6
		L^1-error	5.61e−03	1.41e−03	3.89e−04	9.84e−05	
		Order	–	2.0	1.9	2.0	2.0
	cond. nb. G		7.8e02	4.9e03	2.8e04	1.8e05	
	cond. nb. $\tilde{G}$		4.5e01	1.9e02	7.9e02	3.1e03	

Table 23
Results for the average preprocessing time (in CPU seconds) per degree of freedom as a function of p and optimisation procedure for the test case 1 considering four uniform grids with 117, 437, 1677 and 7565 points.

Optimisation	p				
	2	3	4	5	6
no opt.	(2.9 ± 1.0)e−04	(4.4 ± 2.1)e−04	(4.9 ± 1.4)e−04	(5.9 ± 1.6)e−04	(9.9 ± 2.5)e−04
stencil size opt.	(4.1 ± 1.4)e−04	(6.7 ± 1.6)e−04	(1.5 ± 0.2)e−03	(3.7 ± 0.2)e−03	(9.3 ± 0.3)e−03
fminsearch opt.	(1.4 ± 0.5)e−03	(2.8 ± 0.5)e−03	(7.3 ± 0.5)e−03	(1.9 ± 0.1)e−02	(1.1 ± 0.2)e−01
cloud opt.	(3.0 ± 0.1)e−02	(8.0 ± 0.7)e−02	(5.0 ± 0.2)e−01	(1.6 ± 0.1)e−00	(3.8 ± 0.1)e−00

Table 24
Results for the computation time (in CPU seconds) and solution L^1-error as a function of the optimisation procedure for the test case 1 considering $p = 2$ and four uniform grids with 117, 437, 1677 and 7565 points.

Optimisation	DOF	117	437	1677	7565
no opt.	L^1-error	1.38e−01	6.01e−02	1.24e−02	1.01e−02
	CPU time	1.36e−02	3.33e−02	1.83e−01	2.22e−01
stencil size opt.	L^1-error	2.40e−02	1.04e−02	5.69e−03	2.29e−03
	CPU time	1.35e−02	3.16e−02	1.59e−01	1.69e−01
cloud opt.	L^1-error	1.25e−02	9.85e−04	5.62e−04	1.98e−04
	CPU time	1.32e−02	3.19e−02	1.58e−01	1.70e−01
fminsearch opt.	L^1-error	1.10e−02	2.60e−03	6.26e−04	1.33e−04
	CPU time	1.31e−02	3.18e−02	1.58e−01	1.67e−01

Errors and convergence orders, as well as for the condition number of G and $\widetilde{G}$, are reported in Tables 26 ($p = 2$) and 90–93 ($p = 3, \ldots, 6$). The optimal convergence is achieved for all values of p independently of the shape of the clouds. The average value for the ratio between the L^∞-error and the L^1-error is higher than in the previous test case, with a 95% confidence interval 13.0 ± 3.4, reflecting the increase in the complexity of the BVP due to the curved geometry and mixed boundary conditions. The error distribution over the computational domain can still be considered good despite the moderate augmentation of this ratio.

Comparing the results obtained for each set of grids, one observes that, in general, the smallest errors correspond to the uniform and Delaunay-like grids, while the highest errors are associated with the (P30-2) set of grids. As in the previous test case, we report that the condition number of G and $\widetilde{G}$ increases with the number of degrees of freedom by up to 3 orders of magnitude. As to the effect of the "`equilibrate`" procedure, we observe that it can lead to a reduction of the condition number up to 3 orders of magnitude.

Table 25
Results for the computation time (in CPU seconds) and solution L^1-error for the test case 1. Similar to Table 24, but for $p = 6$.

Optimisation	DOF	117	437	1677	7565
no opt.	L^1-error	1.91e−01	1.29e−03	1.06e−05	4.36e−08
	CPU time	1.82e−02	6.49e−02	4.44e−01	1.84e+01
stencil size opt.	L^1-error	5.79e−04	2.12e−05	3.38e−07	4.25e−09
	CPU time	1.58e−02	5.09e−02	3.40e−01	7.75e+00
cloud opt.	L^1-error	2.45e−04	3.38e−06	3.90e−08	8.38e−10
	CPU time	1.67e−02	5.01e−02	3.29e−01	8.15e+00
fminsearch opt.	L^1-error	9.59e−05	4.00e−06	5.53e−08	5.92e−10
	CPU time	1.74e−02	5.17e−02	3.30e−01	7.42e+00

Table 26
Results for the test case 2 for the fminsearch optimisation procedure and $p = 2$. The error and convergence order of the numerical solution, as well as the condition number of the system matrix (G) and its regularised counterpart ($\widetilde{G}$), are presented for four sets of grid types: a set of uniform grid types, two sets of grid types with perturbation levels up to 15% and 30% with respect to the uniform case, and a set of grid types obtained with a Delaunay triangulation procedure.

Grid type	DOF		190	740	2938	11 257	ACO
Uniform	Solution	L^∞-error	1.95e−01	3.36e−02	5.32e−03	3.03e−03	
		Order	–	2.6	2.7	0.8	2.0
		L^1-error	1.54e−02	2.17e−03	3.77e−04	9.28e−05	
		Order	–	2.9	2.5	2.1	2.5
	cond. nb. G		1.1e03	3.6e03	5.6e04	2.5e05	
	cond. nb. $\tilde{G}$		7.3e01	3.3e02	2.2e03	3.5e04	
	DOF		168	711	2860	11 442	ACO
<15% perturbation	Solution	L^∞-error	5.56e−01	9.05e−02	1.62e−02	5.20e−03	
		Order	–	2.5	2.5	1.6	2.2
		L^1-error	4.02e−02	6.71e−03	1.58e−03	2.17e−04	
		Order	–	2.5	2.1	2.9	2.5
	cond. nb. G		1.6e03	7.8e04	3.2e05	6.5e05	
	cond. nb. $\tilde{G}$		9.9e01	4.9e03	8.4e03	1.2e04	
<30% perturbation	Solution	L^∞-error	7.56e−01	1.31e−01	2.84e−02	6.77e−03	
		Order	–	2.4	2.2	2.1	2.2
		L^1-error	2.37e−02	5.26e−03	1.13e−03	2.58e−04	
		Order	–	2.1	2.2	2.1	2.1
	cond. nb. G		1.8e03	6.9e04	1.2e06	1.7e06	
	cond. nb. $\tilde{G}$		7.8e01	3.6e03	3.5e04	2.0e04	
	DOF		228	880	3369	11 888	ACO
Delaunay	Solution	L^∞-error	1.12e−01	8.45e−03	2.80e−03	5.95e−04	
		Order	–	3.8	1.6	2.5	2.6
		L^1-error	8.76e−03	1.13e−03	3.14e−04	8.74e−05	
		Order	–	3.1	1.9	2.0	2.3
	cond. nb. G		2.7e03	8.0e03	9.7e04	2.0e05	
	cond. nb. $\tilde{G}$		1.4e02	3.5e02	1.4e02	5.0e03	

Table 27
Results for the test case 3 for the fminsearch optimisation procedure and $p = 2$. The error and convergence order of the numerical solution, as well as the condition number of the system matrix (G) and its regularised counterpart ($\widetilde{G}$), are presented for four grids obtained with a Delaunay triangulation technique combined with a level-set procedure.

DOF		217	847	3365	13 696	ACO
Solution	L^∞-error	2.35e−01	9.41e−02	1.74e−02	5.31e−03	
	Order	–	1.3	2.4	1.7	1.8
	L^1-error	5.04e−02	1.85e−02	4.58e−03	1.07e−03	
	Order	–	1.5	2.0	2.1	1.9
cond. nb. G		1.1e03	5.7e03	4.3e04	2.4e05	
cond. nb. $\tilde{G}$		3.4e01	1.4e02	6.6e02	2.8e03	

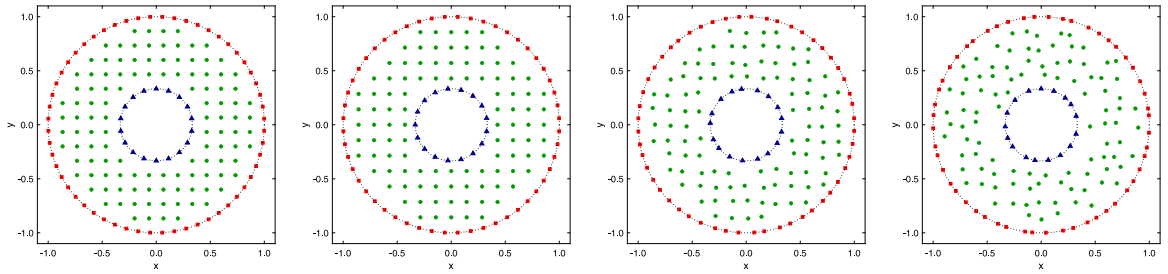

Fig. 9. Grids for the test case 2. For $\overline{\Omega} = \{(x, y) = (\rho\cos\theta, \rho\sin\theta) \in \mathbb{R}^2 : \rho \in [1/3, 1],\ \theta \in [0, 2\pi[\}$ the panels illustrate, from left to right, a uniform grid, two grids with perturbation levels up to 15% and 30% with respect to the uniform case, and a grid derived from a Delaunay-like triangulation procedure. The green filled circles, the red squares and the blue triangles correspond, respectively, to interior points, and boundary points where Dirichlet and Neumann conditions are imposed.

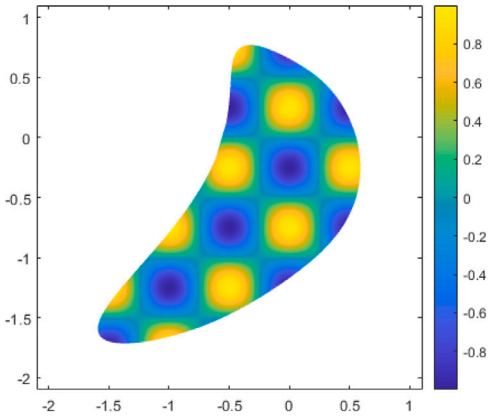

Fig. 10. Test case 3. Geometrical setting and exact solution.

6.2.3. Test case 3

We look at the numerical solution of BVP2 over a complex computational domain to show that one can address numerical approximation with a curved boundary domain given by the level-set

$$\Omega = \left\{ (x, y) \in \mathbb{R}^2 : \sqrt{x^4 + \frac{y^2}{16}} + \left(\frac{3}{2}x + y^2\right)^2 + y < 1 \right\}.$$

Dirichlet condition is prescribed on the boundary

$$\Gamma_D = \left\{ (x, y) \in \mathbb{R}^2 : \sqrt{x^4 + \frac{y^2}{16}} + \left(\frac{3}{2}x + y^2\right)^2 + y = 1 \right\},$$

using the exact solution $\phi = \cos(2\pi x)\sin(2\pi y)$. The geometrical setting and the exact solution of the BVP are illustrated in Fig. 10.

We carried out the simulations with use four grids (GD-3) of 217, 847, 3365, and 13 656 points, respectively. In all cases, the interior points are obtained with a Delaunay-like triangulation procedure, while the points on the boundary are set using a level-set technique. Fig. 11 illustrates the coarsest grids used in the test case 3. As in the test case 2, the simulations are performed using the SPO `fminsearch` procedure.

Errors, convergence orders, and condition numbers of G and $\widetilde{G}$, are depicted in Tables 27 ($p = 2$) and 94–97 for $p = 3, \ldots, 6$. The errors are definitively higher in the last case than in the two previous tests independently of p. This is due to the increased complexity of both the domain shape and the exact solution of the BVP. Nevertheless, optimal convergence is still achieved for all values of p despite the increased complexity of the BVP. We also notice that the average value for the ratio between the L^∞-error and the L^1-error is considerably lower than in the previous tests, with a 95% confidence interval 5.0 ± 0.7, reflecting a very good error distribution over the computational domain, particularly for $p = 2, \ldots, 5$. Similarly to the previous cases, the condition number of G and $\widetilde{G}$ increases with the number of degrees of freedom by up to 3 orders of magnitude. We also report a reduction of the condition number ranging from 1 to 3 orders of magnitude in the framework of the "`equilibrate`" procedure.

7. Conclusions

In the framework of Generalised Finite Difference Methods, we proposed an optimisation procedure to achieve the best stencil (stencil size optimisation) and coefficients h_M and h_W (scaling parameters optimisation) that minimise the condition number of

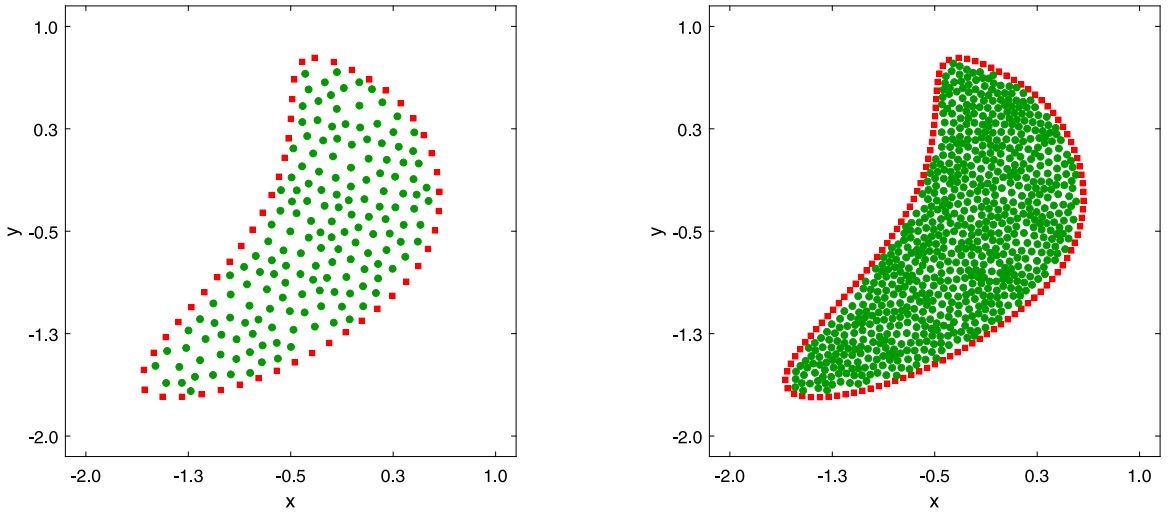

Fig. 11. Grids for the test case 3. For $\overline{\Omega} = \{(x, y) \in \mathbb{R}^2 : \sqrt{x^4 + y^2/16} + \left(3x/2 + y^2\right)^2 + y \leq 1\}$ the panels illustrate two grids, derived from a Delaunay-like triangulation procedure, with 217 (left) and 847 (right) points. The green filled circles and the red squares correspond, respectively, to interior points and boundary points where Dirichlet conditions are imposed.

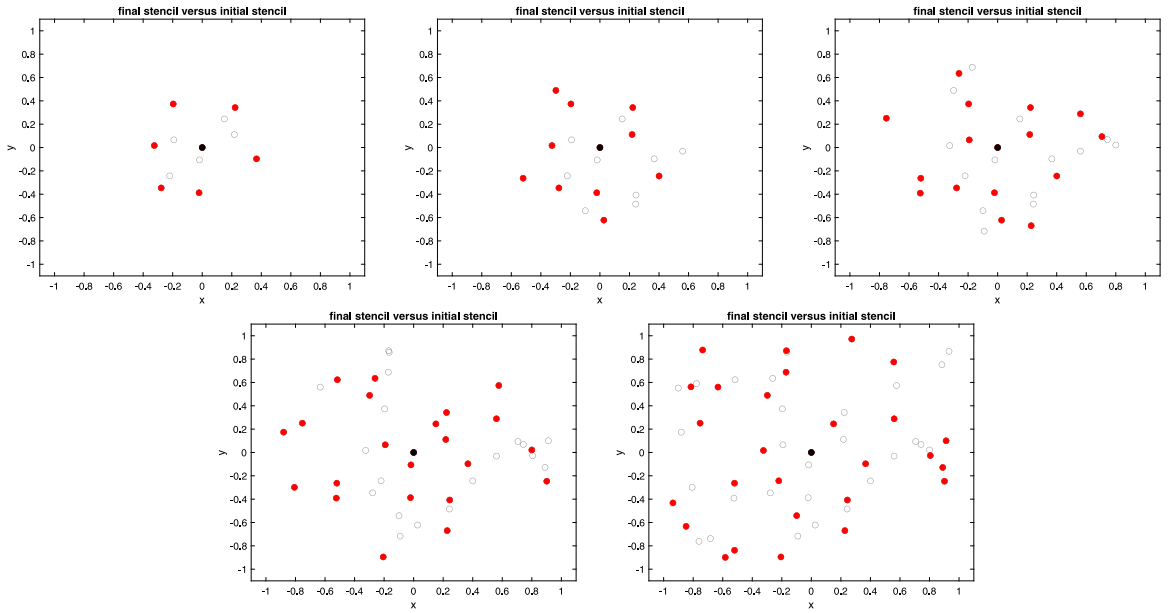

Fig. 12. Stencil choice for the random distribution case with SSO for $p = 2$ (top left panel), $p = 3$ (top middle panel), $p = 4$ (top right panel), $p = 5$ (bottom left panel), $p = 6$ (bottom right panel). The black filled circle corresponds to the reference node, the red filled circles stand for the remaining nodes of the final stencil, while the black circles correspond to the nodes discarded during the optimisation procedure. The figures correspond to $S = 1$ and the relative position of points does not depend on the scale value. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

matrix MWM^{t}. In respect to the stencil, we propose a discrete greedy optimisation procedure by eliminating, one by one, the non-necessary nodes in order to reduce the stencil size while preserving or reducing the condition number. For the continuous optimisation we recurred to two derivative-free optimisation procedures, the MATLAB® function `fminsearch` and a homemade cloud-like optimisation routine, which delivered very similar results for the values of h_M and h_W and reduction of the condition number. Since the `fminsearch` procedure is much faster, we adopted it as the optimisation tool used in the benchmarks presented.

We performed an extensive series of numerical tests to assess the condition number reduction and the accuracy of the first- and second-order derivatives discretisation for different cloud geometries. We found that the use of the full optimisation procedure leads to an optimal convergence with respect to the cloud scaling independently of the geometric scenario considered.

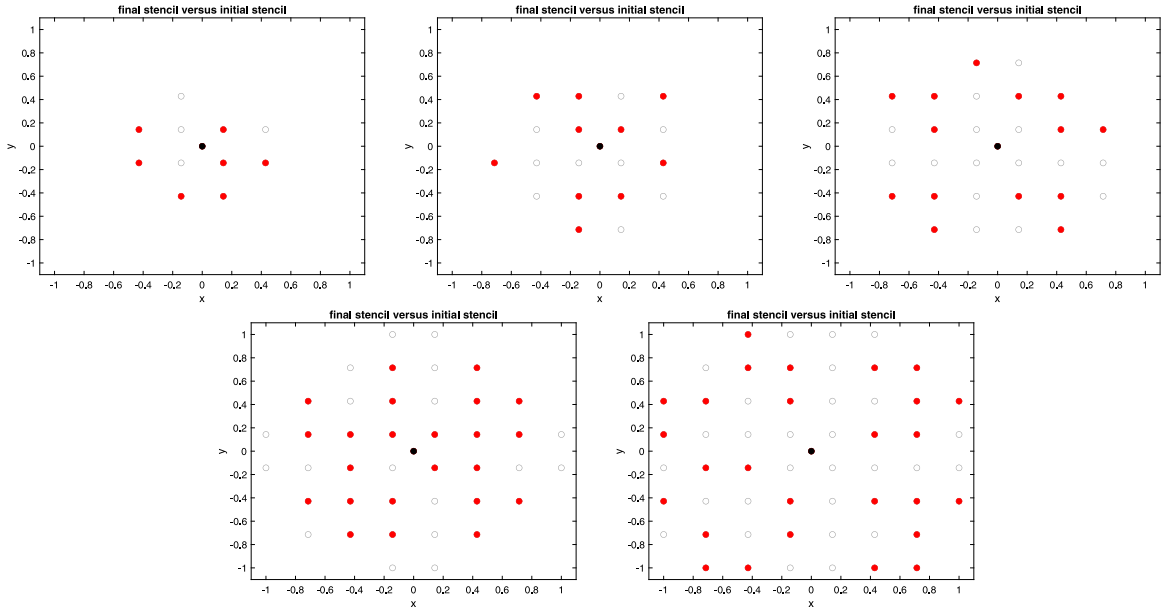

Fig. 13. Stencil choice for the uniform distribution case with SSO for $p = 2$ (top left panel), $p = 3$ (top middle panel), $p = 4$ (top right panel), $p = 5$ (bottom left panel), $p = 6$ (bottom right panel). The black filled circle corresponds to the reference node, the red filled circles stand for the remaining nodes of the final stencil, while the black circles correspond to the nodes discarded during the optimisation procedure. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

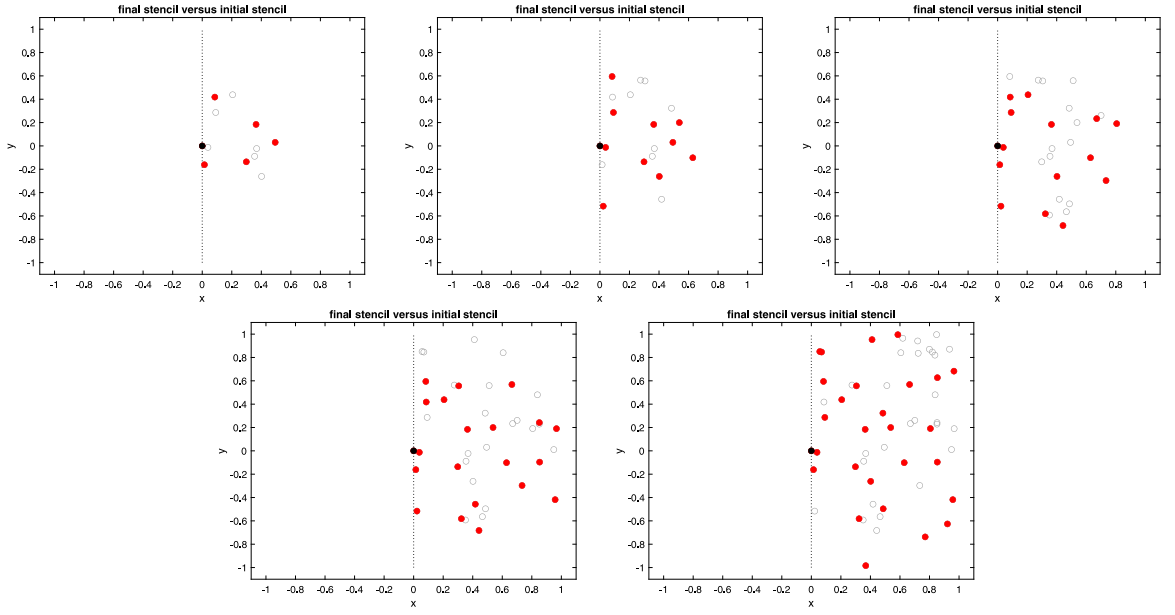

Fig. 14. Stencil choice for the half-plane distribution case with SSO for $p = 2$ (top left panel), $p = 3$ (top middle panel), $p = 4$ (top right panel), $p = 5$ (bottom left panel), $p = 6$ (bottom right panel). The black filled circle corresponds to the reference node, the red filled circles stand for the remaining nodes of the final stencil, while the black circles correspond to the nodes discarded during the optimisation procedure. The figures correspond to $S = 1$ and the relative position of points does not depend on the scale value. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Finally, we performed a set of simulations for three test cases involving the convection diffusion reaction equation supplemented by Dirichlet/Neumann boundary conditions for different geometrical settings and grid types. Using the derivatives computed with the optimal reconstruction, proposed in the present work, we obtained effective convergence rates for the numerical approximation

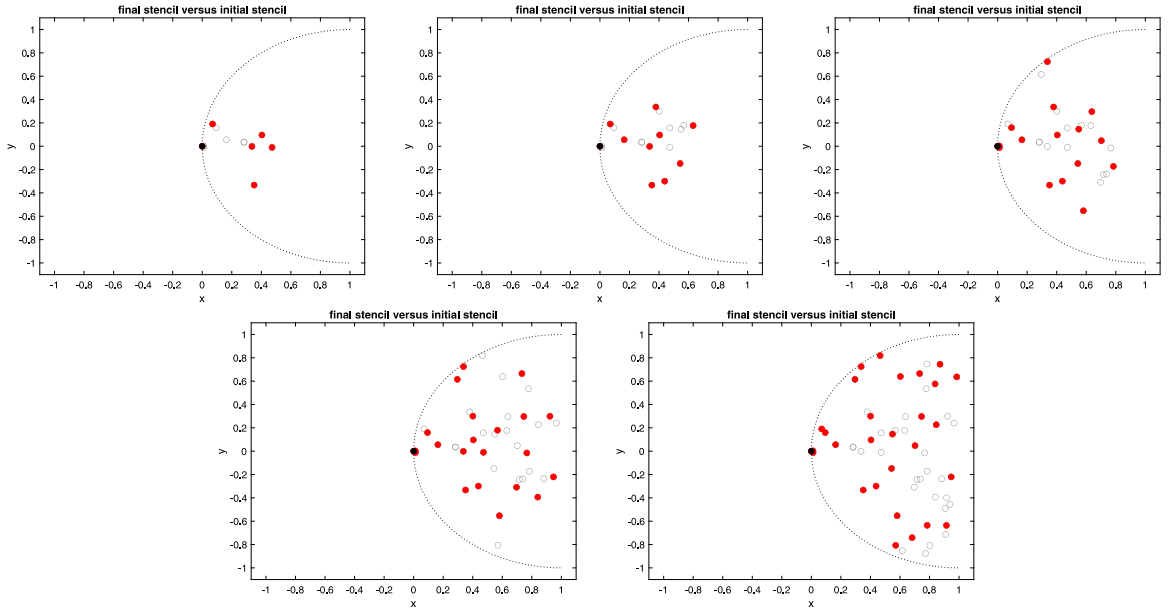

Fig. 15. Stencil choice for the regular convex distribution case with SSO for $p = 2$ (top left panel), $p = 3$ (top middle panel), $p = 4$ (top right panel), $p = 5$ (bottom left panel), $p = 6$ (bottom right panel). The black filled circle corresponds to the reference node, the red filled circles stand for the remaining nodes of the final stencil, while the black circles correspond to the nodes discarded during the optimisation procedure. The figures correspond to $S = 1$ and the relative position of points does not depend on the scale value. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

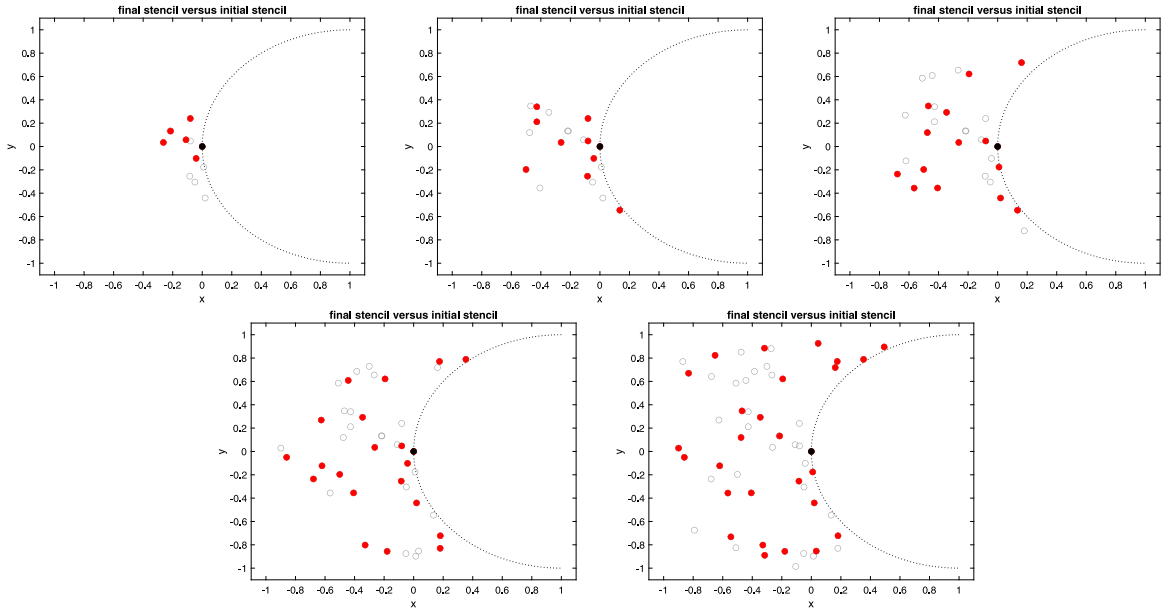

Fig. 16. Stencil choice for the regular concave distribution case with SSO for $p = 2$ (top left panel), $p = 3$ (top middle panel), $p = 4$ (top right panel), $p = 5$ (bottom left panel), $p = 6$ (bottom right panel). The black filled circle corresponds to the reference node, the red filled circles stand for the remaining nodes of the final stencil, while the black circles correspond to the nodes discarded during the optimisation procedure. The figures correspond to $S = 1$ and the relative position of points does not depend on the scale value. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

of the solution up to 6th order. We also showed that an appropriate pre-conditioning of the system matrix can lead to a dramatic reduction of the condition number up to three orders of magnitude, contributing to enhancing the accuracy of the numerical solution.

Extension to three-dimensional geometries will be further considered where one almost uses the same ingredients: stencil, discrete derivatives that match partial derivatives for a set of polynomial functions. Condition number is expected to be a harder issue for 3D

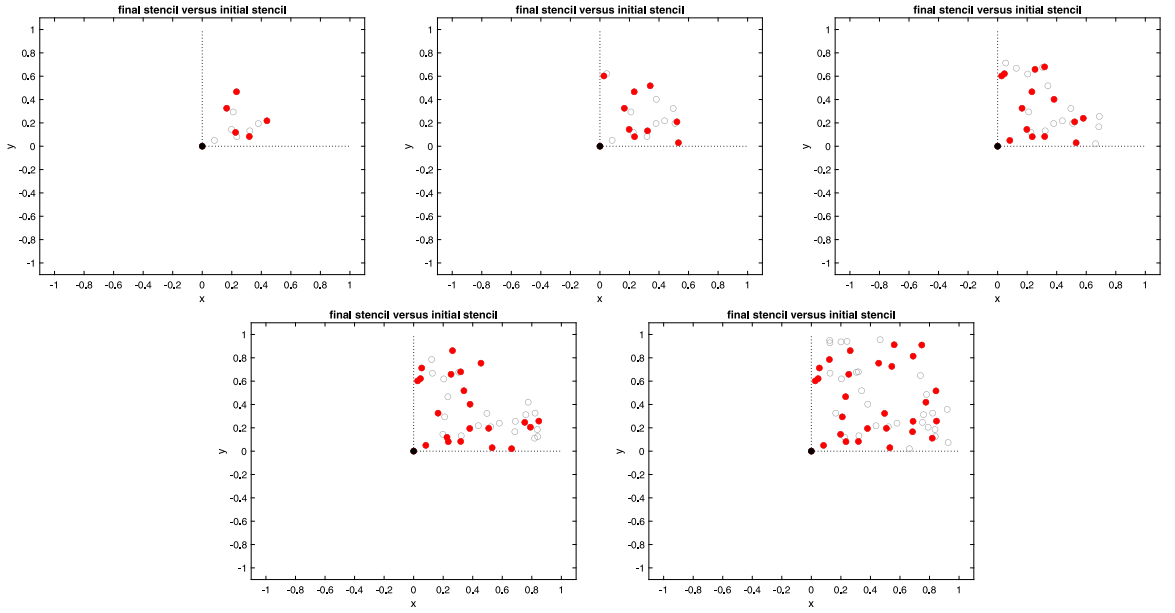

Fig. 17. Stencil choice for the convex corner distribution case with SSO for $p = 2$ (top left panel), $p = 3$ (top middle panel), $p = 4$ (top right panel), $p = 5$ (bottom left panel), $p = 6$ (bottom right panel). The black filled circle corresponds to the reference node, the red filled circles stand for the remaining nodes of the final stencil, while the black circles correspond to the nodes discarded during the optimisation procedure. The figures correspond to $S = 1$ and the relative position of points does not depend on the scale value. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

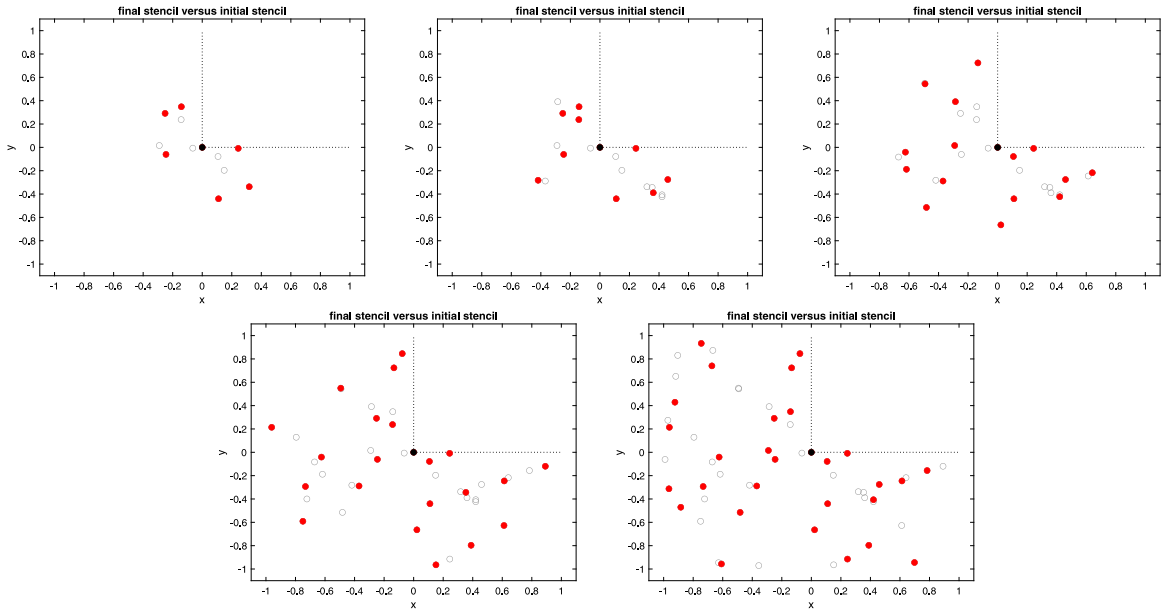

Fig. 18. Stencil choice for the concave corner distribution case with SSO for $p = 2$ (top left panel), $p = 3$ (top middle panel), $p = 4$ (top right panel), $p = 5$ (bottom left panel), $p = 6$ (bottom right panel). The black filled circle corresponds to the reference node, the red filled circles stand for the remaining nodes of the final stencil, while the black circles correspond to the nodes discarded during the optimisation procedure. The figures correspond to $S = 1$ and the relative position of points does not depend on the scale value. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

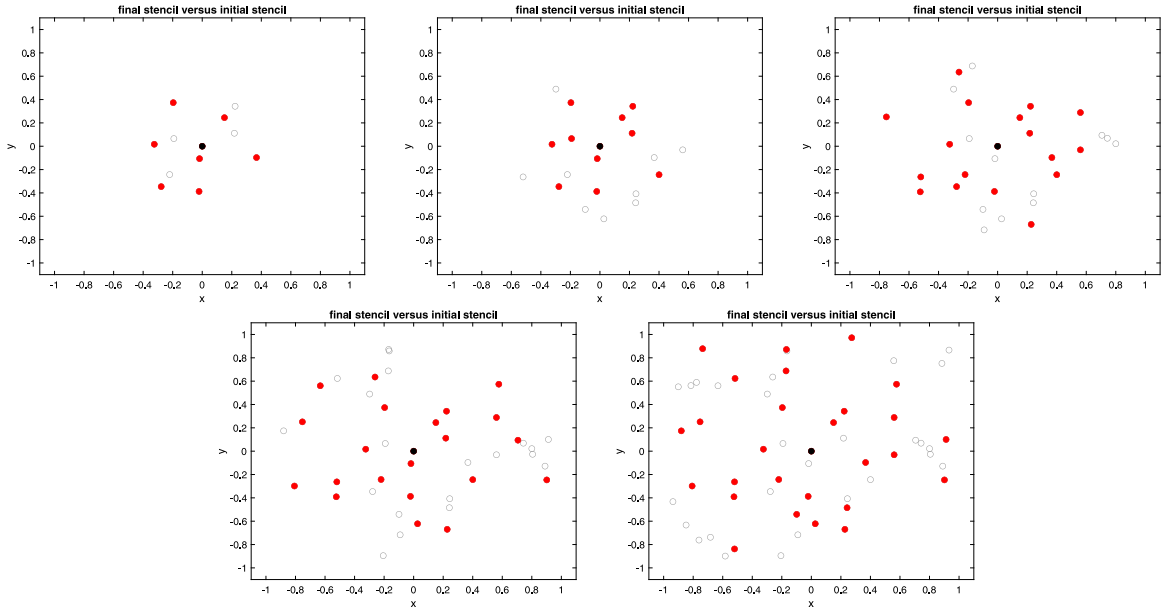

Fig. 19. Stencil choice for the random distribution case with stencil size and scaling parameters optimisation (using the fminsearch procedure) for $p = 2$ (top left panel), $p = 3$ (top middle panel), $p = 4$ (top right panel), $p = 5$ (bottom left panel), $p = 6$ (bottom right panel). The black filled circle corresponds to the reference node, the red filled circles stand for the remaining nodes of the final stencil, while the black circles correspond to the nodes discarded during the optimisation procedure. The figures correspond to $S = 1$ and the relative position of points does not depend on the scale value. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

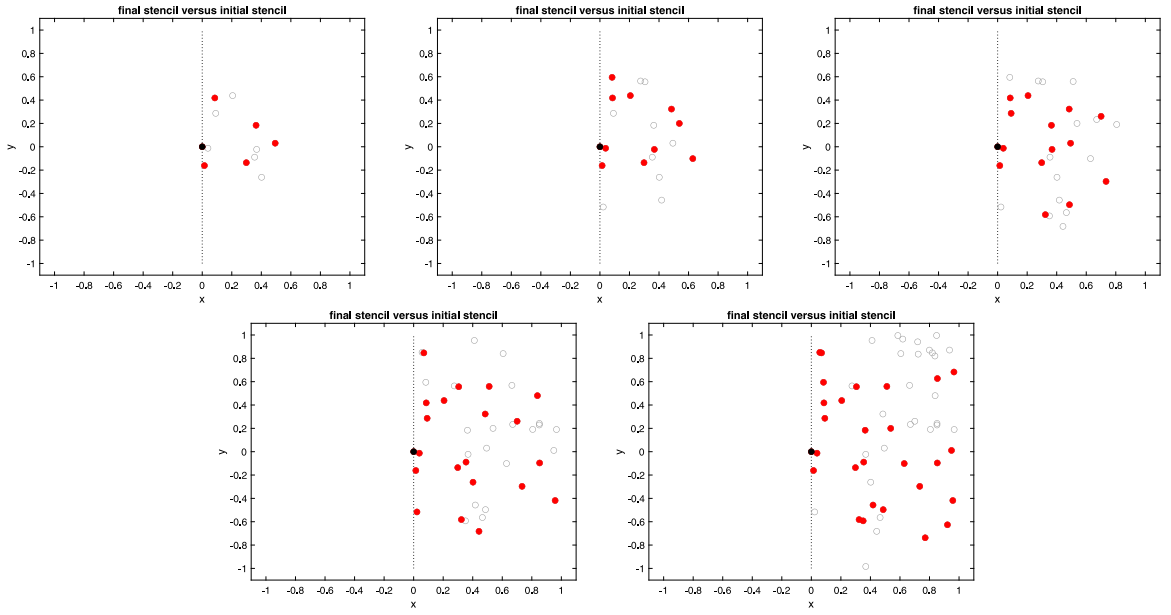

Fig. 20. Stencil choice for the half-plane distribution case with stencil size and scaling parameters optimisation (using the fminsearch procedure) for $p = 2$ (top left panel), $p = 3$ (top middle panel), $p = 4$ (top right panel), $p = 5$ (bottom left panel), $p = 6$ (bottom right panel). The black filled circle corresponds to the reference node, the red filled circles stand for the remaining nodes of the final stencil, while the black circles correspond to the nodes discarded during the optimisation procedure. The figures correspond to $S = 1$ and the relative position of points does not depend on the scale value. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

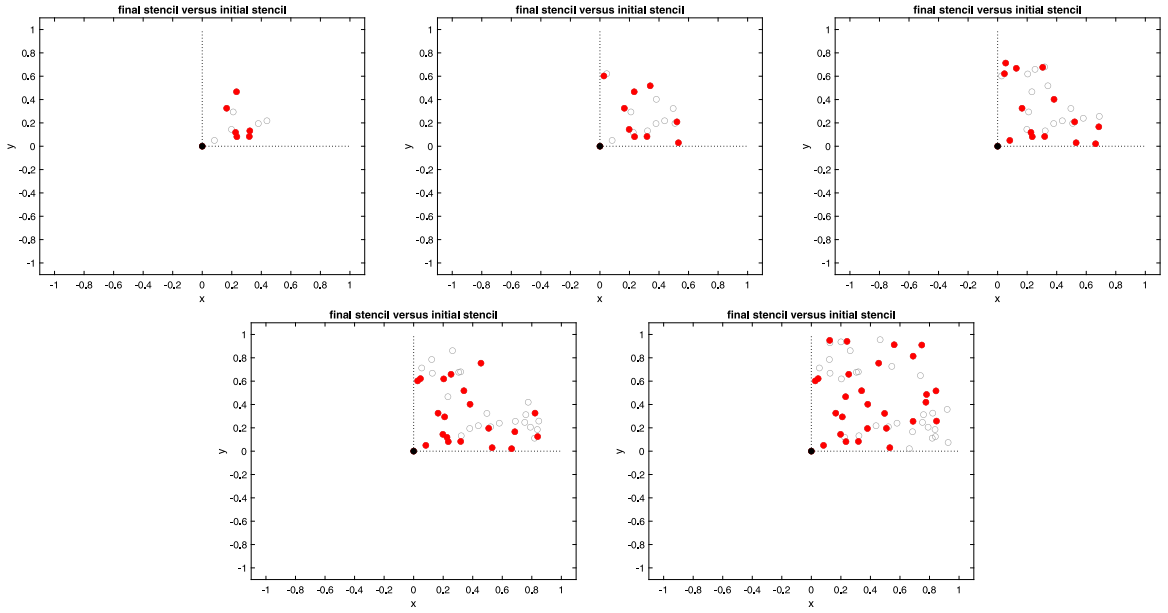

Fig. 21. Stencil choice for the convex corner distribution case with stencil size and scaling parameters optimisation (using the fminsearch procedure) for $p = 2$ (top left panel), $p = 3$ (top middle panel), $p = 4$ (top right panel), $p = 5$ (bottom left panel), $p = 6$ (bottom right panel). The black filled circle corresponds to the reference node, the red filled circles stand for the remaining nodes of the final stencil, while the black circles correspond to the nodes discarded during the optimisation procedure. The figures correspond to $S = 1$ and the relative position of points does not depend on the scale value. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

problems, hence the conditioning minimisation is expected to turn out to be a very significant contribution to preserve the expected accuracy.

Acknowledgements

J. Figueiredo acknowledges the financial support by FEDER – Fundo Europeu de Desenvolvimento Regional, through COMPETE 2020 – Programa Operational Fatores de Competitividade through FCT – Fundação para a Ciência e a Tecnologia, project N° UID/FIS/04650/2019.

S. Clain the financial support by FEDER – Fundo Europeu de Desenvolvimento Regional, through COMPETE 2020 – Programa Operational Fatores de Competitividade through FCT – Fundação para a Ciência e a Tecnologia, project N° UIDB/00324/2020.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.matcom.2023.11.009.

References

[1] J.J. Benito, F. Ureña, L. Gavete, Influence of several factors in the generalized finite difference method, Appl. Math. Model. 25 (2001) 1039–1053.
[2] J. Benito, F. Ureña, L. Gavete, Solving parabolic and hyperbolic equations by the generalized finite difference method, J. Comput. Appl. Math. 209 (2007) 208–233.
[3] B. Boroomand, M. Najjar, E. Oñate, The generalized finite point method, Comput. Mech. 44 (2009) 173–190.
[4] B. Boroomand, A. Tabatabaei, E. Oñate, Simple modifications for stabilization of the finite point method, Internat. J. Numer. Methods Engrg. 63 (2005) 351–379.
[5] I.S. Duff, J. Koster, On algorithms for permuting large entries to the diagonal of a sparse matrix, SIAM J. Matrix Anal. Appl. 22 (4) (2001) 973–996.
[6] L. Gavete, F. Ureña, J. Benito, A. Garcia, M. Ureña, E. Salete, Solving second order non-linear elliptic partial differential equations using generalized finite difference method, J. Comput. Appl. Math. 318 (2017) 378–387.
[7] R.A. Gingold, J.J. Monaghan, Smoothed particle hydrodynamics – theory and application to non-spherical stars, Mon. Not. R. Astron. Soc. 181 (1977) 375–389.
[8] A. Idesman, Optimal local truncation error method for solution of partial differential equations on irregular domains and interfaces using unfitted Cartesian meshes, Arch. Comput. Methods Eng. 30 (2023) 4517–4564.
[9] T. Jacquemin, S. Tomar, K. Agathos, S.M.-M. ans S.P.A. Bordas, Taylor-series expansion based numerical methods: a primer, performance benchmarking and new approaches for problems with non-smooth solutions, Arch. Comput. Methods Eng. 27 (7) (2020) 1465–1513.
[10] I. Jaworska, On the ill-conditioning in the new higher order multipoint method, Comput. Math. Appl. 66 (3) (2013) 238–249.
[11] I. Jaworska, Generalization of the Multipoint meshless FDM application to the nonlinear analysis, Comput. Math. Appl. 87 (2021) 1–11.
[12] I. Jaworska, J. Orkisz, Higher order multipoint method – from Collatz to meshless FDM, Eng. Anal. Bound. Elem. 50 (2015) 341–351.

[13] I. Jaworska, J. Orkisz, On nonlinear analysis by the multipoint meshless FDM, Eng. Anal. Bound. Elem. 92 (2018) 231–243.
[14] P. Jensen, Finite difference technique for variable grids, Comput. Struct. 2 (1972) 17–29.
[15] J. Kuhnert, S. Tiwari, Grid Free Method for Solving the Poisson Equation, Bericht 25, Fraunhofer Institut für Techno- und Wirtschaftsmathematik, 2001.
[16] J. Kuhnert, S. Tiwari, Finite Pointset Method Based on the Projection Method for Simulations of the Incompressible Navier-Stokes Equations, Bericht 30, Fraunhofer Institut für Techno- und Wirtschaftsmathematik, 2001.
[17] J.C. Lagarias, J.A. Reeds, M.H. Wright, P.E. Wright, Convergence properties of the melder-mead simplex method in low dimensions, SIAM J. Optim. 9 (1) (1998) 112–147.
[18] T. Liszka, J. Orkisz, The finite difference method at arbitrary irregular grids and its application in applied mechanics, Comput. Struct. 11 (1980) 83–95.
[19] M.B. Liu, G.R. Liu, Smoothed particle hydrodynamics (SPH): an overview and recent developments, Arch. Comput. Methods Eng. 17 (2010) 25–76.
[20] R. Löhner, C. Sacco, E. Oñate, S. Idelsohn, A finite point method for compressible flow, Internat. J. Numer. Methods Engrg. 53 (2002) 1765–1779.
[21] E. Oñate, S. Idelsohn, A mesh-free finite point method for advective-diffusive transport and fluid flow problems, Comput. Mech. 21 (1998) 283–292.
[22] E. Oñate, S. Idelsohn, O. Zienkiewicz, R. Taylor, A finite point method in computational mechanics—applications to convective transport and fluid flow, Internat. J. Numer. Methods Engrg. 139 (1996) 3839–3866.
[23] E. Oñate, C. Sacco, S. Idelsohn, A finite point method for incompressible flow problems, Comput. Vis. Sci. 2 (2000) 67–75.
[24] N. Perrone, R. Kao, A general finite difference method for arbitrary meshes, Comput. Struct. 5 (1975) 45–58.
[25] E. Reséndiz-Flores, I. García-Calvillo, Application of the finite pointset method to non-stationary heat conduction problems, Int. J. Heat Mass Transfer 71 (2014) 720–723.
[26] E. Reséndiz-Flores, F.R. Saucedo-Zendejo, Two-dimensional numerical simulation of heat transfer with moving heat source in welding using the Finite Pointset Method, Int. J. Heat Mass Transfer 90 (2015) 239–245.
[27] F. Saucedo-Zendejo, E. Reséndiz-Flores, A new approach for the numerical simulation of free surface incompressible flows using a mesh-free method, Comput. Methods Appl. Mech. Engrg. 324 (2017) 619–639.
[28] P. Suchde, J. Kuhnert, S. Tiwari, On meshfree GFDM solvers for the incompressible Navier–Stokes equations, Comput. & Fluids 165 (2018) 1–12.
[29] G. Tinoco-Guerrero, F. Domínguez-Mota, J. Tinoco-Ruiz, A study of the stability for a generalized finite-difference scheme applied to the advection–diffusion equation, Math. Comput. Simulation 176 (2020) 301–311.
[30] F. Ureña, J. Benito, L. Gavete, Application of the generalized finite difference method to solve the advection-diffusion equation, J. Comput. Appl. Math. 235 (2011) 1849–1855.
[31] F. Ureña, L. Gavete, A. García, J.J. Benito, A.M. Vargas, Solving second order non-linear parabolic PDEs using generalized finite difference method (GFDM), J. Comput. Appl. Math. 354 (2019) 221–241.
[32] X. Xing, L. Song, X. He, C. Qiu, A generalized finite difference method for solving elliptic interface problems, Math. Comput. Simulation 178 (2020) 109–124.
[33] J. Zhanga, R.-J. Shuya, C.-L. Chu, C.-M. Fan, Generalized finite difference method for three-dimensional eigenproblems of Helmholtz equation, Math. Comput. Simulation 196 (2022) 45–67.