

Article

A Novel Spatio-Temporal Fully Meshless Method for Parabolic PDEs

Juan José Benito ^{1,†}, Ángel García ^{1,†}, Mihaela Negreanu ^{2,†} , Francisco Ureña ^{1,*,†}  and Antonio M. Vargas ^{2,†}

¹ Escuela Técnica de Ingenieros Industriales, Universidad Nacional de Educación a Distancia, 28040 Madrid, Spain; jjbenito@ind.uned.es (J.J.B.); agarcia@edu.jccm.es (Á.G.)

² Departamento de Análisis Matemático y Matemática Aplicada, Instituto de Matemática Interdisciplinar, Universidad Complutense de Madrid, 28040 Madrid, Spain; negreanu@mat.ucm.es (M.N.); antonvar@ucm.es (A.M.V.)

* Correspondence: furenaprieto@gmail.com

† These authors contributed equally to this work.

Abstract: We introduce a meshless method derived by considering the time variable as a spatial variable without the need to extend further conditions to the solution of linear and non-linear parabolic PDEs. The method is based on a moving least squares method, more precisely, the generalized finite difference method (GFDM), which allows us to select well-conditioned stars. Several 2D and 3D examples, including the time variable, are shown for both regular and irregular node distributions. The results are compared with explicit GFDM both in terms of errors and execution time.

Keywords: generalized finite differences; meshless method; parabolic partial differential equations

MSC: 65M06



Citation: Benito, J.J.; García, Á.; Negreanu, M.; Ureña, F.; Vargas, A.M. A Novel Spatio-Temporal Fully Meshless Method for Parabolic PDEs. *Mathematics* **2022**, *10*, 1870. <https://doi.org/10.3390/math10111870>

Academic Editor: Carlo Bianca

Received: 18 April 2022

Accepted: 27 May 2022

Published: 30 May 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

*Zum Raum wird hier die Zeit
(Here, time becomes space)
(RICHARD WAGNER, PARSIFAL)*

1. Introduction

A wide variety of physical phenomena are studied mathematically through evolution partial differential equations. The difference between parabolic evolution equations and elliptic equations is important, both from the point of view of mathematical analysis and numerical analysis. In this article, we are interested in the numerical treatment of parabolic PDEs. In particular, we focus on the numerical solution at a time T of the general parabolic PDE

$$\frac{\partial U}{\partial t} + \mathcal{L}(x, t, U) = h(x, t), \quad x \in \Gamma, \quad 0 < t \leq T \quad (1)$$

with the boundary and initial condition

$$U(x, t) = g(x, t), \quad x \in \partial\Gamma; \quad U(x, 0) = U_0(x). \quad (2)$$

Here, Γ is a bounded domain of \mathbb{R} or \mathbb{R}^2 , and the non-linear operator

$$\mathcal{L}(x, t, U) = f_1(x, t) \frac{\partial U}{\partial x} + f_2(x, t) \frac{\partial^2 U}{\partial x^2}, \quad \text{if } \Gamma \subset \mathbb{R}$$

or, if $\Gamma \subset \mathbb{R}^2$,

$$\begin{aligned}\mathcal{L}(x, t, U) = & f_1(x, y, t) \frac{\partial U}{\partial x} + f_2(x, y, t) \frac{\partial U}{\partial y} + f_3(x, y, t) \frac{\partial^2 U}{\partial x^2} \\ & + f_4(x, y, t) \frac{\partial^2 U}{\partial y^2} + f_5(x, y, t) \frac{\partial^2 U}{\partial x \partial y}.\end{aligned}$$

Throughout the paper, we assume enough regularity of the functions h, g, U_0 and f_i in order to provide classical solutions.

The time evolution of the modeled physical problem creates the need to study the propagation of the numerical error. This, in some cases, produces consistent numerical schemes that generate unstable solutions. The most common way to solve parabolic partial differential equations numerically is to use a different discretisation for the time and space derivatives. For instance,

$$\frac{\partial U}{\partial t} \approx \frac{U(x, y, t + \Delta t) - U(x, y, t)}{\Delta t}$$

is frequently used for the time derivative, which entails a study of the stability of the numerical method.

In this paper, we introduce a meshless method (see [1–4] for a detailed description of meshless methods) for resolving parabolic PDEs that consists of using the formulas given by the generalized finite difference method to discretise all the derivatives of the equation, including the time derivative. The researchers Jensen [5], Liszka and Orkisz [6], Orkisz [7] and Perrone and Kao [8] have contributed to the development of the GFDM, as well as Benito, Gavete and Ureña [9] who studied the influence of several factors and developed the h-adaptive method for the solution of the PDEs in 2D. During the last two decades, the method has attracted growing attention, not only because of its applications, but also, for instance, because some improvements to the one dimensional method have been made [10].

By considering the time variable as a spatial variable in a one-dimensional problem (or two-dimensional), we can find the 2D (or 3D) Taylor series of the solution, in the variables x and t (or x, y and t), and at any nodal point a desired discretization, without adding any extra conditions. In this way, we avoid the stability issues related to the time evolution of the problem. We transform problem (1) and (2) into an elliptic problem with boundary conditions (see Figure 1).

$$\text{B.C.} = \begin{cases} g(x, t), & (x, t) \in \partial\Gamma \times [0, T], \\ U_0(x), & (x, t) \in \Gamma \times \{t = 0\}, \\ U(x, t), & (x, t) \in \Gamma \times \{t = T\}. \end{cases} \quad (3)$$

Our procedure consists of defining a GFD star at each point of $\Gamma \times [0, T)$ and $\Gamma \times \{t = T\}$ and solving the elliptic problem without adding any extra condition or applying any other numerical method. A pictorial definition of the selection of the stars is given in Figure 2.

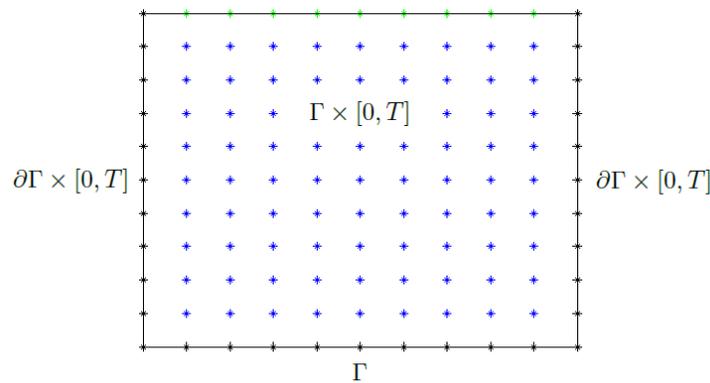


Figure 1. Computational domain of the problem, where the green nodes belongs to $\partial\Gamma \times [0, T]$ and Γ , and the blue nodes correspond to $\Gamma \times [0, T]$. The solution of the original problem corresponds to the solution at $\partial\Gamma \times \{t = T\}$ (green nodes).

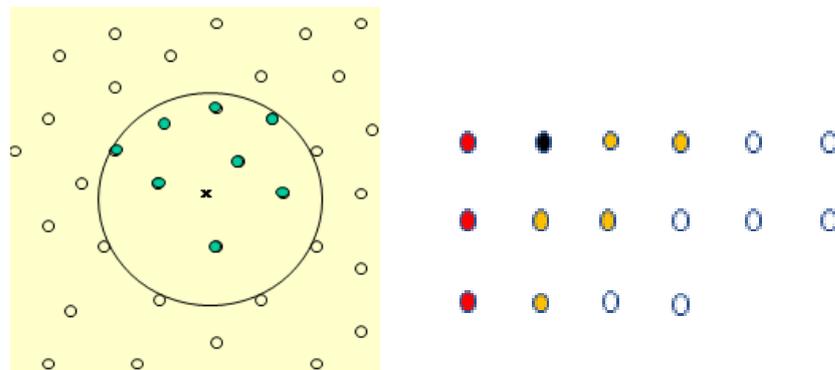


Figure 2. On the **left**, the distance criterion to choose the nodes of a star in the interior of $\Gamma \times [0, T]$ where we choose the eight nearest nodes. On the **right**, the star is centered at a node of $\Gamma \times \{t = T\}$ (the star is centered at the black point, the yellow points belong to $\Gamma \times [0, T]$ and the red ones to $\partial\Gamma \times [0, T]$).

For this reason, we have called this approach the “space time cloud method (STCM)”. The previous discretization of the computational domain for parabolic PDEs is not new. Several meshless numerical methods have recently been applied, such as a kernel-based method [11], the space–time diffuse approximation meshless method [12], the Trefftz method [13], the localized radial basis function collocation method [14] and the multi-quadratic method with the radial basis collocation method [15]. In addition, the GFDM has been applied to solve space-time problems in [16–18]. The novelty of the STCM method is to use only the generalized finite difference formulas to solve a time-dependent problem (parabolic in this case), which makes it unnecessary to study the stability and the calculation of the time step in the explicit case, or not to have to solve for each time step a system of equations in the implicit case. With the solution of a single system, the problem is solved. The advantage of STCM over the explicit GFDM is essentially not having to study the stability and, therefore, not having to calculate the time step and perform the calculations with that value. In addition, the advantage over FEM is that it solves the equation without the need for integration, it does not require mesh and the boundary conditions are simpler to implement, and, with respect to FDM, its main advantage is that irregular meshes can be used.

The structure of the paper is as follows. In Section 2, we introduce the meshless space time cloud method and derive the discretization of the Equations (1)–(3) for two and three dimensions. We provide several examples where we compare the present method with the standard GFDM applied to the original problem. Finally, some conclusions are drawn.

2. Space Time Cloud Method (STCM)

We introduce the STCM for the two dimensional case (that is, $\Gamma \subset \mathbb{R}$). Let $\Gamma^* = \Gamma \times [0, T]$ be a domain and

$$P = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subset \Gamma^*$$

a discretization of Γ^* with n points (see Figure 1). We designate each point of the discretization P as a point. $\Lambda_s = \{\mathbf{x}_0; \mathbf{x}_{0_1}, \dots, \mathbf{x}_{0_s}\} \subset P$ with the central node $\mathbf{x}_0 \in P$ and \mathbf{x}_{0_i} , ($i = 1, \dots, s$) $\in P$ is a set of points located near \mathbf{x}_0 . The selection of the points for each star can be made according to criteria as four quadrants or distance (see [9]).

With the previous notation, $\mathbf{x}_0 = (x_0, t_0)$ and $\mathbf{x}_{0_i} = (x_i, t_i)$, define $h_i = x_i - x_0$, $k_i = t_i - t_0$. Let us put $U_0 = U(\mathbf{x}_0)$ and $U_i = U(\mathbf{x}_{0_i})$, then, by the Taylor series expansion, we have

$$U_i = U_0 + h_i \frac{\partial U_0}{\partial x} + k_i \frac{\partial U_0}{\partial t} + \frac{1}{2} \left(h_i^2 \frac{\partial^2 U_0}{\partial x^2} + k_i^2 \frac{\partial^2 U_0}{\partial t^2} + 2h_i k_i \frac{\partial^2 U_0}{\partial x \partial t} \right) + R_2, \tag{4}$$

for $i = 1, \dots, s$, where R_2 is the remainder of the second order. Let us use the notations

$$\mathbf{v}_i^T = \{h_i, k_i, \frac{h_i^2}{2}, \frac{k_i^2}{2}, h_i k_i\}$$

and

$$\mathbf{\Pi}_5^T = \left\{ \frac{\partial u_0}{\partial x}, \frac{\partial u_0}{\partial t}, \frac{\partial^2 u_0}{\partial x^2}, \frac{\partial^2 u_0}{\partial t^2}, \frac{\partial^2 u_0}{\partial x \partial t} \right\}.$$

By truncation in (4) up to the second order, we can obtain a second-order approximation of U_i , which we shall denote u_i . Then, we define the weighted residual function:

$$T(u) = \sum_{i=1}^s [(u_0 - u_i) + h_i \frac{\partial u_0}{\partial x} + k_i \frac{\partial u_0}{\partial t} + \frac{1}{2} (h_i^2 \frac{\partial^2 u_0}{\partial x^2} + k_i^2 \frac{\partial^2 u_0}{\partial t^2} + 2h_i k_i \frac{\partial^2 u_0}{\partial x \partial t})]^2 \eta_i^2, \tag{5}$$

where $\eta_i = \eta(h_i, k_i)$ are positive weighting functions as [19]. We minimize the norm given by (5) with respect to the partial derivatives by considering the following linear system

$$\mathbf{W}\mathbf{\Pi}_5 = \boldsymbol{\beta}$$

where

$$\mathbf{W} = \begin{pmatrix} h_1 & h_2 & \dots & h_s \\ k_1 & k_2 & \dots & k_s \\ \vdots & \vdots & \vdots & \vdots \\ h_1 k_1 & h_2 k_2 & \dots & h_s k_s \end{pmatrix} \begin{pmatrix} \eta_1^2 & & & \\ & \eta_2^2 & & \\ & & \dots & \\ & & & \eta_s^2 \end{pmatrix} \begin{pmatrix} h_1 & k_1 & \dots & h_1 k_1 \\ h_2 & k_2 & \dots & h_2 k_2 \\ \vdots & \vdots & \vdots & \vdots \\ h_s & k_s & \dots & h_s k_s \end{pmatrix},$$

and

$$\boldsymbol{\beta}^T = \left(\sum_{i=1}^s (-u_0 + u_i) h_i \eta_i^2, \sum_{i=1}^s (-u_0 + u_i) k_i \eta_i^2, \sum_{i=1}^s (-u_0 + u_i) \frac{h_i^2 \eta_i^2}{2}, \sum_{i=1}^s (-u_0 + u_i) \frac{k_i^2 \eta_i^2}{2}, \sum_{i=1}^s (-u_0 + u_i) h_i k_i \eta_i^2 \right).$$

In [20], the authors proved that \mathbf{W} is a positive definite matrix and that the local truncation error is $\mathcal{O}(h_i^2, k_i^2)$. Then, \mathbf{W} admits a unique Cholesty decomposition, and, if we define

$$\mathbf{W}^{-1} = \mathbf{K}\mathbf{K}^T,$$

we have

$$\mathbf{\Pi}_5 = \mathbf{K}\mathbf{K}^T \boldsymbol{\beta}. \tag{6}$$

Thus, Equation (6) can be rewritten as

$$\mathbf{\Pi}_5 = -u_0 \mathbf{K}\mathbf{K}^T \sum_{i=1}^s \eta_i^2 \mathbf{v}_i + \mathbf{K}\mathbf{K}^T \sum_{i=1}^s u_i \eta_i^2 \mathbf{v}_i.$$

We denote the derivatives as follows

$$\left\{ \begin{aligned} \frac{\partial u(x_0, t_0)}{\partial x} &= -\varphi_{01}u_0 + \sum_{i=1}^s \varphi_{i1}u_i + \mathcal{O}(h_i^2, k_i^2), \text{ with } \varphi_{01} = \sum_{i=1}^s \varphi_{i1}, \\ \frac{\partial u(x_0, t_0)}{\partial t} &= -\varphi_{02}u_0 + \sum_{i=1}^s \varphi_{i2}u_i + \mathcal{O}(h_i^2, k_i^2), \text{ with } \varphi_{02} = \sum_{i=1}^s \varphi_{i2}, \\ \frac{\partial^2 u(x_0, t_0)}{\partial x^2} &= -\varphi_{03}u_0 + \sum_{i=1}^s \varphi_{i3}u_i + \mathcal{O}(h_i^2, k_i^2), \text{ with } \varphi_{03} = \sum_{i=1}^s \varphi_{i3}, \\ \frac{\partial^2 u(x_0, t_0)}{\partial t^2} &= -\varphi_{04}u_0 + \sum_{i=1}^s \varphi_{i4}u_i + \mathcal{O}(h_i^2, k_i^2), \text{ with } \varphi_{04} = \sum_{i=1}^s \varphi_{i4}, \\ \frac{\partial^2 u(x_0, t_0)}{\partial x \partial t} &= -\varphi_{05}u_0 + \sum_{i=1}^s \varphi_{i5}u_i + \mathcal{O}(h_i^2, k_i^2), \text{ with } \varphi_{05} = \sum_{i=1}^s \varphi_{i5}, \end{aligned} \right. \tag{7}$$

or, written in vectorial form,

$$\mathbf{\Pi}_5 u(x_0, t_0) = -\boldsymbol{\varphi}_0 u_0 + \sum_{i=1}^s \boldsymbol{\varphi}_i u_i + \mathcal{O}(h_i^2, k_i^2)$$

where $\boldsymbol{\varphi}_0$ and $\boldsymbol{\varphi}_i$ stand for

$$\begin{aligned} \boldsymbol{\varphi}_0 &= \{\varphi_{01}, \varphi_{02}, \varphi_{03}, \varphi_{04}, \varphi_{05}\}^T, \\ \boldsymbol{\varphi}_i &= \{\varphi_{i1}, \varphi_{i2}, \varphi_{i3}, \varphi_{i4}, \varphi_{i5}\}^T, \end{aligned}$$

fulfilling

$$\boldsymbol{\varphi}_0 = \sum_{i=1}^s \boldsymbol{\varphi}_i.$$

Since the method allows us to choose different numbers of points per star, greater or equal to 6, and the selection of these nodes can be made by different criteria (distance, quadrant, octant), in order to obtain well-conditioned stars, the STCM allows us to solve the problem without the requirement of adding further conditions.

Similarly for the 3D case, the approximation derivatives are:

$$\left\{ \begin{aligned} \frac{\partial u(x_0, y_0, t_0)}{\partial x} &= -\varphi_{01}u_0 + \sum_{i=1}^s \varphi_{i1}u_i + \mathcal{O}(h_i^2, k_i^2, \tau_i^2), \\ \frac{\partial u(x_0, y_0, t_0)}{\partial y} &= -\varphi_{02}u_0 + \sum_{i=1}^s \varphi_{i2}u_i + \mathcal{O}(h_i^2, k_i^2, \tau_i^2), \\ \frac{\partial u(x_0, y_0, t_0)}{\partial t} &= -\varphi_{03}u_0 + \sum_{i=1}^s \varphi_{i3}u_i + \mathcal{O}(h_i^2, k_i^2, \tau_i^2), \\ \frac{\partial^2 u(x_0, y_0, t_0)}{\partial x^2} &= -\varphi_{04}u_0 + \sum_{i=1}^s \varphi_{i4}u_i + \mathcal{O}(h_i^2, k_i^2, \tau_i^2), \\ \frac{\partial^2 u(x_0, y_0, t_0)}{\partial y^2} &= -\varphi_{05}u_0 + \sum_{i=1}^s \varphi_{i5}u_i + \mathcal{O}(h_i^2, k_i^2, \tau_i^2), \\ \frac{\partial^2 u(x_0, y_0, t_0)}{\partial t^2} &= -\varphi_{06}u_0 + \sum_{i=1}^s \varphi_{i6}u_i + \mathcal{O}(h_i^2, k_i^2, \tau_i^2), \\ \frac{\partial^2 u(x_0, y_0, t_0)}{\partial x \partial y} &= -\varphi_{07}u_0 + \sum_{i=1}^s \varphi_{i7}u_i + \mathcal{O}(h_i^2, k_i^2, \tau_i^2), \\ \frac{\partial^2 u(x_0, y_0, t_0)}{\partial x \partial t} &= -\varphi_{08}u_0 + \sum_{i=1}^s \varphi_{i8}u_i + \mathcal{O}(h_i^2, k_i^2, \tau_i^2), \\ \frac{\partial^2 u(x_0, y_0, t_0)}{\partial y \partial t} &= -\varphi_{09}u_0 + \sum_{i=1}^s \varphi_{i9}u_i + \mathcal{O}(h_i^2, k_i^2, \tau_i^2), \end{aligned} \right. \tag{8}$$

or, written in the vectorial form,

$$\mathbf{\Pi}_9 u(x_0, y_0, t_0) = -\boldsymbol{\varphi}_0 u_0 + \sum_{i=1}^s \boldsymbol{\varphi}_i u_i + \mathcal{O}(h_i^2, k_i^2, \tau_i^2),$$

where $\boldsymbol{\varphi}_0$ and $\boldsymbol{\varphi}_i$ stand for

$$\begin{aligned} \boldsymbol{\varphi}_0 &= \{\varphi_{01}, \varphi_{02}, \varphi_{03}, \varphi_{04}, \varphi_{05}, \varphi_{06}, \varphi_{07}, \varphi_{08}, \varphi_{09}\}^T, \\ \boldsymbol{\varphi}_i &= \{\varphi_{i1}, \varphi_{i2}, \varphi_{i3}, \varphi_{i4}, \varphi_{i5}, \varphi_{i6}, \varphi_{i7}, \varphi_{i8}, \varphi_{i9}\}^T, \end{aligned}$$

fulfilling

$$\boldsymbol{\varphi}_0 = \sum_{i=1}^s \boldsymbol{\varphi}_i.$$

For the 3D case, the number of nodes per star should be greater or equal to 10 in order to obtain well-conditioned stars.

Remark 1. The consistency of the GFD formulae was proved in [20] for the 2D case (which in our case is $\Gamma \subset \mathbb{R}$ and $\Gamma^* = \Gamma \times [0, T]$) and for the 3D case (for our case, $\Gamma \subset \mathbb{R}^2$ and $\Gamma^* = \Gamma \times [0, T]$) in [21]. Moreover, the rate of convergence is quadratic, which is clear from the cited papers.

3. Numerical Results

In the present section, we give several numerical results obtained by solving the parabolic equations in 2D and 3D using STCM. In order to show the applicability of the present method, we compare the present method with the results obtained previously in the literature. In the following numerical examples, the distance criterion has been used, the number of nodes per star is eight plus the central node and the weighting function is the inverse of the distance cubed, $\eta(d_i) = \frac{1}{d_i^3}$ with d_i being the distance from node i to the central node of the star. More precisely, we compare the errors and execution times of the

STCM and the GFDM. In all the cases under consideration, the accuracy of the STCM and the GFDM are computed using the expressions:

$$\begin{cases} l_2 = \left(\frac{\sum_{i=1}^Z (u_i - U_i)^2}{Z} \right)^{\frac{1}{2}}, \\ l_\infty = \max |u_i - U_i|, \end{cases} \tag{9}$$

where Z is the number of points in $(\Gamma \setminus \partial\Gamma) \times (0, T]$.

3.1. 2D Problems

For the two-dimensional examples (one space variable and one temporal variable), we use the clouds of points of Figures 3 and 4.

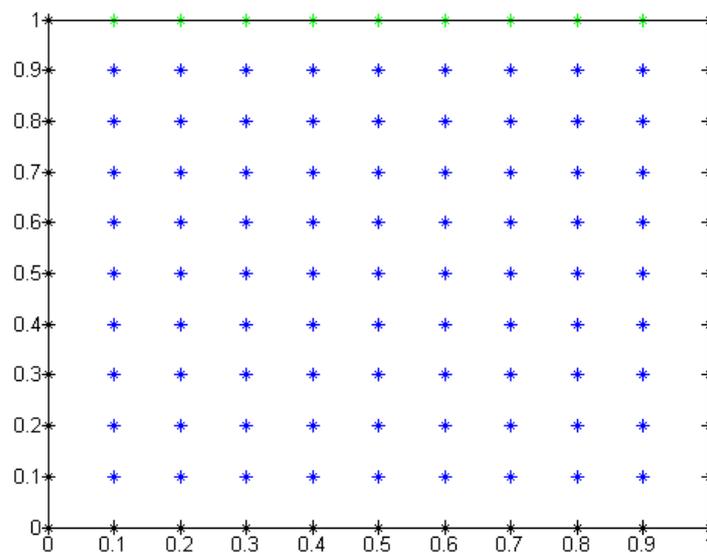


Figure 3. First cloud of points for the 2D examples (Cloud 1).

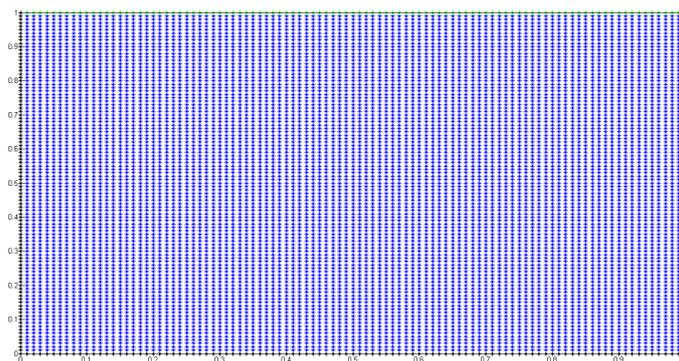


Figure 4. Second cloud of points for the 2D examples (Cloud 2).

3.1.1. Example 1

We consider the equation

$$\pi^2 \frac{\partial u(x, t)}{\partial t} - \frac{\partial^2 u(x, t)}{\partial x^2} = e^{-t} [-\pi^2 x(1 - x) + 2] \tag{10}$$

with initial and boundary conditions:

$$u(x, 0) = \sin(\pi x) + x(1 - x); \quad u(0, t) = u(1, t) = 0 \tag{11}$$

and whose exact solution is

$$u(x, t) = e^{-t}[\sin(\pi x) + x(1 - x)]. \tag{12}$$

Table 1 shows the errors and execution times for the resolution of Example 1 in the clouds of Figures 3 and 4, respectively, using STCM and GFDM explicitly (with step size $\Delta t = 0.001$ s for cloud 1 and $\Delta t = 0.00001$ for cloud 2). It can be seen that the error norms are very similar and the execution times of the STCM are slightly higher.

Table 1. Errors and execution time of the Example 1 in the clouds of Figures 3 and 4.

Cloud 1	l_2	l_∞	$t(s)$
STCM	0.000902	0.001637	0.183955
Explicit-GFDM	0.000813	0.001398	0.029527 s
Cloud 2	l_2	l_∞	$t(s)$
STCM	0.000248	0.000573	20.2965 s
Explicit-GFDM	0.000277	0.000401	11.723 s

3.1.2. Example 2

For our second 2D examples, we choose

$$\frac{\partial u(x, t)}{\partial t} - \frac{\partial u(x, t)}{\partial x} - x \frac{\partial^2 u(x, t)}{\partial x^2} = e^{-t}[(x^2 - x - 1) \sin x - 3 \cos x] \tag{13}$$

together with

$$u(x, 0) = x \sin(x); \quad u(0, t) = 0; \quad u(1, t) = e^{-t} \sin(1). \tag{14}$$

The exact solution of problem (13) and (14) is

$$u(x, t) = e^{-t}[x \sin(x)]. \tag{15}$$

The error norms, run times, and a comparison between the STCM and the explicit GFDM (step sizes are the same as in the previous example as the meshes are the same), for this second example in the clouds of points of Figures 3 and 4, are depicted in Table 2. We observe that the errors are rather similar and the execution times are smaller for the STCM for a great number of points.

Table 2. Errors and execution time of the Example 2 in the clouds of Figures 3 and 4.

Cloud 1	l_2	l_∞	$t(s)$
STCM	0.063348	0.084077	0.10168 s
Explicit-GFDM	0.059003	0.080338	0.096604 s
Cloud 2	l_2	l_∞	$t(s)$
STCM	0.009618	0.013204	26.145s
Explicit-GFDM	0.009787	0.017510	44.409 s

It is clear from the examples that, as the number of nodes increases, the execution times are shorter in the STCM than in the explicit GFDM, which is logical, since, as the time

step decreases (in order to guarantee the stability conditions), the explicit GFDM has to make more iterations.

3.2. 3D Problems

To continue our study, we provide examples in the three-dimensional setting. We use both regular and irregular clouds of points, seen in Figures 5–8. The black points denote the boundary nodes, the blue ones denote the interior nodes and the green points denote the $\partial\Gamma \times \{t = T\}$ boundary nodes. In the following numerical examples, the distance criterion has been used, the number of nodes per star is 24 plus the central node and the weighting function is the inverse of the distance squared, $\eta(d_i) = \frac{1}{d_i^2}$ with d_i being the distance from node i to the central node of the star.

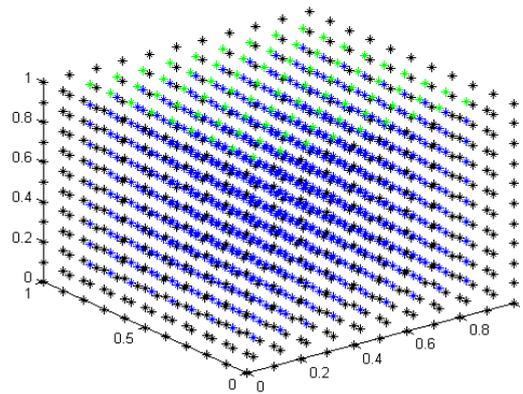


Figure 5. First regular cloud of points for the 3D examples (Cloud 3).

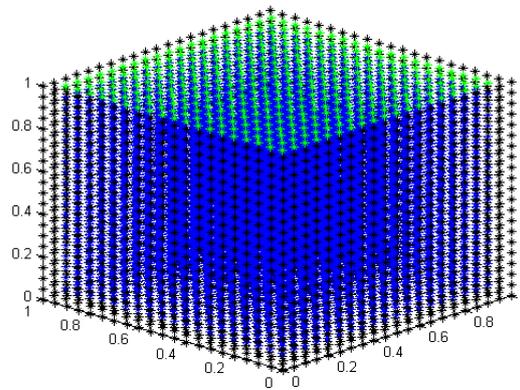


Figure 6. Second regular cloud of points for the 3D examples (Cloud 4).

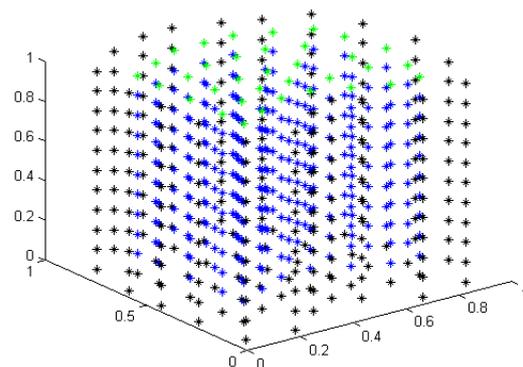


Figure 7. First irregular cloud of points for the 3D examples (Cloud 5).

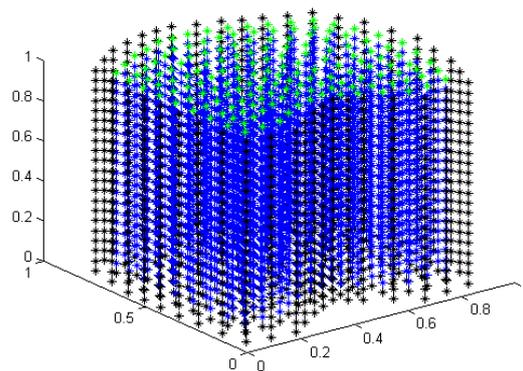


Figure 8. Second irregular cloud of points for the 3D examples (Cloud 6).

3.2.1. Example 3

Consider the following equation

$$2\pi^2 \frac{\partial u(x, y, t)}{\partial t} - \frac{\partial^2 u(x, y, t)}{\partial x^2} - \frac{\partial^2 u(x, y, t)}{\partial y^2} = 2x(1-x)(1-y) \tag{16}$$

with the initial condition

$$u(x, y, 0) = \sin(\pi x) \sin(\pi y) + x(1-x)(1-y). \tag{17}$$

By a direct check, the exact solution is (the Dirichlet boundary conditions are chosen so that the equation is fulfilled)

$$u(x, y, t) = e^{-t}[\sin(\pi x) \sin(\pi y) + x(1-x)(1-y)] \tag{18}$$

We depict in Table 3 the error norms and run times for this third example. We found that for similar errors, the STCM consumes slightly less time than the explicit GFDM (where we use $\Delta t = 0.001$ s for cloud 3, $\Delta t = 0.00001$ s for cloud 4, $\Delta t = 0.05$ s for cloud 5 and $\Delta t = 0.0025$ s for cloud 6). For this third example, we have also compared the STCM with the implicit scheme given by the GFDM formulae (with step size $\Delta t = 0.1$ s for cloud 5 and $\Delta t = 0.05$ s for cloud 6). In this case, the discretisation of the derivatives is

$$\frac{\partial u}{\partial t} = \frac{u_0^{j+1} - u_0^j}{2\Delta t}; \quad \frac{\partial^2 u}{\partial x^2} = -\varphi_{04}u_0^{j+1} + \sum_{i=1}^s \varphi_{i4}u_i^{j+1}; \quad \frac{\partial^2 u}{\partial y^2} = -\varphi_{05}u_0^{j+1} + \sum_{i=1}^s \varphi_{i5}u_i^{j+1}$$

3.2.2. Example 4

Finally, we solve the equation

$$\begin{aligned} &\frac{\partial u(x, y, t)}{\partial t} - \frac{\partial u(x, y, t)}{\partial x} - \frac{\partial u(x, y, t)}{\partial y} - x \frac{\partial^2 u(x, y, t)}{\partial x^2} - y \frac{\partial^2 u(x, y, t)}{\partial y^2} = \\ &= e^{-t}[(x^2 - x - 1) \sin x - 3x \cos x]y \sin y + ((y^2 - y - 1) \sin y - 3y \cos y)x \sin x, \end{aligned} \tag{19}$$

with initial data

$$u(x, y, 0) = xy \sin(x) \sin(y), \tag{20}$$

and exact solution

$$u(x, y, t) = e^{-t}[xy \sin(x) \sin(y)]. \tag{21}$$

Again, in Table 4, we can observe that, even though the errors are rather similar for both methods, the STCM achieves better execution times.

Table 3. Errors and execution time of the Example 3 in the clouds of Figures 5–8.

Cloud 3	l_2	l_∞	$t(s)$
STCM	0.016509	0.032567	0.047252 s
Explicit-GFDM	0.015263	0.025234	0.04283 s
Cloud 4	l_2	l_∞	$t(s)$
STCM	0.000806	0.006275	2.1954 s
Explicit-GFDM	0.000822	0.001541	2.3099 s
Cloud 5	l_2	l_∞	$t(s)$
STCM	0.003672	0.005581	0.010077 s
Explicit-GFDM	0.003923	0.005818	0.007449 s
Implicit-GFDM	0.003804	0.005742	0.01121 s
Cloud 6	l_2	l_∞	$t(s)$
STCM	0.000816	0.001623	0.18481s
Explicit-GFDM	0.000733	0.001283	0.26016 s
Implicit-GFDM	0.000831	0.002432	0.19145 s

Table 4. Errors and execution time of Example 4 in the clouds of Figures 5–8.

Cloud 3	l_2	l_∞	$t(s)$
STCM	0.004338	0.008377	0.87247 s
Explicit-GFDM	0.004098	0.007860	1.1949 s
Cloud 4	l_2	l_∞	$t(s)$
STCM	0.004136	0.006394	3.2172 s
Explicit-GFDM	0.004166	0.006567	3.4186 s
Cloud 5	l_2	l_∞	$t(s)$
STCM	0.003383	0.005183	0.35261 s
Explicit-GFDM	0.002923	0.004987	0.5059 s
Cloud 6	l_2	l_∞	$t(s)$
STCM	0.000871	0.001276	1.94437 s
Explicit-GFDM	0.000931	0.001905	2.2854 s

4. Conclusions

The STCM allows the solution of the numerically parabolic PDEs in an efficient way without the need for any additional condition to the well-defined problem (initial and boundary conditions). In the various examples treated, both in 2D and 3D, linear and non-linear, in regular and irregular domains, the numerical results show that the errors are similar to those obtained with the explicit GFDM. The run times are also similar, but the numerical examples show that the time consumption of the STCM is smaller if the number of points in the computational domain is high.

The potential of the STCM consists of the elimination of the time step constraint forced by the stability of the explicit method, which is a real advance in the resolution of linear and non-linear parabolic PDEs.

Author Contributions: Formal analysis, A.M.V.; Methodology, F.U.; Project administration, J.J.B.; Resources, J.J.B.; Software, Á.G.; Writing—original draft, M.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors acknowledge the support of the Escuela Técnica Superior de Ingenieros Industriales (UNED) of Spain, project 2021-IFC02. This work is also partially supported by the Project MTM2017-42907-P from MICINN (Spain).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, S.; Liu, W.K. *Meshfree Particle Methods*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2007.
2. Nguyen, V.P.; Rabczuk, T.; Bordas, S.; Duflo, M. Meshless methods: A review and computer implementation aspects. *Math. Comput. Simul.* **2008**, *79*, 763–813. [[CrossRef](#)]
3. Huang, T.H.; Wei, H.; Chen, J.S.; Hillman, M.C. RKPM2D: An open-source implementation of nodally integrated reproducing kernel particle method for solving partial differential equations. *Comput. Part. Mech.* **2020**, *7*, 393–433. [[CrossRef](#)]
4. Zhang, C.; Rezavand, M.; Zhu, Y.; Yu, Y.; Wu, D.; Zhang, W.; Wang, J.; Hu, X. SPHinXsys: An open-source multi-physics and multi-resolution library based on smoothed particle hydrodynamics. *Comput. Phys. Commun.* **2021**, *267*, 108066. [[CrossRef](#)]
5. Jensen, P.S. Finite difference technique for variable grids. *Comput. Struct.* **1972**, *2*, 17–29. [[CrossRef](#)]
6. Liszka, T.; Orkisz, J. The finite difference method at arbitrary irregular grids and its application in applied mechanics. *Comput. Struct.* **1980**, *11*, 83–95. [[CrossRef](#)]
7. Orkisz, J. Finite difference method (Part, III). In *Handbook of Computational Solid Mechanics*; Kleiber, M., Ed.; Springer: Berlin/Heidelberg, Germany, 1998.
8. Perrone, N.; Kao, R. A general finite difference method for arbitrary meshes. *Comput. Struct.* **1975**, *5*, 45–58. [[CrossRef](#)]
9. Benito, J.J.; Ureña, F.; Gavete, L. Influence of several factors in the generalized finite difference method. *Appl. Math. Model.* **2001**, *25*, 1039–1053. [[CrossRef](#)]
10. Albuquerque-Ferreira, A.C.; Ureña, M.; Ramos, H. The generalized finite difference method with third-and fourth-order approximations and treatment of ill-conditioned stars. *Eng. Anal. Bound. Elem.* **2021**, *127*, 29–39. [[CrossRef](#)]
11. Uddin, M.; Ali, H. The space-time kernel-based numerical method for Burgers' equations. *Mathematics* **2018**, *6*, 212. [[CrossRef](#)]
12. Sophy, T.; Silva, A.D.; Kribèche, A.; Sadat, H. An alternative space-time meshless method for solving transient heat transfer problems with high discontinuous moving sources. *Numer. Heat Transf. Part B Fundam.* **2016**, *69*, 377–388. [[CrossRef](#)]
13. Ku, C.-Y.; Liu, C.-Y.; Yeh, W.; Liu, C.-S.; Fan, C.-M. A novel space-time meshless method for solving the backward heat conduction problem. *Int. J. Heat Mass Transf.* **2019**, *130*, 109–122. [[CrossRef](#)]
14. Hamaidi, M.; Naji, A.; Charafi, A. Space-time localized radial basis function collocation method for solving parabolic and hyperbolic equations. *Eng. Anal. Bound. Elem.* **2016**, *67*, 152–163. [[CrossRef](#)]
15. Li, Z.; Mao, X.Z. Global space-time multiquadric method for inverse heat conduction problem. *Int. J. Numer. Methods Eng.* **2011**, *85*, 355–379. [[CrossRef](#)]
16. Lei, J.; Wei, X.; Wang, Q.; Gu, Y.; Fan, C.-M. A novel space-time generalized FDM for dynamic coupled thermoelasticity problems in heterogeneous plates. *Arch. Appl. Mech.* **2022**, *92*, 287–307. [[CrossRef](#)]
17. Li, P.-W. Space-time generalized finite difference nonlinear model for solving unsteady Burgers' equations. *Appl. Math. Lett.* **2021**, *114*, 106896. [[CrossRef](#)]
18. Qu, W.; He, H. A spatial-temporal GFDM with an additional condition for transient heat conduction analysis of FGMs. *Appl. Math. Lett.* **2020**, *110*, 106579. [[CrossRef](#)]
19. Ureña, F.; Gavete, L.; Garcia, A.; Benito, J.J.; Vargas, A.M. Solving second order non-linear parabolic PDEs using generalized finite difference method (GFDM). *J. Comput. Appl. Math.* **2019**, *354*, 221–241. [[CrossRef](#)]
20. Gavete, L.; Ureña, F.; Benito, J.J.; Garcia, A.; Ureña, M.; Salet, E. Solving second order non-linear elliptic partial differential equations using generalized finite difference method. *J. Comput. Appl. Math.* **2017**, *318*, 378–387. [[CrossRef](#)]
21. Ureña, F.; Gavete, L.; Garcia, A.; Benito, J.J.; Vargas, A.M. Non-linear Fokker-Planck equation solved with generalized finite differences in 2D and 3D. *Appl. Math. Comput.* **2020**, *368*, 124801. [[CrossRef](#)]