Received: Added at production | Revised: Added at production | Accepted: Added at production
DOI: xxx/xxxx

ARTICLE TYPE

CMMSE: A technique for generating adapted discretizations to solve partial differential equations with the generalized finite difference method

A. C. Albuquerque-Ferreira*[1] | Miguel Ureña[2] | Higinio Ramos[3]

[1]Department of Computer Engineering, Universidad de Salamanca, Salamanca, Spain
[2]Statistics Spain (INE), Madrid, Spain
[3]Department of Applied Mathematics, Universidad de Salamanca, Salamanca, Spain

Correspondence
*Augusto César Albuquerque Ferreira, Department of Computer Engineering, Universidad de Salamanca, 37008 Salamanca, Spain. Email: augustocesar@usal.es

Summary

The generalized finite difference method is a meshless method for solving partial differential equations that allows arbitrary discretizations of points. Typically, the discretizations have the same density of points in the domain. We propose a technique to get adapted discretizations for the solution of partial differential equations. This strategy allows using a smaller number of points and a lower computational cost to achieve the same accuracy that would be obtained with a regular discretization.

KEYWORDS:
generalized finite difference method, adapted discretization, fourth-order approximations.

1 | INTRODUCTION

Meshless methods appeared from 1970 onwards in numerical simulations of astrophysics problems through the well-known smooth particle hydrodynamics (SPH) method[1]. Since then, many researchers have endeavored to increase the accuracy and the computational performance of such methods. Among the meshless methods, there are those used in the strong form, such as the finite point method[2], the regularized local collocation method[3] or the generalized finite difference method (GFDM).

An important step in the development of the GFDM was the introduction of weighted moving least squares in the derivative approximations[4,5]. Liszka and Orkisz (1980)[5] developed a robust algorithm for the solution of physical and geometrical nonlinearities. The influence of key parameters of the method was studied by Benito (2001)[6]. In the last years, the development of researches involving higher-order approximations has gained strength[7,8,9,10,11]. The different applications of the GFDM in current engineering problems show the versatility of this method[12,13,14,15,16,17,18,19,20].

The discretization of the domain when applying GFDM to solve a partial differential equation has been addressed in many ways. It can be discretized regularly whenever possible and irregularly only in regions where it cannot be done otherwise. In the latter case, there is usually some minimum distance criterion to prevent two points from being too close together.

It can also be discretized irregularly, either arbitrarily[21,22,23] or on the basis of some kind of structure such as, for example, using triangular elements (Delaunay triangulation)[24,7,25,26], quadrilateral elements[27,28], partitions into nodal subdomains (Voronoi tessellation)[29,7] or Coatmèlec distribution of points[30].

All these discretizations have in common that the initial distribution of points is of approximately constant density throughout the domain. Liszka and Orkisz (1980)[5] developed a pre-process based on density functions that are defined by the user. However, there are not many papers where initial distributions adapted to the problem are applied, i.e., with a distribution of points that allows capturing the particularities of the problem or part of them. For example, in[31] a higher density of points near the interfaces is used, and in[26] a higher density of points near the boundary is used.

Ph.D. Miguel Ureña contributed to this article in his personal capacity. The views expressed are his own and do not necessarily represent the views of Statistics Spain.

There are many papers where discretizations with different densities are used in different regions of the domain, but these discretizations are the result of adaptive algorithms that refine the discretization in several steps depending on where the highest errors are.

This paper aims at designing a strategy to generate a discretization adapted to the problem in a general way. To do this, we consider two stages, a first one that uses a coarse regular discretization to calculate the gradients and distribute the points according to the gradient values, and a second one to solve the equation in the new discretization.

2 | THE GENERALIZED FINITE DIFFERENCE METHOD

Assuming an interior bounded domain $\Omega \subset \mathbb{R}^2$ with boundary Γ, and enough regularity of the known functions $f : \Omega \to \mathbb{R}$ and $g : \Gamma \to \mathbb{R}$, we want to find a function $U(x, y)$ satisfying the Dirichlet problem

$$\mathcal{L}\big(U(x, y)\big) = f(x, y),\ (x, y) \in \Omega \tag{1}$$

$$U(x, y) = g(x, y),\ (x, y) \in \Gamma, \tag{2}$$

where $\mathcal{L}$ is a second-order linear differential operator with constant coefficients.

Let us consider a discretization M of $D = \Omega \cup \Gamma$. We assume that for each interior point $(x_0, y_0) \in M \cap \Omega$ there is an associated star $V = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\} \subset M$, with m points.

For each point of V, we consider the Taylor expansion of $U(x, y)$ around (x_0, y_0) up to fourth-order terms. Denoting by u_i the approximate value of $U(x_i, y_i)$ after truncation, we have the residual vector given by

$$\mathbf{r} = \Delta\mathbf{u} - \mathbf{P}\mathbf{D}_\mathbf{u}, \tag{3}$$

where

$$\mathbf{P} = \begin{bmatrix} h_1 & k_1 & \frac{h_1^2}{2} & \frac{k_1^2}{2} & h_1k_1 & \cdots & \frac{h_1k_1^3}{6} & \frac{k_1^4}{24} \\ h_2 & k_2 & \frac{h_2^2}{2} & \frac{k_2^2}{2} & h_2k_2 & \cdots & \frac{h_2k_2^3}{6} & \frac{k_2^4}{24} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ h_i & k_i & \frac{h_i^2}{2} & \frac{k_i^2}{2} & h_ik_i & \cdots & \frac{h_ik_i^3}{6} & \frac{k_i^4}{24} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ h_m & k_m & \frac{h_m^2}{2} & \frac{k_m^2}{2} & h_mk_m & \cdots & \frac{h_mk_m^3}{6} & \frac{k_m^4}{24} \end{bmatrix}. \tag{4}$$

The values $h_i = x_i - x_0$ and $k_i = y_i - y_0$, $i = 1, \ldots, m$, are the relative distances from the central point to the point (x_i, y_i) (see Figure 1), $\mathbf{D}_\mathbf{u}$ is the approximate value of the derivatives at the central point, that is,

$$\mathbf{D}_\mathbf{u} = \left(\frac{\partial u_0}{\partial x}, \frac{\partial u_0}{\partial y}, \frac{\partial^2 u_0}{\partial x^2}, \frac{\partial^2 u_0}{\partial y^2}, \frac{\partial^2 u_0}{\partial x \partial y}, \ldots, \frac{\partial^4 u_0}{\partial x \partial y^3}, \frac{\partial^4 u_0}{\partial y^4}\right)^T, \tag{5}$$

and $\Delta\mathbf{u} = (u_1 - u_0, u_2 - u_0, \ldots, u_m - u_0)^T$.

The minimization of the weighted residuals $\mathbf{r}^T\mathbf{W}^2\mathbf{r}$ (where $\mathbf{W} = \mathrm{diag}(\omega_1, \omega_2, \ldots, \omega_m)$ is a diagonal matrix of weights) by the least squares method produces the following equation

$$\frac{\partial(\mathbf{r}^T\mathbf{W}^2\mathbf{r})}{\partial\mathbf{D}_\mathbf{u}} = -2\mathbf{P}^T\,\mathbf{W}^2\Delta\mathbf{u} + 2\mathbf{P}^T\,\mathbf{W}^2\,\mathbf{P}\,\mathbf{Du} = 0. \tag{6}$$

Solving for $\mathbf{Du}$ in (6), we have

$$\mathbf{Du} = \mathbf{A}^{-1}\,\mathbf{P}^T\,\mathbf{W}^2\Delta\mathbf{u}, \tag{7}$$

where $\mathbf{A} = \mathbf{P}^T\mathbf{W}^2\mathbf{P}$.

To avoid problems with singularities or ill-conditioning in matrix **A**, we use the strategy in[11], where stars with different number of points and different formation criteria of the stars are considered, according to the condition number of matrix **A**.

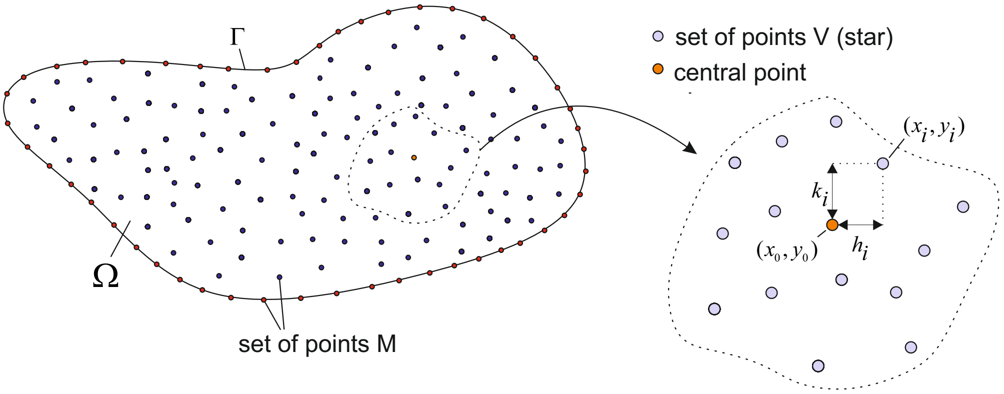


FIGURE 1 Discretization of the domain and star formation.

We apply (7) to each of the NI interior points to approximate the derivatives in (1), and use the boundary condition in (2) to get the exact values of $\mathsf{U}(x, y)$ at the points in $\mathsf{M} \cap \Gamma$. In this way, we obtain a linear system of NI equations and NI unknowns whose solution provides a discrete approximate solution of the problem (1)-(2).

3 | A PROCEDURE FOR GENERATING DISCRETIZATIONS ADAPTED TO THE PARTIAL DIFFERENTIAL EQUATION

Our goal is to obtain a discretization adapted to the problem to be solved. The strategy we propose consists of solving the problem twice but with different resolution purposes and different discretizations. In the first step, we use a coarse uniform discretization and compute the absolute values of the gradients. Depending on these values, we place more points where the gradients are higher and fewer points where the gradients are lower. In the second stage, we calculate the approximate solution using the adapted discretization generated in the previous stage. To generate the adapted discretization and for each point of the initial discretization, we consider an area of influence where the new points are placed.

We have divided this section into two parts, addressing the generation of interior points and the generation of boundary points. In the first example, we distinguish between interior points generated from interior points and from boundary points.

3.1 | Procedure for generating interior points

Firstly, we use a coarse uniform discretization to get an initial approximate solution applying a second-order approximation. With these results, we compute the euclidean norm of the gradients, ∇_k, at the interior points by means of the GFDM,

$$\nabla_k = \sqrt{\left(\frac{\partial u_0}{\partial x}\right)_k^2 + \left(\frac{\partial u_0}{\partial y}\right)_k^2}, \; k = 1, \ldots, \mathsf{NI}. \tag{8}$$

Each interior point of the domain has the same square influence area, with the geometric center at the respective point (see Figure 3). The side length Δ of this area is the shortest distance between two points in all the domain.

Let $\bar{\nabla}$ be the average gradient, σ the standard deviation of the gradients and β a constant. The general rule of thumb for the number of points that each area of influence will have is as follows:

i) 4 points if $\bar{\nabla} + \beta\sigma \leqslant \nabla_k < \bar{\nabla} + 2\beta\sigma$.

ii) 9 points if $\nabla_k \geqslant \bar{\nabla} + 2\beta\sigma$.

iii) 1 point, otherwise.

Within the influence areas the points are distributed in a regular way as shown in Figure 3. If the influence area has 4 points, then we have a distance between them of $h = \Delta/3$ and if it has 9 points, the distance is $h = \Delta/4$. To facilitate a smooth transition between the different influence areas, we multiply h by an expansion coefficient α, checking that the points do not leave their influence area. This α is given by

$$\alpha = 1 + \frac{2\eta}{n-1}, \tag{9}$$

where η is an expansion percentage ($0 \leq \eta \leq 1$) and n is the number of points inside of the influence area in one direction, ($\mathsf{n} = 2$ in case of 4 points and $\mathsf{n} = 3$ in case of 9 points). So, α is just a percentage of broadening of the regular grid in the influence area. We give below the details of how α is obtained.

As the Δ value is invariable considering the same discretization M, then we established a limit in α to avoid that any point be inserted outside of the influence area to which it belongs.

Considering $\alpha \geq 1$, we have

$$(n-1)h \leq \sum_{i=1}^{n-1} \alpha h \leq \Delta \Rightarrow (n-1)h \leq \alpha h(n-1) \leq \Delta. \tag{10}$$

Since $\mathsf{h} = \dfrac{\Delta}{\mathsf{n}+1}$, then it is

$$(n-1)\frac{\Delta}{n+1} \leq \alpha\Delta\frac{n-1}{n+1} \leq \Delta, \tag{11}$$

from which it follows that

$$1 \leq \alpha \leq \frac{n+1}{n-1}. \tag{12}$$

The parameters involved in (12) are illustrated in Figure 2.

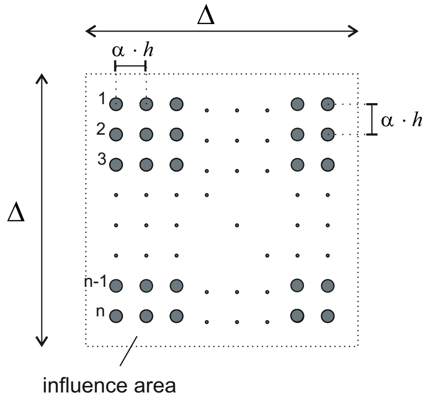


FIGURE 2 Parameters involved inside of an influence area.

Finally, we adopted an expansion coefficient given by

$$\alpha = 1 + \eta(\mathsf{max}(\alpha) - \mathsf{min}(\alpha)) = 1 + \frac{2\eta}{n-1}. \tag{13}$$

After inserting the corresponding points in the influence areas, we check if there are influence areas with a single point and at least one of the four closest areas of influence has 9 points. In such a case, we treat the area of influence of the single point as if it were the case with 4 points. In this way, we smooth the transition between influence areas, as can be seen in Figure 3.

Further, since we do not have the gradient values for the boundary points, we associate to each boundary point the number of points containing the area of influence of the closest interior point. Then we place the points as before but without considering the points that fall outside the domain and that are located at a distance less than $\Delta/10$ from the boundary. We also use two additional restrictions in this case, on the one hand, an expansion percentage (η) 50% higher than that used in the previous case and, on the other hand, if one of the interior points added from the boundary is less than $\Delta/2$ from any other interior point, we keep it and remove the other interior point. Note that the inserted points in this section are not in Γ (see Figure (4) for details).

3.2 | Procedure for generating boundary points

In case a boundary point generates an area of influence with 4 or 9 points, then we do an interpolation to also insert points in the boundary Γ.

First we do a cubic spline interpolation considering all the boundary points. The interpolation requires two essential steps: (1) a spline representation of the boundary points is computed, and (2) the spline is evaluated at the desired points. Assuming a central point and its two closest neighbors at Γ, we insert two points, each one parametrically equidistant from the central point to each neighbor. Of course, in case of overlapping points, only one of them is added (see red points in Figure 4).

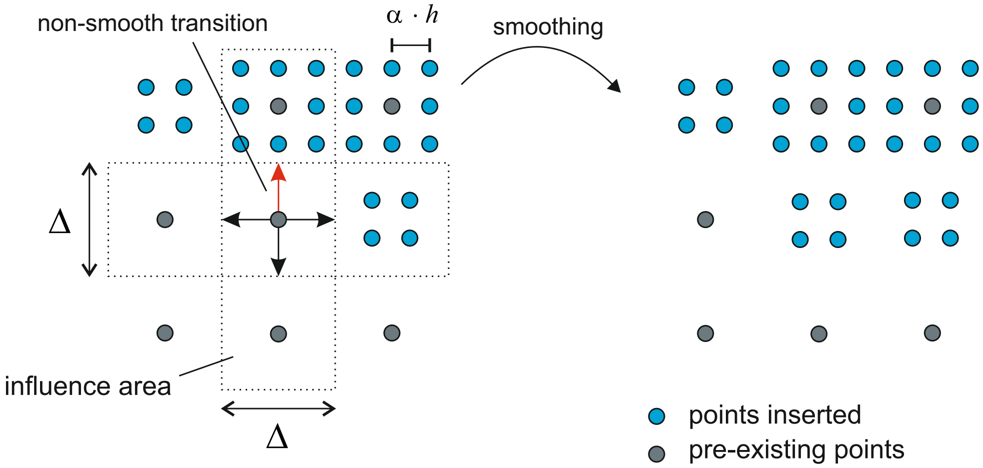


FIGURE 3 The insertion of points within the areas of influence (left) and the discretization smoothing process (right).

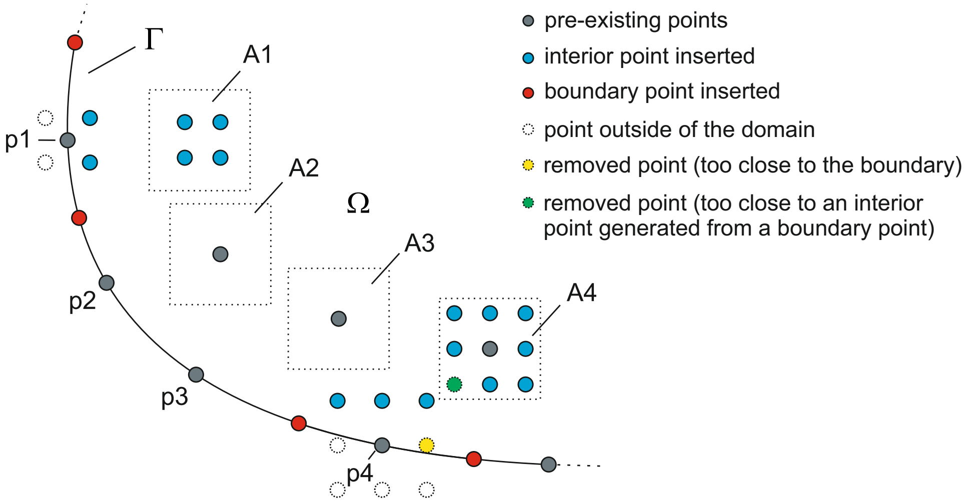

Γ
A1
p1
A2
Ω
A3
p2
A4
p3
p4
pre-existing points
interior point inserted
boundary point inserted
point outside of the domain
removed point (too close to the boundary)
removed point (too close to an interior point generated from a boundary point)

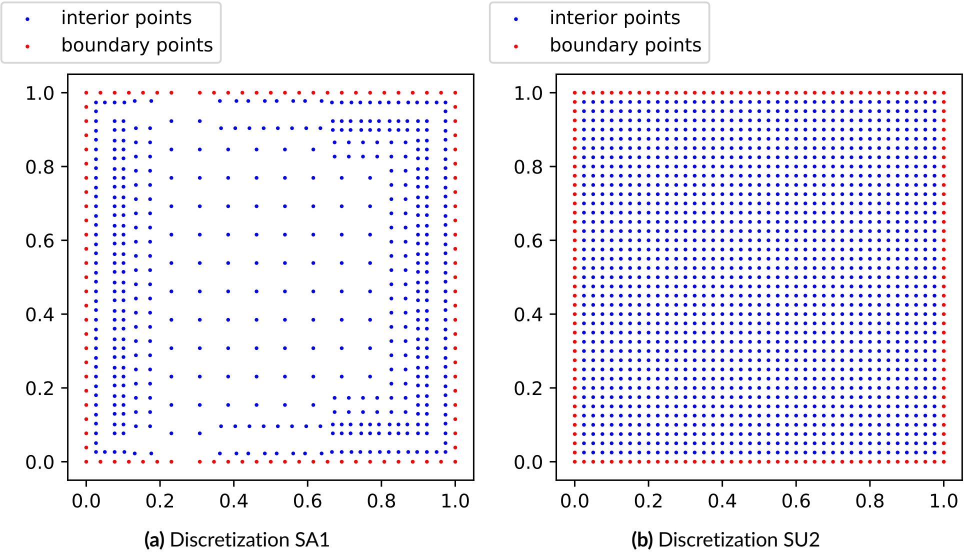
FIGURE 4 Interior points generated from boundary points and boundary points inserted. The points p1, p2, p3,and p4 have the closest interior areas A1, A2, A3, and A4 with 4, 1, 1, and 9 points, respectively. The white points do not exist because they are outside the domain, the yellow point is removed because it is too close to the boundary, and the green point is removed because it is close to an interior point generated from a boundary point. The blue and red points are interior and boundary points inserted, respectively.

4 | NUMERICAL RESULTS

In all examples, we used the weighting function $w_i = \|\mathbf{x}_i - \mathbf{x}_0\|_2^{-4}$, $i = 1, 2, \dots, m$. For the first stage, the calculation of the gradients, we used the second-order approximation and stars with 8 points formed by the distance criterion. For the second stage, the resolution of the problem, we used the fourth-order approximation and stars with 18 initial points for examples in sections 4.1 and 4.2, and we used the second-order approximation and stars with 16 points formed by the distance criterion for examples in section 4.3.

In all examples, we chose $\eta = 0.25$ and $\beta = 0.5$ and compared the adapted discretization with the two regular discretizations whose errors contain the error due to the adapted discretization. We used the following global error formula:

$$\text{error}(\%) = \frac{\sqrt{\dfrac{\sum_{j=1}^{NI} (U_j - u_j)^2}{NI}}}{u_{\text{máx}}} \cdot 100, \tag{14}$$



(a) Discretization SA1 **(b)** Discretization SU2

FIGURE 5 Adapted and uniform discretizations in Example 1.

where $\mathrm{U_j}$ is the exact value of the solution at the point $(\mathrm{x_j}, \mathrm{y_j})$, $\mathrm{u_j}$ is the approximate solution provided by the GFDM at the same point, and $\mathrm{u_{max}}$ is the maximum value of the approximations given by the GFDM.

In all examples, by uniform discretization we mean a uniform discretization over the entire domain except for the boundaries.

We wrote the codes in Python and used an Intel i7-8750H processor to execute the codes. The calculated values of speedups in each example mean the ratio between the runtime of an uniform and a adapted discretization. Of course, when we put the runtime of an adapted discretization, we included the stage of the calculation of the gradients.

In section 4.1, we show the errors and the code runtimes in adapted discretizations. We study the influence on the proposed technique for different values of η and β concerning the accuracy in section 4.2. Finally, in section 4.3, we show the errors and the code runtimes of adapted discretizations with second-order approximations.

4.1 | Adapted discretizations with fourth-order approximation

4.1.1 | Example 1

Consider the square domain

$$\mathrm{D1} = \{(\mathrm{x}, \mathrm{y}) \in \mathbb{R}^2 | 0 \leq \mathrm{x}, \mathrm{y} \leq 1\}$$

and the boundary value problem given by the following partial differential equation:

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = 2500\mathrm{e}^{-50x} + 25\mathrm{e}^{-5+5x} + 25\mathrm{e}^{-5y} + 25\mathrm{e}^{-5+5y} \tag{15}$$

whose exact solution is

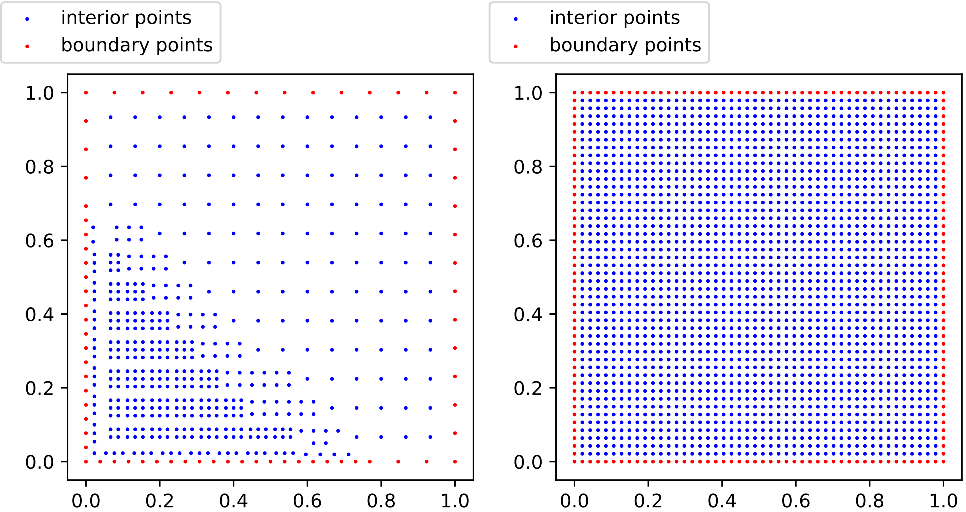
$$U(x, y) = \mathrm{e}^{-50x} + \mathrm{e}^{-5y} + \mathrm{e}^{-5+5x} + \mathrm{e}^{-5+5y}. \tag{16}$$

We solve (15) in the domain D1 considering an initial discretization with 196 points. We denote by SA1 the resulting adapted discretization that can be seen in Figure (5a) with 596 points. The uniform discretizations SU1 and SU2 have 1600 and 1681 points, respectively. The uniform discretization SU2 is shown in Figure 5b.

Table 1 shows the errors and execution times of all discretizations. In the adapted discretization SA1, we achieved similar accuracy with a decrease of approximately 65% of the number of points and with a speedup of approximately 4 relative to the uniform discretization SU2.

TABLE 1 Global error and execution times in Example 1

Discretization	Total number of points	Error (%)	Time (s)
SU1	1600	8.63e-1	7.02
SU2	1681	7.93e-1	7.92
SA1 (adapted)	596	8.10e-1	2.05



(a) Discretization SA2

(b) Discretization SU4

FIGURE 6 Adapted and uniform discretizations in Example 2.

TABLE 2 Global error and execution times in Example 2

Discretization	Total number of points	Error (%)	Time (s)
SU3	2254	5.96e-1	11.00
SU4	2304	1.94e-2	11.19
SA2 (adapted)	526	3.34e-2	1.79

4.1.2 | Example 2

Consider the boundary value problem given by the following partial differential equation:

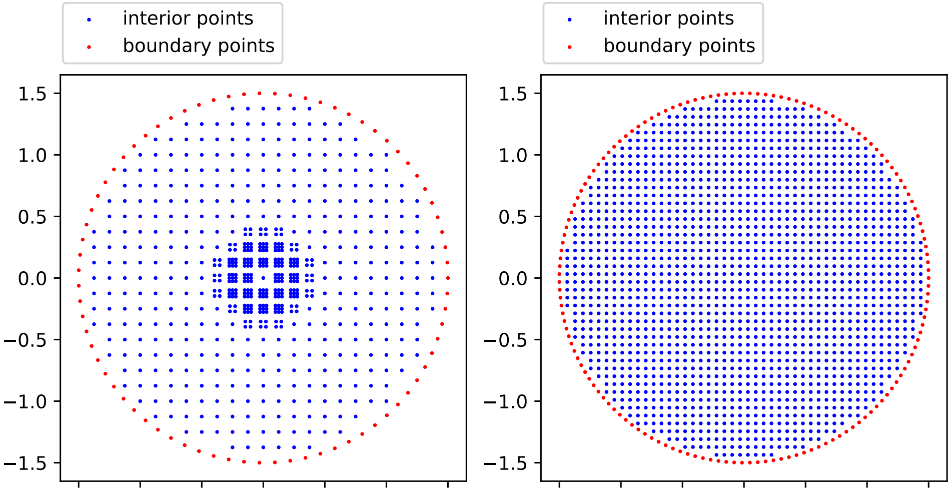
$$\frac{\partial^2 U}{\partial x^2} - \frac{\partial^2 U}{\partial y^2} = 0, \tag{17}$$

whose exact solution is

$$U(x, y) = \mathrm{e}^{-x-y-2}. \tag{18}$$

We solve (17) in the domain D1 using an initial discretization with 220 points. We denote by SA2 the adapted discretization that can be seen in Figure (6a) with 526 points. The uniform discretizations SU3 and SU4 have 2254 and 2304 points, respectively. The discretization SU4 is shown in Figure 6b.

Table 2 shows the errors and execution times of all discretizations. In the adapted discretization SA2, we achieved similar accuracy with a decrease of approximately 75% of the number of points and with a speedup of approximately 6 relative to the uniform discretization SU4.



(a) Discretization CA1 **(b)** Discretization CU2

FIGURE 7 Adapted and uniform discretizations in Example 3.

TABLE 3 Global error and execution times in Example 3

Discretization	Total number of points	Error (%)	Time (s)
CU1	1719	9.61e-1	15.94
CU2	1799	6.17e-1	21.75
CA1 (adapted)	696	9.38e-1	4.29

4.1.3 | Example 3

Consider the circular domain

$$D2 = \{(x, y) \in \mathbb{R}^2 | x^2 + y^2 \leq 1.5^2\}$$

and the boundary value problem given by the following partial differential equation

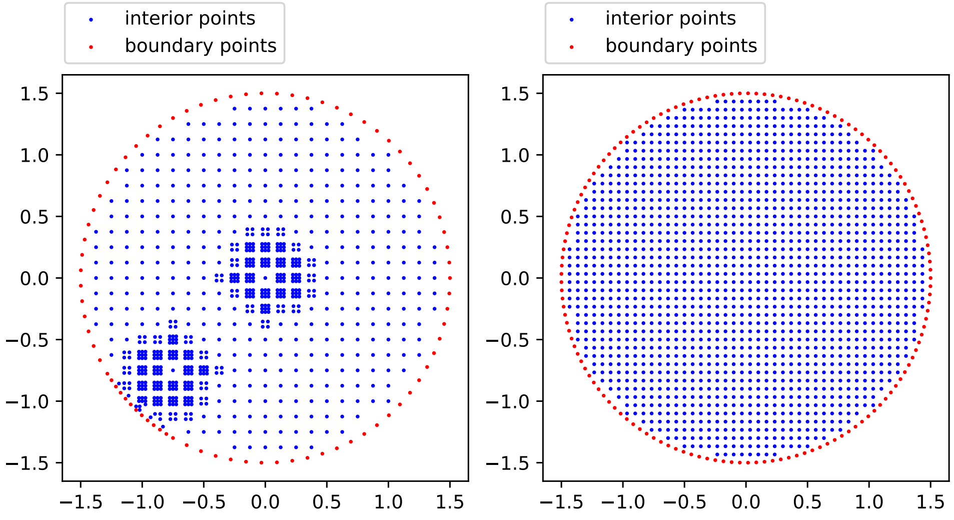
$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = \frac{8x^2 - 4 + 8y^2}{(x^2 + y^2 + 0.01)^3}, \tag{19}$$

with exact solution

$$U(x, y) = \frac{1}{x^2 + y^2 + 0.01}. \tag{20}$$

We solve (19) in the domain D2 using an initial discretization with 488 points. We denote by CA1 the adapted discretization that can be seen in Figure (7a) with 696 points. The uniform discretizations CU1 and CU2 have 1719 and 1799 points, respectively. The discretization CU2 is shown in Figure 7b.

Table 3 shows the errors and execution times of all discretizations. In the adapted discretization CA1, we achieved similar accuracy with a decrease of approximately 60% of the number of points and with a speedup of approximately 4 relative to the uniform discretization CU1.



(a) Discretization CA2 **(b)** Discretization CU4

FIGURE 8 Adapted and uniform discretizations in Example 4.

TABLE 4 Global error and execution times in Example 4

Discretization	Total number of points	Error (%)	Time (s)
CU3	1579	1.56	13.30
CU4	1653	1.00	14.20
CA2 (adapted)	849	1.19	7.75

4.1.4 | Example 4

Consider the boundary value problem given by the following partial differential equation

$$\frac{\partial^2 U}{\partial x^2}+\frac{\partial^2 U}{\partial y^2}=\frac{8x^2}{(x^2+y^2+0.01)^3}-\frac{4}{(x^2+y^2+0.01)^2}+\frac{2(2x+1.50)^2}{((x+0.75)^2+(y+0.75)^2+0.01)^3}-\frac{4}{((x+0.75)^2+(y+0.75)^2+0.01)^2}$$
$$+\frac{8y^2}{(x^2+y^2+0.01)^3}+\frac{2(2y+1.50)^2}{((x+0.75)^2+(y+0.75)^2+0.01)^3} \tag{21}$$

with exact solution

$$U(x,y)=\frac{1}{x^2+y^2+0.01}+\frac{1}{(x+0.75)^2+(y+0.75)^2+0.01}. \tag{22}$$

We solve (21) in D2 using an initial discretization with 220 points. We denote by CA2 the adapted discretization that can be seen in Figure (8a) with 849 points. The uniform discretizations CU3 and CU4 have 1579 and 1653 points, respectively. The discretization CU4 is shown in Figure (8b).

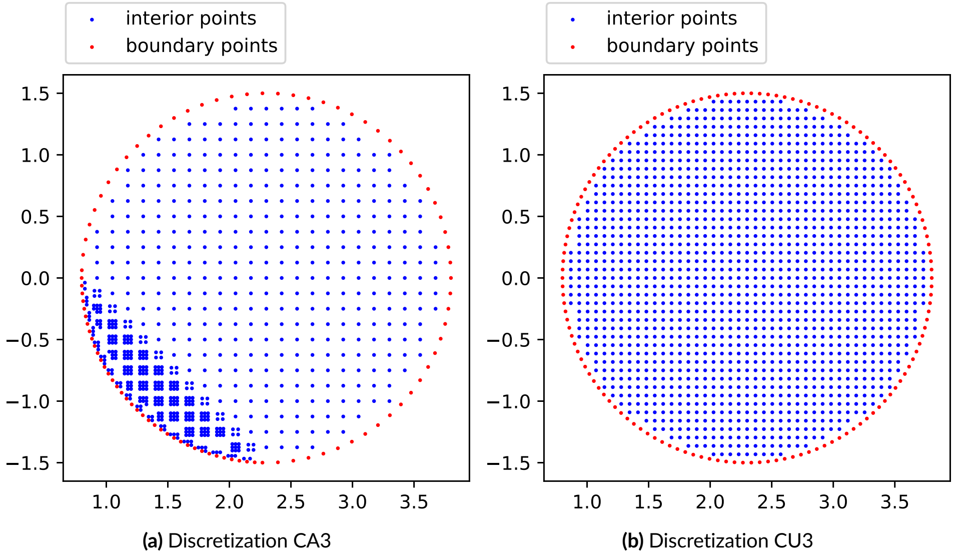
Table 4 shows the errors and execution times of all discretizations. In the adapted discretization CA2 we achieved similar accuracy with a decrease of approximately 50% of the number of points and with a speedup of approximately 2 relative to the uniform discretization CU4.

4.1.5 | Example 5

Consider the circular domain

$$D3=\{(x,y)\in\mathbb{R}^2|(x-2.3)^2+y^2\leq 1.5^2\}$$

and the boundary value problem given by the following partial differential equation



(a) Discretization CA3 **(b)** Discretization CU3

FIGURE 9 Adapted and uniform discretizations in Example 5.

TABLE 5 Global error and execution times in Example 5

Discretization	Total number of points	Error (%)	Time (s)
CU6	1506	1.56e-1	10.13
CU3	1579	1.48e-1	13.30
CA3 (adapted)	798	1.54e-1	4.95

$$\frac{\partial^2 U}{\partial x^2} - 2\frac{\partial^2 U}{\partial x \partial y} + \frac{\partial^2 U}{\partial y^2} + \frac{\partial U}{\partial x} - \frac{\partial U}{\partial y} = 0, \tag{23}$$

with exact solution

$$U(x, y) = \frac{1}{x + y}. \tag{24}$$

We solve (23) in D3 using an initial discretization with 220 points. We denote by CA3 the adapted discretization that can be seen in Figure (9a) with 798 points. The uniform discretizations CU3 and CU6 have 1579 and 1506 points, respectively. The discretization CU3 is shown in Figure (9b).

Table 5 shows the results and execution times of all discretizations. In the adapted discretization CA3 we achieved similar accuracy with a decrease of approximately 50% of the number of points and with a speedup of approximately 3 relative to the uniform discretization CU3.

4.1.6 | Example 6

Consider a circle with a lemniscate geometry inside it

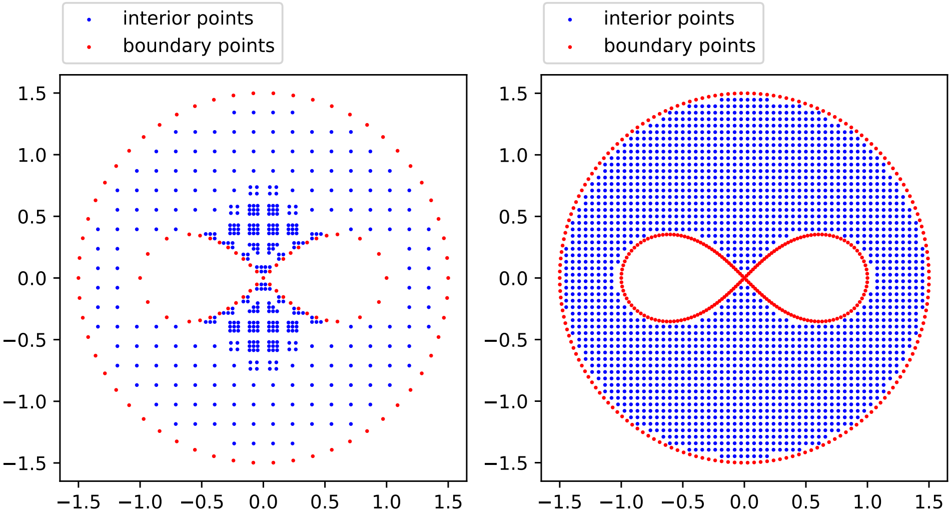
$$D4 = C - L,$$

where $C = \{(x, y) \in \mathbb{R}^2 | x^2 + y^2 <= 1.5^2\}$ and $L = \{(x, y) \in \mathbb{R}^2 | (x^2 + y^2)^2 - (x^2 - y^2) < 0\}$.

Consider the boundary value problem given by the following partial differential equation

$$\frac{\partial^2 U}{\partial x^2} - \frac{\partial^2 U}{\partial y^2} = \frac{8x^2 - 8y^2}{(x^2 + y^2 + 0.01)^3} \tag{25}$$

with exact solution given in (20).



(a) Discretization LA

(b) Discretization LU2

FIGURE 10 Adapted and uniform discretizations in Example 6.

TABLE 6 Global error and execution times in Example 6 with fourth-order approximations.

Discretization	Total number of points	Error (%)	Time (s)
LU1	2032	4.24e-1	23.50
LU2	2416	1.71e-1	73.40
LA (Adapted)	493	1.19e-1	1.61

We solve (25) in the domain D4 using an initial discretization with 280 points. We denote by LA the adapted discretization that can be seen in Figure (10a) with 493 points. The uniform discretizations LU1 and LU2 have 2032 and 2416 points, respectively. The discretization LU2 is shown in Figure (10b).

Table 6 shows the errors and runtime of all discretizations. In the adapted discretization LA, we achieved a similar accuracy with a decrease of approximately 75% of the number of points and with a speedup of approximately 37 relative to the uniform discretization SU4.

4.2 | Influence of the parameters η and β

On the one hand, the parameter η influences the distance between points within the area of influence and, therefore, affects the accuracy of the derivatives. On the other hand, the parameter β controls the total amount of inserted points, the smaller the β, the more points will be added.

Figure 11 shows the global errors resulting from the variation of the parameter η in all adapted discretizations used in section 4.1. We varied η between 0.1 and 0.4 with a step equal to 0.05. In general, the results indicate that intermediate values of η provide more accurate results. In this paper, we have chosen the value of $\eta = 0.25$.

Figure 12 shows the global errors due to variation of the parameter β. We varied β between 0 and 2 with a step equal to 0.5. In general, values of β lower than 1 generate more accurate results. In this paper, we have chosen the value of $\beta = 0.5$.

4.3 | Adapted discretizations with second-order approximations

We solve (15) and (17) using the second-order approximation and an initial discretization with 504 points.

We denote by SA3 and SA4 the adapted discretizations with 1134 and 1233 points, respectively (see Figures (13a) and (13c)). The uniform discretizations SU5, SU6, SU7, and SU8 have 2916, 3136, 5329 and 5476 points, respectively. The discretizations SU6 and SU8 are shown in Figures 13b and 13d, respectively.

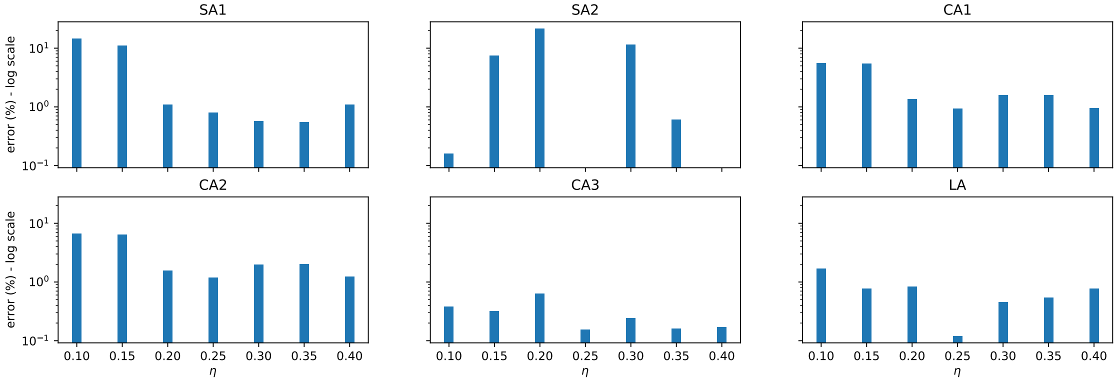


FIGURE 11 Relation between the value of η and the global errors.

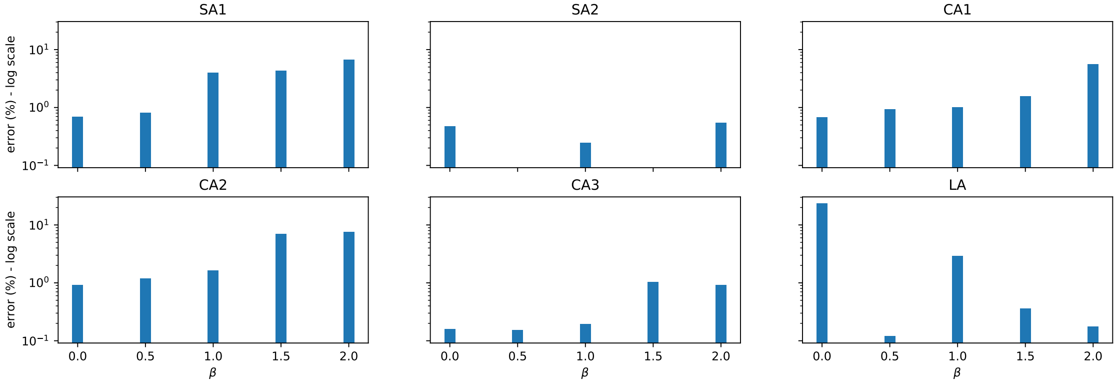


FIGURE 12 Relation between the value of β and the global errors.

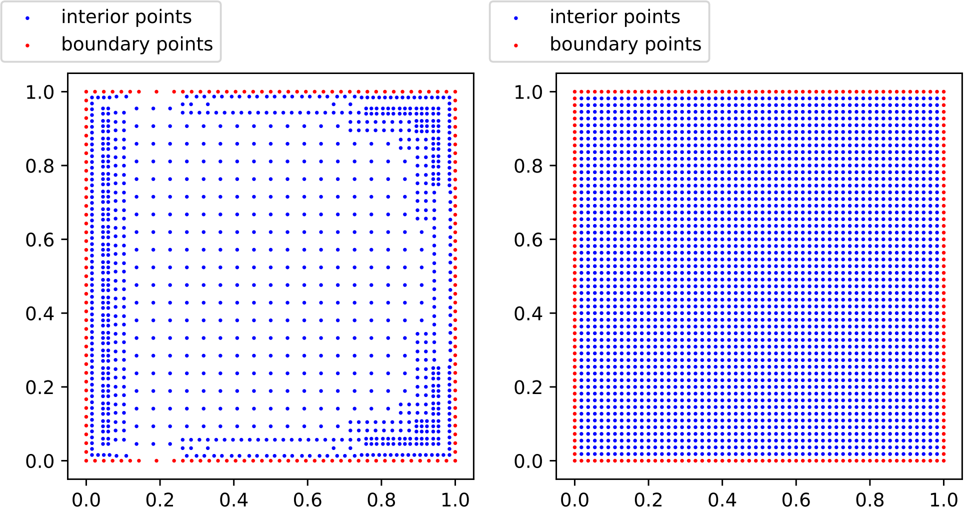
TABLE 7 Global error and execution times in Example 1 with second-order approximation

Discretization	Total number of points	Error (%)	Time (s)
SU5	2916	1.38	6.86
SU6	3136	1.32	7.93
SA3 (adapted)	1134	1.32	1.48

Tables 7 and 8 show the results and execution times of all discretizations. In the adapted discretizations SA3 and SA4, we achieved a similar accuracy with a decrease of approximately 65% and 75%, respectively, of the number of points relative to the uniform discretizations SU6 and SU8, respectively. Comparing the same discretizations, we get a speedup of approximately 5 and 15, respectively.

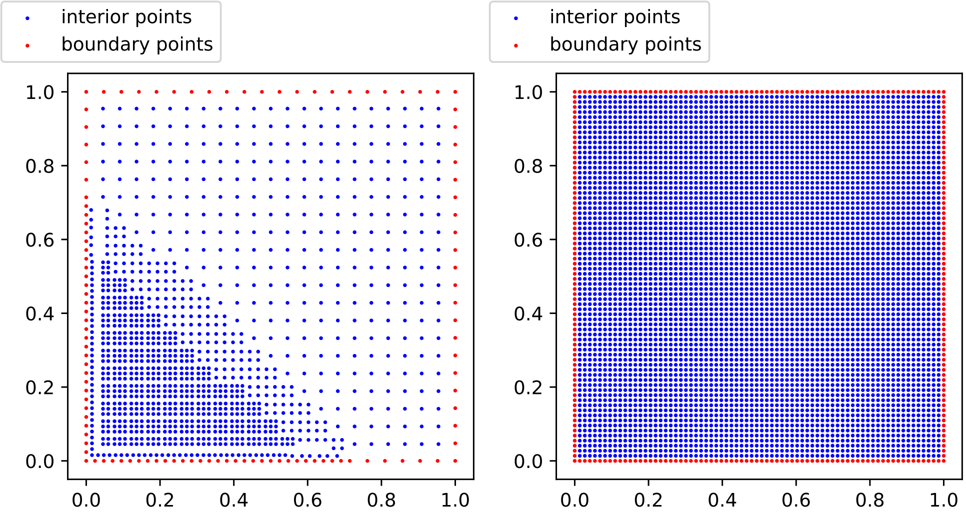
5 | CONCLUSIONS

Given a problem for which it is necessary to solve a differential equation in a domain employing the GFDM, the discretizations performed in such a domain generally have an approximately constant point density. Possibly, the most widespread ways of discretizing are by means of mesh-based preprocessors or simply in a regular way, allowing for irregularities where this is not possible.



(a) Discretization SA3

(b) Discretization SU6



(c) Discretization SA4

(d) Discretization SU8

FIGURE 13 Adapted and uniform discretizations in examples 1 and 2 with second-order approximations

TABLE 8 Global error and execution times in Example 2 with second-order approximation

Discretization	Total number of points	Error (%)	Time (s)
SU7	5329	9.59e-1	22.55
SU8	5476	8.06e-2	25.85
SA4 (adapted)	1233	1.56e-1	1.75

However, these forms of discretization do not allow to capture the characteristics of the problem and to take advantage of the benefits of GFDM. There are particular cases where the authors use discretizations with non-constant density in the domain to obtain higher accuracy using a smaller number of points.

We propose in this paper to use discretizations adapted to the problem in general. To do so, we solve the problem in two stages. In the first stage, we solve the problem to compute the gradients using a regular coarse discretization. Once the gradients are calculated, we distribute the points according to the gradient values. Finally, we solve the problem considering the adapted discretization.

We have shown the performance of the proposed strategy for fourth-order approximations but we have also shown some examples with second-order approximations where the results have been similar.

On the one hand, the adapted discretizations provide the same accuracy as a regular discretization, but with a smaller number of points, with reductions above 50% in all our examples. In addition, the initial coarse discretization, that automatically generates the adapted discretization, has required between 9% and 27% of the points of the regular discretization.

On the other hand, the computational time required to solve the problem with these adapted discretizations, taking into account the entire process, is less, with reductions above 50% in all our examples.

References

1. Gingold RA, Monaghan JJ. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society* 1977; 181(3): 375-389. doi: 10.1093/mnras/181.3.375

2. Oñate E, Idelsohn S, Zienkiewicz O, Taylor R, Sacco C. A stabilized finite point method for analysis of fluid mechanics problems. *Computer Methods in Applied Mechanics and Engineering* 1996; 139(1): 315-346. doi: https://doi.org/10.1016/S0045-7825(96)01088-2

3. Liu GR, Gu Y. A point interpolation method for two-dimensional solids. *International Journal for Numerical Methods in Engineering* 2001; 50(4): 937–951. doi: https://doi.org/10.1002/1097-0207(20010210)50:4<937::AID-NME62>3.0.CO;2-X

4. Wyatt M, Taylor , Davies G, snell C. A new difference based finite element method. *Proceedings of the Institution of Civil Engineers* 1975; 59(3): 395–409. doi: https://doi.org/10.1680/iicep.1975.3672

5. Liszka T, Orkisz J. The finite difference method at arbitrary irregular grids and its application in applied mechanics. *Computers & Structures* 1980; 11(1-2): 83–95. doi: https://doi.org/10.1016/0045-7949(80)90149-2

6. Benito JJ, Ureña F, Gavete L. Influence of several factors in the generalized finite difference method. *Applied Mathematical Modelling* 2001; 25(12): 1039–1053. doi: https://doi.org/10.1016/S0307-904X(01)00029-4

7. Milewski S. Selected computational aspects of the meshless finite difference method. *Numerical Algorithms* 2012; 63(1): 107–126. doi: 10.1007/s11075-012-9614-6

8. Milewski S, Putanowicz R. Higher order meshless schemes applied to the finite element method in elliptic problems. *Computers & Mathematics with Applications* 2019; 77(3): 779–802. doi: https://doi.org/10.1016/j.camwa.2018.10.016

9. Jaworska I, Orkisz J. Higher order multipoint method–from Collatz to meshless FDM. *Engineering Analysis with Boundary Elements* 2015; 50: 341–351. doi: https://doi.org/10.1016/j.enganabound.2014.09.007

10. Jaworska I. Higher order multipoint Meshless Finite Difference Method for two-scale analysis of heterogeneous materials. *International Journal for Multiscale Computational Engineering* 2019; 17(3). doi: 10.1615/IntJMultCompEng.2019028866

11. Albuquerque-Ferreira AC, Ureña M, Ramos H. The generalized finite difference method with third-and fourth-order approximations and treatment of ill-conditioned stars. *Engineering Analysis with Boundary Elements* 2021; 127: 29–39. doi: https://doi.org/10.1016/j.enganabound.2021.03.005

12. Benito JJ, García Á, Gavete ML, Negreanu M, Ureña F, Vargas AM. Convergence and Numerical Solution of a Model for Tumor Growth. *Mathematics* 2021; 9(12): 1355. doi: 10.3390/math9121355

13. Li PW, Fan CM, Grabski JK. A meshless generalized finite difference method for solving shallow water equations with the flux limiter technique. *Engineering Analysis with Boundary Elements* 2021; 131: 159-173. doi: https://doi.org/10.1016/j.enganabound.2021.06.022

14. Ureña M, Benito JJ, Ureña F, Salete E, Gavete L. Application of generalised finite differences method to reflection and transmission problems in seismic SH waves propagation. *Mathematical Methods in the Applied Sciences* 2018; 41(6): 2328–2339. doi: https://doi.org/10.1002/mma.4268

15. Korkut F, Tokdemir T, Mengi Y. The use of generalized finite difference method in perfectly matched layer analysis. *Applied Mathematical Modelling* 2018; 60: 127-144. doi: https://doi.org/10.1016/j.apm.2018.03.014

16. Korkut F, Mengi Y, Tokdemir T. On the use of complex stretching coordinates in generalized finite difference method with applications in inhomogeneous visco-elasto dynamics. *Engineering Analysis with Boundary Elements* 2022; 134: 466-490. doi: https://doi.org/10.1016/j.enganabound.2021.10.014

17. Shao M, Song L, Li PW. A generalized finite difference method for solving Stokes interface problems. *Engineering Analysis with Boundary Elements* 2021; 132: 50-64. doi: https://doi.org/10.1016/j.enganabound.2021.07.002

18. García A, Negreanu M, Ureña F, Vargas AM. Convergence and numerical solution of nonlinear generalized Benjamin-Bona-Mahony-Burgers equation in 2D and 3D via generalized finite difference method. *International Journal of Computer Mathematics* 2021; 0(0): 1-21. doi: 10.1080/00207160.2021.1989423

19. Hosseini SM. Analysis of elastic wave propagation in a functionally graded thick hollow cylinder using a hybrid mesh-free method. *Engineering Analysis with Boundary Elements* 2012; 36(11): 1536-1545. doi: https://doi.org/10.1016/j.enganabound.2012.05.001

20. Hosseini SM. Application of a hybrid mesh-free method based on generalized finite difference (GFD) method for natural frequency analysis of functionally graded nanocomposite cylinders reinforced by carbon nanotubes. *Computer Modeling in Engineering and Sciences-CMES* 2013; 95(1): 1–29. doi: 10.3970/cmes.2013.095.001

21. Collatz L. *The numerical treatment of differential equations*. Springer Science & Business Media . 2012.

22. Forsythe GE, Wasow WR. Finite-difference methods for partial differential equations. 1960.

23. Jensen PS. Finite difference techniques for variable grids. *Computers & Structures* 1972; 2(1-2): 17–29. doi: https://doi.org/10.1016/0045-7949(72)90020-X

24. Milewski S. Development of simple effective cloud of nodes and triangular mesh generators for meshless and element-based analyses-implementation in Matlab. *Computer Assisted Methods in Engineering and Science* 2018; 24(3): 157–180. doi: http://dx.doi.org/10.24423/cames.192

25. Wei J, Wang S, Hou Q, Dang J. Generalized finite difference time domain method and its application to acoustics. *Mathematical Problems in Engineering* 2015; 2015. doi: https://doi.org/10.1155/2015/640305

26. Ferreira ACA, Ribeiro PMV. Reduced-order strategy for meshless solution of plate bending problems with the generalized finite difference method. *Latin American Journal of Solids and Structures* 2019; 16(1): 1–21. doi: https://doi.org/10.1590/1679-78255191

27. Chi-Mou N. A quadrilateral finite difference plate element for nonlinear transient analysis of panels. *Computers & Structures* 1982; 15(1): 1-10. doi: https://doi.org/10.1016/0045-7949(82)90028-1

28. Tseng A, Gu S. A finite difference scheme with arbitrary mesh systems for solving high-order partial differential equations. *Computers & Structures* 1989; 31(3): 319-328. doi: https://doi.org/10.1016/0045-7949(89)90379-9

29. Orkisz J. Finite difference method. *Handbook of Computational Solid Mechanics* 1998: 336–432.

30. García-March MA, Arevalillo-Herráez M, Villatoro FR, Giménez F, de Córdoba PF. A generalized finite difference method using Coatmèlec lattices. *Computer Physics Communications* 2009; 180(7): 1125-1133. doi: https://doi.org/10.1016/j.cpc.2009.01.015

31. Benito JJ, Ureña F, Ureña M, Salete E, Gavete L. A new meshless approach to deal with interfaces in seismic problems. *Applied Mathematical Modelling* 2018; 58: 447-458. doi: https://doi.org/10.1016/j.apm.2018.02.014

□