frontiers in Physics

ORIGINAL RESEARCH
published: 09 September 2020
doi: 10.3389/fphy.2020.00367

ESERK Methods to Numerically Solve Nonlinear Parabolic PDEs in Complex Geometries: Using Right Triangles

*Jesús Martín-Vaquero**

Department on Applied Mathematics, Instituto Universitario de Física Fundamental y Matemáticas, Universidad de Salamanca, Salamanca, Spain

In this paper Extrapolated Stabilized Explicit Runge-Kutta methods (ESERK) are proposed to solve nonlinear partial differential equations (PDEs) in right triangles. These algorithms evaluate more times the function than a standard explicit Runge–Kutta scheme (n_t times per step), and these extra evaluations do not increase the order of convergence but the stability region grows with $\mathcal{O}(n_t^2)$. Hence, the total computational cost is $\mathcal{O}(n_t)$ times lower than with a traditional explicit algorithm. Thus, these algorithms have been traditionally considered to solve stiff PDEs in squares/rectangles or cubes. In this paper, for the first time, ESERK methods are considered in a right triangle. It is demonstrated that such type of codes keep the convergence and the stability properties under certain conditions. This new approach would allow to solve nonlinear parabolic PDEs with stabilized explicit Runge–Kutta schemes in complex domains, that would be decomposed in rectangles and right triangles.

Keywords: complex geometries, higher-order codes, multi-dimensional partial differential equations, nonlinear PDEs, Stabilized Explicit Runge-Kutta methods

OPEN ACCESS

Edited by:
Poom Kumam,
King Mongkut's University of Technology Thonburi, Thailand

Reviewed by:
Abdullahi Yusuf,
Federal University Dutse, Nigeria
Kazuharu Bamba,
Fukushima University, Japan

****Correspondence:***
Jesús Martín-Vaquero
jesmarva@usal.es

Specialty section:
This article was submitted to Mathematical and Statistical Physics, a section of the journal Frontiers in Physics

Received: *15 February 2020*
Accepted: *29 July 2020*
Published: *09 September 2020*

Citation:
Martín-Vaquero J (2020) ESERK Methods to Numerically Solve Nonlinear Parabolic PDEs in Complex Geometries: Using Right Triangles. Front. Phys. 8:367. doi: 10.3389/fphy.2020.00367

1. INTRODUCTION

Let us suppose that we have to solve a nonlinear PDE with dominating diffusion:

$$u_t = d(u_{\bar{x}\bar{x}} + u_{\bar{y}\bar{y}}) + f(t, \bar{x}, \bar{y}, u) \qquad (\bar{x}, \bar{y}) \in \Omega \subset \mathbb{R}^2, \tag{1}$$

subject to traditional initial and Dirichlet boundary conditions:

$$u(0, \bar{x}, \bar{y}) = g_1(\bar{x}, \bar{y}) \qquad (\bar{x}, \bar{y}) \in \Omega, \tag{2}$$

and

$$u(t, \bar{x}, \bar{y}) = g_2(\bar{x}, \bar{y}) \qquad (\bar{x}, \bar{y}) \in \partial(\Omega). \tag{3}$$

These types of problems are very common in a large amount of areas such as atmospheric phenomena, biology, chemical reactions, combustion, financial mathematics, industrial engineering, laser modeling, malware propagation, medicine, mechanics, molecular dynamics, nuclear kinetics, etc., see [1–9], to mention a few.

A widely-used approach for solving these time-dependent and multi-dimensional PDEs is to first discretize the space variables (with finite difference or spectral methods) to obtain a very large system of ODEs of the form

$$y' = f(t, y); \qquad y(t_0) = y_0; \tag{4}$$

where $y, y_0 \in \mathbb{R}^n$, $t \geq t_0$, and $f(t, y)$ takes value in $\mathbb{R}^n$, this procedure is well-known as the method of lines (MOL). But these systems of ODEs not only have a huge dimension, additionally they might become very stiff problems.

Hence, traditional explicit methods are usually very slow, due to absolute stability, it is necessary to use very small length steps, see [6, 7] and references therein. Therefore, these schemes are not usually considered.

Implicit schemes based on BDF and Runge–Kutta methods have much better stability properties. However, since the dimension of the ODE system is very high, then it is necessary to solve very large nonlinear systems at each iteration.

Other numerous techniques have also been proposed based on ETD schemes (but it is necessary to approximate operators including matrix exponentials), alternating direction implicit methods (they have limitations on the order of convergence) and explicit-implicit algorithms. However, in any case the number of operations is huge when the system dimension is high.

For those cases where it is known that the Jacobian eigenvalues of the function are all real negative or are very close to this semi-axis, there is another option: stabilized explicit Runge–Kutta methods (they are also called Runge-Kutta-Chebyshev methods). This happens, for example, when diffusion dominates in the PDE, when the Laplacian is discretized using finite differences or some spectral techniques, then the associated matrix has this type of eigenvalues.

These types of algorithms are totally explicit, but they have regions of stability extended along the real negative axis. These schemes typically have order 2 or 4 [8, 10–16]. Recently, we propose a new procedure combined with Richardson extrapolation to obtain methods with other orders of convergence [17, 18], but in all these methods, these integrators have many more stages than the order of convergence. Most of these extra stages seek to extend as much as possible the region of stability along the negative real axis. Regions of stability increase quadratically with the number of stages. Thus, the cost per step is greater than in a classic Runge–Kutta, because it is necessary to evaluate the function in Equation (4) n_t times. However, the number of steps reduce proportionally with n_t^2, thus the total computational cost is reduced proportionally with n_t.

These schemes have been traditionally considered in squares/rectangles or cubes. But this makes difficult to apply them in PDEs with complex geometries, which happens in most of the cases. Some different strategies have been proposed to apply them when the original domain is not a square nor a cube (see [3–5]). They implemented stabilized Runge–Kutta methods after using adaptive multiresolution techniques or fixed mesh codes in space. But simulations in complex geometries constitute a very challenging problem, see (section 4, [5]), where they stated for their results based on adaptive multiresolution techniques that they "will only present here 2D and 3D simulations in simplified geometries for the sake of assessing our results and perspectives in the field."

As far as we know, stabilized explicit Runge–Kutta methods have not been tested in triangles yet. For this reason, in this paper, we are analysing how ESERK methods can be employed to solve nonlinear PDEs in these types of regions and their convergence.

In this paper, a summary on ESERK4 methods is provided in section 2. The major advance of our contribution is given in section 3: it is explained how ESERK4 can be utilized for (1) when Ω is a right triangle. After some linear transformations and spatial discretizations ESERK4 is numerically stable and fourth-order convergence in time, and second-order in space is obtained. This allows a new way to numerically approach parabolic nonlinear PDEs in complex domains in the plane, which can be easier decomposed in a sum of triangles and rectangles. Finally, some conclusions and future goals are outlined.

2. ESERK4 METHODS

2.1. Construction of First-Order Stabilized Explicit Methods

In [17], ESERK4 schemes were developed for nonlinear PDEs in several dimensions with good stability properties and numerical results in squares and cubes. The idea is quite simple: first-order stabilized explicit Runge–Kutta (SERK) methods are derived using Chebyshev polynomials of the first kind:

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_s(x) = 2xT_{s-1}(x) - T_{s-2}(x), \tag{5}$$

s being the number of stages of the first-order method.

If we consider

$$R_s(z) = \frac{T_s(w_{0,s} + w_{1,s}z)}{T_s(w_{0,s})}, \quad w_{0,s} = 1 + \frac{\mu_4}{s^2}, \quad w_{1,s} = \frac{T_s(w_{0,s})}{T_s'(w_{0,s})}, \tag{6}$$

then $|R_s(z)|$ oscillates between $-\lambda_4$ and λ_4 (for a value $0 < \lambda_4 = 0.311688 < 1$ that we will calculate later) in a region which is $O(s^2)$, and $R_s(z) = 1 + z + O(z^2)$.

We can construct Runge–Kutta schemes with $|R_s(z)|$ as stability functions by just changing $x = 1 + \alpha_p \bar{x}$ (and considering that 1, $T_s(x)$ and $\bar{x}$ are the stability functions of Identity operator, g_s, and $hf(\cdot)$, and writing $R_s(z)$ as a linear combination of the Chebyshev polynomials, see Theorem 1 [12] for more details).

2.2. Construction of Higher-Order ESERK Schemes

Once first-order SERK methods have been derived, they approximate the solution of the initial value problem (4), by performing n_i steps with constant step size h_i at $x_0 + h$, i.e., $y_{h_i}(x_0 + h) := S_{i,1}$, with step sizes $h_1 > h_2 > h_3 > \ldots$ (taking $h_i = h/n_i$, $n_i = 1, \ldots, 4$).

Finally

$$\begin{aligned} S_{4,4} &= \frac{-S_{1,1} + 24S_{2,1} - 81S_{3,1} + 64S_{4,1}}{6} = \\ &= \frac{64y_{h/4}(x_0 + h) - 81y_{h/3}(x_0 + h) + 24y_{h/2}(x_0 + h) - y_h(x_0 + h)}{6} \end{aligned} \tag{7}$$

is a fourth-order approximation with

$$P_{4s}(z) = \frac{-R_s(z)+24(R_s(z/2))^2-81(R_s(z/3))^3+64(R_s(z/4))^4}{6}. \tag{8}$$

as its stability function. Additionally, we have that

$$|P_{4s}(z)| \leq \frac{|R_s(z)| + 24|R_s(z/2)|^2 + 81|R_s(z/3)|^3 + 64|R_s(z/4)|^4}{6}.$$

The positive real solution of

$$\frac{x + 24x^2 + 81x^3 + 64x^4}{6} = 0.95$$

is $\lambda_4 = 0.311688$. Hence, whenever $|R_s(z)| < 0.311688$, then $|P_{4s}(z)| < 0.95$. Taking $\mu_4 = \frac{27}{16}$, it can be checked numerically that $|R_{s,4}(z)| \leq 0.311688$ for $z \in [-s^2, -1]$ and $9 \leq s \leq 4000$, and therefore the ESERK4 methods derived in this way are fourth-order approximations and numerically stable in a region including $[-s^2, 0]$.

2.3. Parallel, Variable-Step, and Number of Stages Algorithm

In [17], we constructed a variable-step and number of stages algorithm combining all the schemes derived there, with s up to 4,000. The idea is quite simple: (i) First, we select the step size in order to control the local error; the best results were obtained using techniques considered for standard extrapolation methods (see Equations (8–11) in [17]). (ii) Later the minimum s is chosen so that the absolute stability is satisfied.

Recently, we are working developing the parallel version of this code (see [19]). Using 4 threads, CPU times are up to 2.5 times smaller than in the previous sequential algorithm. The new parallel code also has a decreasing memory demand, and therefore it is possible to solve problems with higher dimension in regular PCs.

3. DECOMPOSITION OF COMPLEX GEOMETRIES INTO RIGHT TRIANGLES

Complex geometric shapes are ubiquitous in our natural environment. In this paper, we are interested in numerically solving partial differential equations (PDEs) in such types of geometries, which are very common in problems related with human bodies, materials, or simply a complicated engine in classical engineering applications.

One very well-known strategy, within a finite element context, is to build the necessary modifications in the vicinity of the boundary. Such an approach is studied in the composite finite element method (FEM). Those methods based on finite element are usually proposed only for linear PDEs. FEM is a numerical method for solving problems of engineering and mathematical physics (typical problems include structural analysis, heat transfer, fluid flow, mass transport, or electromagnetic potential, because these problems generally require numerically approximating the solution of linear partial differential equations). The finite element method allows the transformation of the problem in a system of algebraic equations. Unfortunately, it is more difficult to employ these techniques with nonlinear parabolic PDEs in several dimensions, although some results have been obtained to know when the resulting discrete Galerkin equations have a unique solution in [20]. However, for some problems, some of these techniques are not easy to be employed numerically, they are computationally very expensive because they require solving nonlinear systems with huge dimension at every step, or it is difficult to demonstrate that the numerical schemes have unique solution in a general case.

On the other hand, Implicit–Explicit (IMEX) methods have been employed to solve a very stiff nonlinear system of ODEs coming from the spatial discretization of nonlinear parabolic PDEs that appeared in the modelization of an ischemic stroke in [5]. The authors employed an adaptive multiresolution approach and a finite volume strategy for the spatial discretizations. And a Strang splitting method in time, combining ROCK4, an explicit Stalized Explicit Runge–Kutta scheme for the diffusion part, and Radau5, an implicit A-stable method for the reaction. These methods were previously analyzed in [3] for streamer discharge simulations, and the authors demonstrated second-order convergence in time. Later, they employed similar strategies for different physical problems in [4, 21]. As the authors state, some of these procedures are complicated except in simple domains like squares and cubes, and only second-order convergence in time is possible. However, there are complex problems where nonlinear terms have potentially large stiffness, and at the same time, it is necessary to efficiently solve the model with small errors. This motivates to derive high-order schemes with good internal stability properties.

In what follows we will explain a new strategy to numerically solve the nonlinear parabolic PDE given by Equation (1) where Ω is any right triangle, and therefore any researcher can combine the theory (utilized with FEM) to spatially decomposed any complex geometry into triangles (since any acute triangle and obtuse triangle can be decomposed into two right triangles), and later employing the method described in this paper. Additionally, schemes proposed in this work are fourth-order ODE solvers (in time), and numerical spatial approximations will be second-order (although fourth-order formulae can be explored except for the closest points to vertices).

3.1. Higher-Order Spatial Approximations in the Triangle

Without loss of generality we can consider that our right triangle is T_R, the one with vertices $(0, 1), (0, 0), (1, 0)$. Otherwise we first use a linear transformation of the right triangle with vertices $P_1 = (\bar{x}_1, \bar{y}_1)$, $P_0 = (\bar{x}_0, \bar{y}_0)$, $P_2 = (\bar{x}_2, \bar{y}_2)$ [where $(\bar{x}_0, \bar{y}_0)$ is the vertex of the right angle]:

$$(x, y) = L(\bar{x}, \bar{y}) = (a_1(\bar{x}-\bar{x}_0)+a_2(\bar{y}-\bar{y}_0), b_1(\bar{x}-\bar{x}_0)+b_2(\bar{y}-\bar{y}_0)), \tag{9}$$

where the parameters a_1, a_2, b_1, b_2 can be computed easily as

$$\begin{aligned} a_1 &= \tfrac{\bar{y}_2-\bar{y}_0}{\mathrm{Det}}, \quad & a_2 &= \tfrac{\bar{x}_0-\bar{x}_2}{\mathrm{Det}}, \\ b_1 &= \tfrac{\bar{y}_0-\bar{y}_1}{\mathrm{Det}}, \quad & a_2 &= \tfrac{\bar{x}_0-\bar{x}_1}{\mathrm{Det}}, \end{aligned} \tag{10}$$

where

$$\mathrm{Det} = \begin{vmatrix} 1 & 1 & 1 \\ \bar{x}_0 & \bar{x}_1 & \bar{x}_2 \\ \bar{y}_0 & \bar{y}_1 & \bar{y}_2 \end{vmatrix} \tag{11}$$

and it is easy to check that Det $\neq 0$ if and only the three points are not in a line (but we always have a triangle).

The main reason of decomposing our general region $\Omega \in R^2$ into right triangles (and not other triangles) is, that after this linear transformation given by Equations (9) and (10), our PDE given by Equation (1) transforms into the Equation

$$u_t = c_1 u_{xx} + c_2 u_{yy} + f(t, x, y, u) \qquad (x, y) \in T_R, \tag{12}$$

subject to (traditional) initial and Dirichlet boundary conditions. Therefore, let us first study Equation (12), together with

$$u(0, x, y) = g_1(x, y) \qquad (x, y) \in T_R, \tag{13}$$

and

$$u(t, x, y) = g_2(x, y) \qquad (x, y) \in \partial(T_R), \tag{14}$$

where $\partial(T_R)$ is the border of the triangle with vertices $(0, 1), (0, 0), (1, 0)$. One positive issue is that, after the traditional spatial discretization described below, the matrix obtained from the diffusion term has all the eigenvalues real, and therefore we can utilize the ESERK methods proposed in the previous section 2.

Now, let us define the spatial discretization of our continuous problem provided by Equation (12), the problem domain T_R is discretized by the grid points (x_i, y_j), where

$$\begin{aligned} &x_i = i \cdot h, \quad i = 0, 1, \ldots, N, \quad N = \frac{1}{h}, \qquad y_j = j \cdot h, \\ &j = 0, 1, \ldots, N - i, \quad h = \Delta x = \Delta y, \end{aligned} \tag{15}$$

since $x_i + y_j \leq 1$.

With this semidiscretizations we will approximate u_{xx} and u_{yy} at point (x_i, y_j) with the following second-order formulae:

$$\frac{\partial^2 u_{i,j}}{\partial x^2} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2}, \qquad \frac{\partial^2 u_{i,j}}{\partial y^2} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2}. \tag{16}$$

After the linear transformation given by Equations (9) and (10), our PDE given by Equation (1) may transform into one Equation where one term in u_{xy} would appear. Normally, this term can be approximated in the square or the rectangle through the formula

$$\begin{aligned} &\frac{\partial^2 u_{i,j}}{\partial x \partial y} \\ &= \frac{u_{i+1,j+1} - u_{i,j+1} - u_{i+1,j} + 2u_{i,j} - u_{i-1,j} - u_{i,j-1} + u_{i-1,j-1}}{2h^2}, \end{aligned} \tag{17}$$

however, in T_R, we can obtain that the point (x_i, y_j) is in T_R, i.e., $x_i + y_j < 1$, but the point (x_{i+1}, y_{j+1}) might not satisfy that $x_{i+1} + y_{j+1} \leq 1$, and therefore we cannot employ these finite difference formulae if a term in u_{xy} appears. Fortunately, we will check that this term cancels after this transformation [given by Equations (9 and 10)] whenever the original triangle with vertices $(\bar{x}_1, \bar{y}_1), (\bar{x}_0, \bar{y}_0)$, and $(\bar{x}_2, \bar{y}_2)$ is a right triangle and $(\bar{x}_0, \bar{y}_0)$ is the vertex of the right angle. This fact is explained in **Figure 1**. If we would need to approximate $\frac{\partial^2 u_{2,1}}{\partial x \partial y}$, then it would be necessary to obtain an approximation of $u_{3,2}$, but this point is outside of the T_R, the region of study.

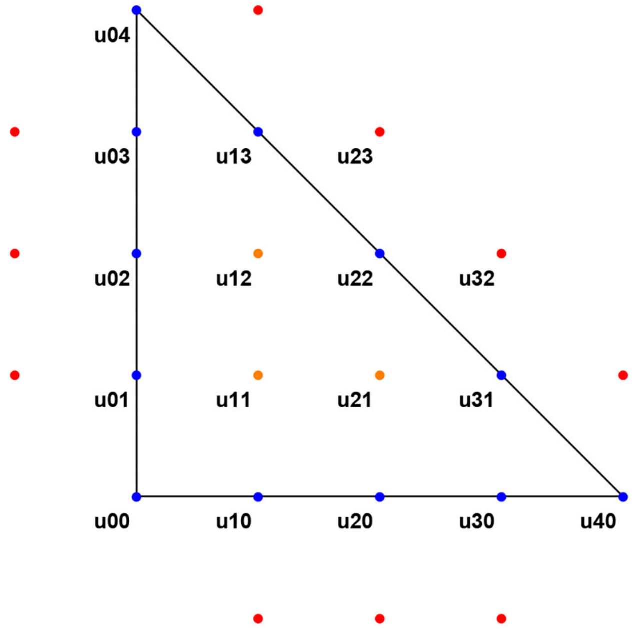

FIGURE 1 | Spatial discretization in the right triangle T_R. We need to approximate partial derivatives of u in the interior (orange points), and therefore we have obtained in the previous steps approximations of the function in the interior, and the points in the border (blue points), but we do not have these values outside of T_R (red points).

In this work, we are employing only second-order approximations in space. In other works, for example [13], we have also employed SERK codes after higher-order discretizations in space, but in rectangles. Normally, in rectangles, we can use formulae similar to

$$\begin{aligned} &\frac{\partial^2 u_{i,j}}{\partial x^2} = \frac{-u_{i+2,j} + 16u_{i+1,j} - 30u_{i,j} + 16u_{i-1,j} - u_{i-2,j}}{12h^2}, \\ &i = 2, \ldots, N - 2, \end{aligned} \tag{18}$$

and in the lower edge

$$\frac{\partial^2 u_{1,j}}{\partial x^2} = \frac{10u_{0,j} - 15u_{1,j} - 4u_{2,j} + 14u_{3,j} - 6u_{4,j} + u_{5,j}}{12h^2}. \tag{19}$$

However, in the triangle, again we can observe in **Figure 1**, that we would need to approximate the solution in points outside T_R before we can calculate (19) near the vertex $(0, 1)$. Obviously, one possible idea for the future is considering the decomposition of complex regions into bigger rectangles in the interior, and small right triangles near the border of the complex region where it is necessary to solve the PDE.

Now, we are ready to understand why we chose right triangles in the decomposition of complex regions. The main reason is, that simple calculations give us [after linear transformations given by Equations (9–11)]:

$$u_{\bar{x}\bar{x}} + u_{\bar{y}\bar{y}} = a_1\left(a_1 u_{xx} + b_1 u_{xy}\right) + b_1\left(a_1 u_{xy} + b_1 u_{yy}\right) \\ +a_2\left(a_2 u_{xx} + b_2 u_{xy}\right) + b_2\left(a_2 u_{xy} + b_2 u_{yy}\right), \tag{20}$$

and therefore, after this linear transformation, $u_{\bar{x}\bar{x}} + u_{\bar{y}\bar{y}}$ has the following term in u_{xy}

$$\left(2a_1 b_1 + 2a_2 b_2\right) u_{xy}. \tag{21}$$

If we change a_1, a_2, b_1, and b_2 for their values given by Equation (10)

$$a_1 b_1 + a_2 b_2 = \frac{\bar{y}_2 - \bar{y}_0}{Det}\frac{\bar{x}_0 - \bar{x}_2}{Det} + \frac{\bar{y}_0 - \bar{y}_1}{Det}\frac{\bar{x}_0 - \bar{x}_1}{Det} \tag{22}$$

which is 0 if and only if the vectors $\overrightarrow{P_2P_0}$ and $\overrightarrow{P_0P_1}$ are orthogonal, i.e., if they form a right angle at P_0.

Thus, if the original triangle has a right angle at P_0, there is not a term in u_{xy}, and we can use the second-order approximations in space, with the spatial discretization described above. Additionally, the following theorem guarantee that ESERK methods can be employed (with numerical stability and good results) in this right triangle to solve the PDE given by Equations (1)–(3) after the linear transformation given by Equations (9)–(11) and the spatial discretization given by Equation (15):

Theorem: Let Equations (1)–(3) be the PDE to be solved, and Ω a right triangle with a right angle at P_0. After linear transformation given by Equations (9) and (10), this PDE transforms into Equations (12)–(14), which can be discretized by Equations (15) and (16), transforming into the system of ODEs

$$\begin{pmatrix} u_{11} \\ u_{21} \\ \vdots \\ u_{N-2\ 1} \\ u_{12} \\ \vdots \\ u_{N-3\ 2} \\ \vdots \\ u_{1\ N-2} \end{pmatrix}' = A \begin{pmatrix} u_{11} \\ u_{21} \\ \vdots \\ u_{N-2\ 1} \\ u_{12} \\ \vdots \\ u_{N-3\ 2} \\ \vdots \\ u_{1\ N-2} \end{pmatrix} + \begin{pmatrix} F(t, x_1, y_1, u_{11}) \\ F(t, x_2, y_1, u_{21}) \\ \vdots \\ F(t, x_{N-2}, y_1, u_{N-2\ 1}) \\ F(t, x_1, y_2, u_{12}) \\ \vdots \\ F(t, x_{N-3}, y_2, u_{N-3\ 2}) \\ \vdots \\ F(t, x_1, y_{N-2}, u_{1\ N-2}) \end{pmatrix}, \tag{23}$$

$F(t, x_i, y_j, u_{ij})$ being the sum of $f(t, x, y, u)$ at the grid points plus the function given by the spatial discretization of the derivatives at the boundary.

The associate matrix, A, to the terms $c_1 u_{xx} + c_2 u_{yy}$ (with $c_1, c_2 \geq 0$) is real and symmetric, and therefore all the eigenvalues of this matrix are negative and real. Hence, Extrapolated Stabilized Explicit Runge–Kutta are numerically stable whenever $\partial_u[F(t, x, y, u)]$ does not modify this type of eigenvalues (real and negative) in the Jacobian function and $s > \sqrt{4\Delta t(\mu + \sigma)}$ (μ being $\frac{c_1}{h^2}$ and $\sigma = \frac{c_2}{h^2}$). Therefore, ESERK4 methods can solve Equations (1)–(3) with a fourth-order convergence in time, and second in space.

Proof: It only remains to study the associate matrix A. But simple calculations allow us to obtain that

$$A = \begin{pmatrix} B_{N-2} & C_{N-2,N-3} & 0_{N-2,N-4} & \dots & 0_{N-2,2} & 0_{N-2,1} \\ C^t_{N-2,N-3} & B_{N-3} & C_{N-3,N-4} & \dots & 0_{N-3,2} & 0_{N-2,1} \\ 0_{N-4,N-2} & C^t_{N-3,N-4} & B_{N-4} & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots \\ 0_{2,N-2} & 0_{2,N-3} & \dots & \dots & B_2 & C_{2,1} \\ 0_{1,N-2} & 0_{1,N-3} & \dots & \dots & C^t_{2,1} & B_1 \end{pmatrix}, \tag{24}$$

where B_i is the square matrix with dimension i

$$B_i = \begin{pmatrix} -2\mu - 2\sigma & \mu & 0 & \dots & 0 \\ \mu & -2\mu - 2\sigma & \mu & \ddots & 0 \\ 0 & \mu & -2\mu - 2\sigma & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \mu \\ 0 & 0 & \dots & \mu & -2\mu - 2\sigma \end{pmatrix}, \tag{25}$$

$$C_{i+1,i} = \begin{pmatrix} \sigma\ \mathrm{Id}_i \\ 0_{i,1} \end{pmatrix},$$

$0_{i,j}$ is the $i \times j$ matrix with all the values equal 0, Id_i is the identity matrix of dimension i, $\mu = \frac{c_1}{h^2}$ and $\sigma = \frac{c_2}{h^2}$, and therefore A is a symmetric real matrix.

Finally, it is well-known that all the eigenvalues of any symmetric real matrix A are real. Let us suppose that (λ, v) is a complex pair of A, i.e., an eigenvector $v = x + yi \in \mathbb{C}^n$, where $x, y \in \mathbb{R}^n$ and $\lambda = a + bi \in \mathbb{C}$ is the corresponding eigenvalue with $a, b \in \mathbb{R}$. Therefore,

$$Ax + iAy = Av = \lambda v = (ax - by) + i(bx + ay). \tag{26}$$

Hence, equalizing real and imaginary parts, we have

$$Ax = (ax - by), \quad Ay = (bx + ay), \tag{27}$$

and therefore

$$Ax \cdot y = a(x \cdot y) - b||y||^2, \quad x \cdot Ay = b||x||^2 + a(x \cdot y). \tag{28}$$

In this way we can conclude that

$$0 = x \cdot Ay - Ax \cdot y = b(||x||^2 + ||y||^2), \tag{29}$$

and, since $||x||^2 + ||y||^2 \neq 0$, then $b = 0$ and $\lambda = a \in \mathbb{R}$

Additionally, since $\sigma, \mu \geq 0$, the Gershgoring theorem guarantees for all the eigenvalues of A that $4(\mu + \sigma) \leq \lambda_i \leq 0$.

Therefore, whenever the nonlinear part does not modify this type of eigenvalues (real and negative) in the Jacobian function, a bound of the spectral radius of the Jacobian is $4(\mu + \sigma)$, and we merely need to use an ESERK method with $s > \sqrt{4\Delta t(\mu + \sigma)}$ to guarantee numerical stability.

TABLE 1 | Analysis of the numerical convergence at points $p_1 = (t, x, y) = (1, 0.15, 0.15)$ (top) and $p_2 = (t, x, y) = (1, 0.5, 0.25)$ (bottom) for the ESERK4 algorithm with $s = 100$ with $k = \Delta t = 0.2, 0.1$ and 0.05, and $h = \Delta x = \Delta y = 0.025, 0.0125$, and 0.00625.

$s = 100, p_1$	**$k = 0.2$**	**$k = 0.1$**	**$k = 0.05$**	**Temporal conv.**
$h = 0.025$	$2.264e_{-4}$	$1.215e_{-5}$	$1.595e_{-5}$	
$h = 0.0125$	$3.355e_{-4}$	$6.364e_{-6}$	$2.479e_{-6}$	
$h = 0.00625$	$4.430e_{-4}$	$5.842e_{-5}$	$3.109e_{-6}$	**3.577**
Spatial conv.			1.180	
$s = 100, p_2$	**$k = 0.2$**	**$k = 0.1$**	**$k = 0.05$**	**Temporal conv.**
$h = 0.025$	$3.449e_{-4}$	$2.709e_{-6}$	$4.073e_{-5}$	
$h = 0.0125$	$1.412e_{-3}$	$8.117e_{-5}$	$1.479e_{-5}$	
$h = 0.00625$	$1.649e_{-3}$	$1.975e_{-5}$	$4.536e_{-6}$	**4.253**
Spatial conv.			1.583	

TABLE 2 | Analysis of the numerical convergence at points $p_1 = (t, x, y) = (1, 0.15, 0.15)$ (top) and $p_2 = (t, x, y) = (1, 0.5, 0.25)$ (bottom) for the ESERK4 algorithm with $s = 150$ with $k = \Delta t = 0.2, 0.1$ and 0.05, and $h = \Delta x = \Delta y = 0.025, 0.0125$, and 0.00625.

$s = 150, p_1$	**$k = 0.2$**	**$k = 0.1$**	**$k = 0.05$**	**Temporal conv.**
$h = 0.025$	$2.150e_{-4}$	$1.228e_{-5}$	$1.599e_{-5}$	
$h = 0.0125$	$3.235e_{-4}$	$5.693e_{-6}$	$2.132e_{-6}$	
$h = 0.00625$	$4.401e_{-4}$	$5.842e_{-5}$	$2.625e_{-6}$	**3.695**
Spatial conv.			1.303	
$s = 150, p_2$	**$k = 0.2$**	**$k = 0.1$**	**$k = 0.05$**	**Temporal conv.**
$h = 0.025$	$3.275e_{-4}$	$3.657e_{-6}$	$4.042e_{-5}$	
$h = 0.0125$	$1.327e_{-3}$	$7.585e_{-5}$	$1.788e_{-5}$	
$h = 0.00625$	$1.568e_{-3}$	$1.975e_{-5}$	$3.824e_{-6}$	**4.340**
Spatial conv.			1.701	

TABLE 3 | Analysis of the numerical convergence at points $p_1 = (t, x, y) = (1, 0.15, 0.15)$, and $p_2 = (t, x, y) = (1, 0.5, 0.25)$ for the ESERK4 algorithm withs $= 100$ and $s = 150$ with $k = \Delta t = 0.025$, and $h = \Delta x = \Delta y = 0.025, 0.0125$, and 0.00625.

	$s = 100, p_1$	**$s = 100, p_2$**	**$s = 150, p_1$**	**$s = 150, p_2$**
$h = 0.025$	$1.749e_{-5}$	$3.489e_{-5}$	$1.750e_{-5}$	$3.490e_{-5}$
$h = 0.0125$	$4.544e_{-6}$	$9.593e_{-6}$	$4.542e_{-6}$	$9.575e_{-6}$
$h = 0.00625$	$2.086e_{-6}$	$9.398e_{-7}$	$2.026e_{-6}$	$7.835e_{-7}$
Spatial conv.	**1.534**	**2.607**	**1.555**	**2.738**

4. NUMERICAL EXAMPLE

Let us now study the numerical behavior of ESERK methods in a right triangle with one example. We will consider

$$u_t = \frac{5}{\pi^2}(u_{\bar{x}\bar{x}} + u_{\bar{y}\bar{y}}) + (1-u)^3 + f(t, \bar{x}, \bar{y}) \qquad (\bar{x}, \bar{y}) \in \Omega \subset \mathbb{R}^2, \tag{30}$$

where

$$f(t, \bar{x}, \bar{y}) = e^{-3t}\left(\sin\left(\frac{\pi(\bar{y} - 2\bar{x} - 3)}{5}\right) - e^t\right)^3,$$

Ω is the triangle with vertices $(-1, 1)$, $(-3, 2)$, and $(0, 3)$ and initial and boundary conditions are taken such that $u(t, \bar{x}, \bar{y}) = e^{-t}\sin\left(\frac{\pi(\bar{y}-2\bar{x}-3)}{5}\right)$ is its solution.

Hence, we first consider the linear transformation given by Equations (9) and (10), i.e., $a_1 = -2/5, a_2 = 1/5, b_1 = 1/5, b_2 = 2/5$. In this way Equation (30) transforms into the Equation

$$u_t = \frac{1}{\pi^2}(u_{xx} + u_{yy}) + (1-u)^3 + f(t, x, y) \qquad (x, y) \in T_R, \tag{31}$$

where

$$f(t, x, y) = e^{-3t}\left(\sin(\pi x) - e^t\right)^3,$$

and initial and boundary conditions are taken such that $u(t, x, y) = e^{-t}\sin(\pi x)$ is its solution.

Now, it is possible to utilize second-order approximations in space, as it was explained in the previous section. ESERK4 with $s = 100$ and 150 where considered for this numerical experiment with different values of $h = \Delta x = \Delta y$ and $k = \Delta t$. Numerical convergence at several points was analyzed with both methods, and numerical errors at two points [$p_1 = (t, x, y) = (1, 0.15, 0.15)$ and $p_2 = (t, x, y) = (1, 0.5, 0.25)$] are shown in **Tables 1**, **2**.

First of all, we calculated all the eigenvalues of the matrix A after spatial discretization. As it was demonstrated in Theorem 23, they are real and negative, and they are inside the intervals $[-1, 292, 0]$ for $h = 0.025$; $[-5, 183, 0]$ for $h = 0.0125$; and $[-20, 746, 0]$ for $h = 0.00625$. In the three cases, the bound $4(\mu + \sigma)$ given by Gershgoring theorem is a good approximation for the spectral radius [$4(\mu+\sigma)$ is 1296.91 for $h = 0.025$, 5187.64 for $h = 0.0125$, and 20750.6 for $h = 0.00625$, less than a 1% over the real values].

ESERK4 schemes are stable in $[-s^2, 0]$ therefore any ESERK method with $s > \sqrt{20750.6k} \geq \sqrt{4150.2} = 64.4214$ (since our bigger $k = 0.2$) is stable in this numerical example.

In both **Tables 1**, **2**, if we take $k = 0.2$ (also with $k = 0.1$), we can observe that errors are similar with the three different values $h = 0.025, 0.0125$, and 0.00625 at many of the points. In this case, most of the error is due to the temporal discretization. Actually, in L_2 norm, errors with constant $k = 0.2$ grow when h decrease for the three step lengths in space, this is because there are more points and they are close to the border.

If we take $h = 0.025$ constant, and we vary $k = 0.2, 0.1$, and $k = 0.05$, in general we observe that errors in most of the points decrease between $k = 0.2$ and 0.1, however, if we only compare the errors with $h = 0.025, k = 0.1$ and $k = 0.05$, errors are similar at most points (and also in L_2 norm). Obviously, this is because, with $h = 0.025$, $k = 0.1$, or $k = 0.05$, part of the error is due to the spatial discretization.

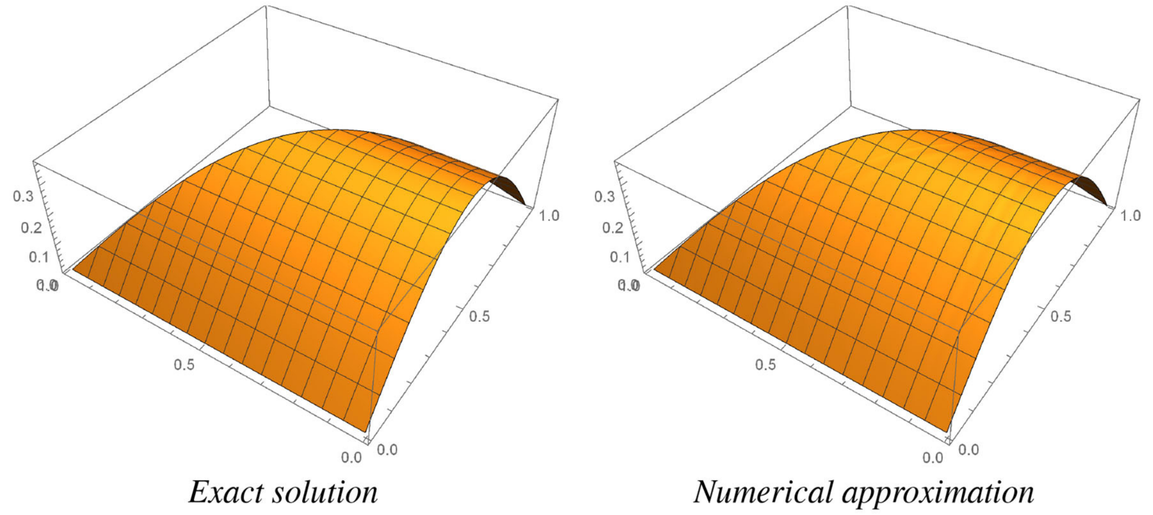

FIGURE 2 | Exact and numerical solutions in the right triangle T_R. Numerical approximation is obtained with ESERK4 with $s = 150$, $k = \Delta t = 0.025$, $h = \Delta x = \Delta y = 0.00625$.

Therefore, it is not so easy to observe 4−th order convergence in time and 2−nd in space. If we choose $h = 0.00625$, then most of the error with $k_1 = 0.2$, $k_2 = 0.1$, and $k_3 = 0.05$ is due to temporal discretization. Hence, calculating $\log_{k_1/k_3}\left(\frac{err_1}{err_3}\right)$ (these values are called Temporal convergence in **Tables 1**, **2**, err_1 being the error with k_1, and err_3 being the error with k_3) we can observe numerical rates in the range 3.6–4.3 in general, which gives us a good idea of the fourth-order convergence in time of ESERK4 schemes.

Now, if we fix $k = 0.05$, and we repeat the process with $h_1 = 0.025$, $h_2 = 0.0125$, and $h_3 = 0.00625$, we observe that between h_1 and h_2 errors divide (more or less) by 4 which gives us a good idea of the second order in space of the discretization proposed for the right triangle. However, with $k = 0.05$, and h_3 a part of the error is due to the temporal discretization. Thus, if we calculate $\log_{h_1/h_3}\left(\frac{err_1}{err_3}\right)$ (these values are called Spatial convergence in **Tables 1**, **2**, err_1 being the error with h_1, and err_3 being the error with h_3), we observe numerical rates in the range 1.2–1.7.

Since, part of the error with $k = 0.05$ is due to the temporal discretization, and the temporal convergence is fourth-order, let us choose a smaller $k_4 = 0.025$, and repeat the process with this length step in time. In **Table 3**, errors with both methods ($s = 100$ and $s = 150$), and $h = \Delta x = \Delta y = 0.025, 0.0125$, and 0.00625 are shown at both points, p_1 and p_2.

Now, most of the errors are because of the spatial discretization, and we can observe that the numerical spatial convergence rates are in the range 1.5–2.7. They suggest that the numerical convergence rate is 2 as it was expected from the previous theoretical analysis.

In **Figure 2**, the exact solution and the numerical approximation obtained with ESERK4 with $s = 150$, $k = \Delta t = 0.025$, $h = \Delta x = \Delta y = 0.00625$ are shown. We can check that both plots look identical.

5. CONCLUSIONS AND FUTURE GOALS

In this paper, for the first time, ESERK schemes are proposed to solve nonlinear partial differential equations (PDEs) in right triangles. These codes are explicit, they do not require to solve very large systems of linear nor nonlinear equations at each step. It is demonstrated that such type of codes are able to solve nonlinear PDEs in right triangles. They keep the order of convergence and the absolute stability property under certain conditions. Hence, this paper opens a new line of research, because this new approach will allow, in the future, to solve nonlinear parabolic PDEs with stabilized explicit Runge–Kutta schemes in complex domains, that would be decomposed in rectangles and right triangles.

Additionally, we consider that this procedure can be extended to tetrahedron and other simplixes for the solution of multi-dimensional nonlinear PDEs in complex regions in $\mathbb{R}^n$.

DATA AVAILABILITY STATEMENT

The datasets generated for this study are available on request to the corresponding author.

AUTHOR CONTRIBUTIONS

The author confirms being the sole contributor of this work and has approved it for publication.

FUNDING

The authors acknowledges support from the University of Salamanca through its own Programa Propio I, Modalidad C2 grant 18.KB2B.

REFERENCES

1. Aiken RC. *Stiff Computation*. New York, NY: Oxford University Press, Inc. (1985).
2. Cubillos-Moraga MA. *General-Domain Compressible Navier-Stokes Solvers Exhibiting Quasi-Unconditional Stability and High-Order Accuracy in Space and Time*. PhD thesis. Pasadena, CA: California Institute of Technology (2015). Available online at: https://thesis.library.caltech.edu/8851/
3. Duarte M, Bonaventura Z, Massot M, Bourdon A, Descombes S, Dumont T. A new numerical strategy with space-time adaptivity and error control for multi-scale streamer discharge simulations. *J Comput Phys*. (2012) **231**:1002–19. doi: 10.1016/j.jcp.2011.07.002
4. Duarte M, Descombes S, Tenaud C, Candel S, Massot M. Time-space adaptive numerical methods for the simulation of combustion fronts. *Combust Flame*. (2013) **160**:1083–101. doi: 10.1016/j.combustflame.2013.01.013
5. Dumont T, Duarte M, Descombes S, Dronne MA, Massot M, Louvet V. Simulation of human ischemic stroke in realistic 3D geometry. *Commun Nonlin Sci Numer Simul*. (2013) **18**:1539–57. doi: 10.1016/j.cnsns.2012.10.002
6. Hairer E, Wanner G. *Solving Ordinary Differential Equations. II: Stiff and Differential-Algebraic Problems*. Berlin: Springer (1996). doi: 10.1007/978-3-642-05221-7
7. Hundsdorfer WH, Verwer JG. *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*. Berlin; Heidelberg: Springer (2007).
8. Martín-Vaquero J, Khaliq AQM, Kleefeld B. Stabilized explicit Runge-Kutta methods for multi-asset American options. *Comput Math Appl*. (2014) **67**:1293–308. doi: 10.1016/j.camwa.2014.01.018
9. Zlatev Z. *Computer Treatment of Large Air Pollution Models*. Dordrecht: Environmental Science and Technology Library; Springer Netherlands (2012). Available online at: https://www.springer.com/gp/book/9780792333289
10. Abdulle A, Medovikov AA. Second order Chebyshev methods based on orthogonal polynomials. *Numer Math*. (2001) **90**:1–18. doi: 10.1007/s002110100292
11. Abdulle A. Fourth order Chebyshev methods with recurrence relation. *SIAM J Sci Comput*. (2001) **23**:2041–54. doi: 10.1137/S1064827500379549
12. Kleefeld B, Martín-Vaquero J. SERK2v2: A new second-order stabilized explicit Runge-Kutta method for stiff problems. *Numer Methods Part Differ Equat*. (2013) **29**:170–85. doi: 10.1002/num.21704
13. Kleefeld B, Martín-Vaquero J. SERK2v3: solving mildly stiff nonlinear Partial Differential Equations. *J Comput Appl Math*. (2016) **299**:194–206. doi: 10.1016/j.cam.2015.11.045
14. Martín-Vaquero J, Janssen B. Second-order stabilized explicit Runge-Kutta methods for stiff problems. *Comput Phys Commun*. (2009) **180**:1802–10. doi: 10.1016/j.cpc.2009.05.006
15. Medovikov AA. High order explicit methods for parabolic equations. *BIT Numer Math*. (1998) **38**:372–90. doi: 10.1007/BF02512373
16. Sommeijer BP, Shampine LF, Verwer JG. RKC: An explicit solver for parabolic PDEs. *J Comput Appl Math*. (1997) **88**:315–26. doi: 10.1016/S0377-0427(97)00219-7
17. Martín-Vaquero J, Kleefeld B. Extrapolated stabilized explicit Runge-Kutta methods. *J Comput Phys*. (2016) **326**:141–155. doi: 10.1016/j.jcp.2016.08.042
18. Martín-Vaquero J, Kleefeld A. ESERK5: A fifth-order extrapolated stabilized explicit Runge- Kutta method. *J Comput Appl Math*. (2019) **356**:22–36. doi: 10.1016/j.cam.2019.01.040
19. Martín-Vaquero J, Kleefeld A. Solving nonlinear parabolic PDEs in several dimensions: parallelized ESERK codes. *J. Comput. Phys*. (2020) 109771. doi: 10.1016/j.jcp.2020.109771 Available online at: https://www.sciencedirect.com/science/article/pii/S0021999120305453
20. Farago I. Finite element method for solving nonlinear parabolic equations. *Comput Math Appl*. (1991) **21**:59–69. doi: 10.1016/0898-1221(91)90231-R
21. Duarte M, Bonaventura Z, Massot M, Bourdon A. A numerical strategy to discretize and solve the Poisson equation on dynamically adapted multiresolution grids for time-dependent streamer discharge simulations. *J Comput Phys*. (2015) **289**:129–48. doi: 10.1016/j.jcp.2015.02.038

Conflict of Interest: The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.