

Article

Low-Complexity Numerical Approach for the Diffusion Equation with Variable Diffusion Coefficient

Marta Zárraga-Rodríguez ¹, Patricio Fuentes ² and Xabier Insausti ^{1,*}¹ Department of Biomedical Engineering and Sciences, Tecnun-School of Engineering, University of Navarra, 20018 Donostia-San Sebastian, Spain; mzarraga@unav.es² Photonic Inc., Vancouver, BC V3K 6T1, Canada; pfuentes@photonic.com

* Correspondence: xinsausti@unav.es

Abstract

The diffusion equation models a wide variety of physical and chemical processes and has significant interest in many scientific disciplines. Analytical and numerical methods found in the literature for solving the diffusion equation consider a constant diffusion coefficient. However, in reality, it is not constant. In this paper, we present a numerical approach to solve the diffusion equation when the diffusion coefficient is not constant. Unlike existing methods that require solving non-linear systems with iterative schemes, our approach transforms the problem into a linear system, drastically reducing computational cost while preserving temporal accuracy.

Keywords: numerical methods; diffusion equation; non-constant diffusion coefficient**MSC:** 65M06; 65M22

1. Introduction

The phenomenon of diffusion is one of the most important and well-documented mechanisms in the fields of physics and chemistry. It is present in myriads of physical events and materials, having been studied in metals, alloys, polymers, fluids, reactive molecules, and even quantum mechanical systems [1–6]. Naturally, this makes diffusion a phenomenon of significant interest in many scientific disciplines such as chromatography, catalysis, metallurgy, and semiconductor technology [7–12]. Diffusion is described by the diffusion equation, which can be used to represent a wide variety of processes like the movement of matter as a result of random molecular motion, the stationary motion of a boundary layer of fluid over a plate, or heat transfer in tumors [13,14].

Given its wide variety of applications, scientists have always been interested in building methods capable of efficiently solving the diffusion equation [15]. This is reflected in the literature, where examples in Cartesian coordinates [16], cylindrical/polar coordinates [17], and spherical coordinates [18] can be found. Generally, these methods fall into one of two overarching approaches: analytical or numerical. Analytically solving a nonlinear diffusion equation is not an easy task. In the literature, exact solutions can be found for a relatively small number of cases under restrictive assumptions regarding the diffusion coefficient or the boundary and initial conditions (see, for instance [19–21]). For this reason, analytical solutions are limited in terms of their utility.

In contrast, numerical methods are useful for handling many practical problems that involve nonlinearities, complex geometries, and even complicated boundary conditions [22]. There are many strategies based on numerical methods that can be used to



Academic Editor: Héctor Jairo Martínez-Romero

Received: 4 December 2025

Revised: 23 December 2025

Accepted: 10 January 2026

Published: 13 January 2026

Copyright: © 2026 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

compute solutions to the diffusion equation. In [23], nine different numerical schemes for the solution of the three-dimensional heat diffusion equation are compared. Among these methods, the finite difference methods stand out due to their relative accuracy and reduced computational complexity. Moreover, the Crank–Nicolson method [24] is said to be advantageous in other instances due to its unconditional stability. However, when different numerical methods are compared to solve the diffusion equation, parameters like the diffusion coefficient are considered to be constant. Recent advancements have addressed stability analysis and error estimation in variable coefficient convection–diffusion equations, providing insights into generalized numerical fluxes [25]. In reality, these parameters are actually not constant, but such assumption facilitates the resolution of the equations. Heat diffusion can also be influenced by external fields. Specifically, in optimal control problems for reaction–diffusion–convection equations with variable coefficients, it is crucial to consider the effects of the external environment, as explored in recent studies [25,26]. However, heat transfer control problems are beyond the scope of this paper, and we have considered a system without external environmental effects. Several recent contributions have addressed the numerical solution of diffusion-type equations using different strategies, such as boundary element formulations [27] and iterative schemes for non-linear PDEs [28]. However, these approaches often involve significant computational cost, which our method aims to reduce.

When the Crank–Nicolson method is used to numerically solve the diffusion equation with a non-constant diffusion coefficient, a system of non-linear equations arises. The standard approach to solve such problem is to combine the Crank–Nicolson method with the multivariate Newton–Raphson method (see, e.g., Refs. [29,30]).

In this paper, we present a low-complexity numerical approach to solve the diffusion equation when the diffusion coefficient is not constant based on a modification of the Crank–Nicolson finite difference method. This modification leads to a system of linear equations that can be efficiently solved using basic matrix algebra. Moreover, we will prove that our approach and the standard approach have the same accuracy with respect to the time step, although the accuracy of our approach decreases with respect to the spatial step. Hence, our approach is appropriate for performing long-term simulations with very low complexity. To the best of our knowledge, no previous work has proposed a modification of the Crank–Nicolson scheme that avoids Newton–Raphson iterations for non-constant diffusion coefficients. This paper introduces such an approach, enabling long-term simulations with minimal complexity.

The remainder of this paper is organized as follows. Section 2 states preliminary considerations regarding the diffusion equation, the finite difference approximation of partial derivatives, and the multivariate Newton–Raphson method. Section 3 presents the standard method and our alternative approach to solve the diffusion equation when the diffusion coefficient is not constant. Section 4 provides an application example to clarify the implementation of the two considered approaches. Finally, an illustrative example and some conclusions are given in Sections 5 and 6.

2. Preliminaries

This section is devoted to reviewing the diffusion equation and the bases of the numerical methods that will be used in the problem statement.

2.1. Diffusion Equation

The diffusion equation is a second-order non-linear parabolic partial differential equation, given as follows:

$$\frac{\partial u(t, \mathbf{p})}{\partial t} = \operatorname{div} \left(\lambda(u) \nabla u(t, \mathbf{p}) \right), \quad (1)$$

where u is a function of the independent variables t and $\mathbf{p}$, t represents time, $\mathbf{p} = (\alpha_1, \alpha_2, \alpha_3)$ represents a position on a certain three-dimensional coordinate system, div denotes the divergence operator, ∇ denotes the gradient operator, and $\lambda(u)$ is known as the isotropic diffusion coefficient. If the diffusion coefficient is constant, Equation (1) becomes a linear partial differential equation. In the sequel, we assume that $u(t, \mathbf{p})$ is continuously differentiable up to the fourth order in all the variables within the domain of interest. Similarly, we assume that $\lambda(u)$ is continuously differentiable.

For linear partial differential equations, there are several well-known numerical methods that provide an approximation of the solution by solving a system of linear equations. Due to the structure of the matrices that arise in such systems, they can be solved with very low complexity. However, the main difficulty for numerically solving Equation (1) with low complexity arises from its non-linearity due to the non-constant diffusion coefficient.

2.2. Finite Difference Approximations of Derivatives

Let f be a real-valued function defined on an interval $[a, b]$ in $\mathbb{R}$, and assume it has derivatives of every order at each point of $[a, b]$. Then, according to Taylor's formula (see, e.g., Ref. [31]), for every $x \in [a, b]$ and for every $n \in \mathbb{N}$, we have

$$f(x) = \sum_{k=0}^{n-1} \frac{f^{(k)}(c)}{k!} (x-c)^k + \frac{f^{(n)}(x_0)}{n!} (x-c)^n,$$

whenever there is a positive constant M such that

$$|f^{(n)}(x_0)| \leq M^n,$$

with $c \in [a, b]$, $f^{(0)} = f$, $f^{(n)}$ being the n th derivative of f and x_0 being some point between x and c .

Hence, for every $h > 0$ satisfying that $c+h \in [a, b]$, we have

$$f(c+h) = \sum_{k=0}^{n-1} \frac{f^{(k)}(c)}{k!} h^k + \frac{f^{(n)}(x_1)}{n!} h^n, \quad (2)$$

where $x_1 \in [c, c+h]$, and for every $h > 0$ satisfying that $c-h \in [a, b]$, we have

$$f(c-h) = \sum_{k=0}^{n-1} \frac{f^{(k)}(c)}{k!} (-h)^k + \frac{f^{(n)}(x_2)}{n!} (-h)^n, \quad (3)$$

where $x_2 \in [c-h, c]$.

For $n = 2$, from (2), we have

$$f(c+h) = f(c) + f^{(1)}(c)h + \frac{f^{(2)}(x_1)}{2!} h^2, \quad (4)$$

therefore, we obtain the Forward Difference method for approximating the first derivative of f :

$$\frac{f(c+h) - f(c)}{h} = f^{(1)}(c) + \frac{f^{(2)}(x_1)}{2!} h = f^{(1)}(c) + O(h), \quad (5)$$

where O denotes the Big O notation of Landau and represents the truncation error. For $n = 2$, from (3), we have

$$f(c-h) = f(c) + f^{(1)}(c)(-h) + \frac{f^{(2)}(x_1)}{2!} (-h)^2. \quad (6)$$

The addition of (4) and (6) yields

$$f(c+h) + f(c-h) = 2f^{(0)}(c) + \frac{h^2}{2!} \left(f^{(2)}(x_1) + f^{(2)}(x_2) \right) = 2f(c) + O(h^2). \quad (7)$$

For $n = 3$, from (2), we have

$$f(c+h) = f(c) + f^{(1)}(c)h + \frac{f^{(2)}(c)}{2}h^2 + \frac{f^{(3)}(x_1)}{3!}h^3, \quad (8)$$

and from (3), we have

$$f(c-h) = f(c) + f^{(1)}(c)(-h) + \frac{f^{(2)}(c)}{2}(-h)^2 + \frac{f^{(3)}(x_2)}{3!}(-h)^3. \quad (9)$$

Subtracting (9) from (8) yields

$$f(c+h) - f(c-h) = 2f^{(1)}(c)h + \frac{h^3}{3!} \left(f^{(3)}(x_1) + f^{(3)}(x_2) \right),$$

and therefore, we obtain the Central Difference method for approximating the first derivative of f :

$$\frac{f(c+h) - f(c-h)}{2h} = f^{(1)}(c) + O(h^2). \quad (10)$$

For $n = 4$, from (2), we have

$$f(c+h) = f(c) + f^{(1)}(c)h + \frac{f^{(2)}(c)}{2}h^2 + \frac{f^{(3)}(c)}{3!}h^3 + \frac{f^{(4)}(x_1)}{4!}h^4, \quad (11)$$

and from (3), we have

$$f(c-h) = f(c) + f^{(1)}(c)(-h) + \frac{f^{(2)}(c)}{2}(-h)^2 + \frac{f^{(3)}(c)}{3!}(-h)^3 + \frac{f^{(4)}(x_2)}{4!}(-h)^4. \quad (12)$$

The addition of (11) and (12) yields

$$f(c+h) + f(c-h) = 2f(c) + f^{(2)}(c)h^2 + \frac{h^4}{4!} \left(f^{(4)}(x_1) + f^{(4)}(x_2) \right),$$

and therefore, we obtain the Central Difference method for approximating the second derivative of f :

$$\frac{f(c+h) - 2f(c) + f(c-h)}{h^2} = f^{(2)}(c) + O(h^2). \quad (13)$$

2.3. Finite Difference Approximations of Partial Derivatives: Crank–Nicolson Method

Crank–Nicolson is a well-known finite difference (implicit) method for numerically solving partial differential equations. The basic idea behind the Crank–Nicolson method is to discretize both the spatial and the time domain and approximate the partial derivatives by the corresponding finite differences expressions at those points [32]. This method has been extensively studied and modified to enhance its performance and applicability in various contexts. For instance, comparative analyses between the classic and modified versions have been conducted to assess their efficiencies [33]. Furthermore, its integration into mixed finite element schemes has demonstrated its versatility in more complex numerical frameworks [34].

Specifically, the Crank–Nicolson method approximates the partial derivatives by the corresponding finite difference expressions at point $(t + \frac{\Delta t}{2}, \mathbf{p})$ as follows:

$$\frac{\partial u}{\partial t} \left(t + \frac{\Delta t}{2}, \mathbf{p} \right) = \frac{u(t + \Delta t, \mathbf{p}) - u(t, \mathbf{p})}{\Delta t} + O(\Delta t^2), \quad (14)$$

$$\begin{aligned} \frac{\partial u}{\partial \alpha_i} \left(t + \frac{\Delta t}{2}, \mathbf{p} \right) &= \frac{1}{2} \left(\frac{u(t + \Delta t, \mathbf{p} + \Delta \alpha_i \mathbf{e}_i) - u(t + \Delta t, \mathbf{p} - \Delta \alpha_i \mathbf{e}_i)}{2\Delta \alpha_i} \right. \\ &\quad \left. + \frac{u(t, \mathbf{p} + \Delta \alpha_i \mathbf{e}_i) - u(t, \mathbf{p} - \Delta \alpha_i \mathbf{e}_i)}{2\Delta \alpha_i} \right) + O(\Delta \alpha_i^2), \end{aligned} \quad (15)$$

$$\begin{aligned} \frac{\partial^2 u}{\partial \alpha_i^2} \left(t + \frac{\Delta t}{2}, \mathbf{p} \right) &= \frac{1}{2} \left(\frac{u(t + \Delta t, \mathbf{p} + \Delta \alpha_i \mathbf{e}_i) - 2u(t + \Delta t, \mathbf{p}) + u(t + \Delta t, \mathbf{p} - \Delta \alpha_i \mathbf{e}_i)}{\Delta \alpha_i^2} \right. \\ &\quad \left. + \frac{u(t, \mathbf{p} + \Delta \alpha_i \mathbf{e}_i) - 2u(t, \mathbf{p}) + u(t, \mathbf{p} - \Delta \alpha_i \mathbf{e}_i)}{\Delta \alpha_i^2} \right) + O(\Delta \alpha_i^2), \end{aligned} \quad (16)$$

where Δt is the time step, $\Delta \alpha_i$ is the spatial step in coordinate α_i with $i \in \{1, 2, 3\}$, $\mathbf{e}_1 = (1, 0, 0)$, $\mathbf{e}_2 = (0, 1, 0)$, and $\mathbf{e}_3 = (0, 0, 1)$. Observe that Crank–Nicolson is a second-order method. The main advantage of this method is that it is unconditionally stable for linear partial differential equations. Although the approximations of the partial derivatives in (14)–(16) are well-known and commonly used, we have not found a formal explanation regarding the point at which the approximations are made or how the expressions with their corresponding truncation error are derived. Therefore, for the convenience of the reader, the details are explained below.

We first consider the Central Difference method for approximating the first derivative of f in (10) with $f = u$, $c = t_0$, and $h = \frac{\Delta t}{2}$. Hence, we have

$$\frac{u(t_0 + \frac{\Delta t}{2}, \mathbf{p}) - u(t_0 - \frac{\Delta t}{2}, \mathbf{p})}{\Delta t} = \frac{\partial u}{\partial t}(t_0, \mathbf{p}) + O(\Delta t^2), \quad (17)$$

for all $t_0 \in [0, \infty)$. Consequently, as (10) also holds for $t = t_0 + \frac{\Delta t}{2}$, (14) is directly obtained.

Now we consider again (10), but with $f = u$, $c = \mathbf{p}$, and $h = \Delta \alpha_i \mathbf{e}_i$. Hence, for each $i \in \{1, 2, 3\}$, we have

$$\frac{u(t + \frac{\Delta t}{2}, \mathbf{p} + \Delta \alpha_i \mathbf{e}_i) - u(t + \frac{\Delta t}{2}, \mathbf{p} - \Delta \alpha_i \mathbf{e}_i)}{2\Delta \alpha_i} = \frac{\partial u}{\partial t} \left(t + \frac{\Delta t}{2}, \mathbf{p} \right) + O(\Delta \alpha_i^2). \quad (18)$$

Consider (7) with $f = u$, $c = t + \frac{\Delta t}{2}$, and $h = \frac{\Delta t}{2}$. Hence, we have

$$u(t + \Delta t, \mathbf{p}) + u(t, \mathbf{p}) = 2u(t + \frac{\Delta t}{2}, \mathbf{p}) + O(\Delta t^2). \quad (19)$$

By combining (18) and (19), (15) is obtained.

Finally, we consider the Central Difference method for approximating the second derivative of f in (13) with $f = u$, $c = \mathbf{p}$, and $h = \Delta \alpha_i \mathbf{e}_i$. Hence, for each $i \in \{1, 2, 3\}$, we have

$$\frac{u(t + \frac{\Delta t}{2}, \mathbf{p} + \Delta \alpha_i \mathbf{e}_i) - u(t + \frac{\Delta t}{2}, \mathbf{p}) + u(t + \frac{\Delta t}{2}, \mathbf{p} - \Delta \alpha_i \mathbf{e}_i)}{\Delta \alpha_i^2} = \frac{\partial^2 u}{\partial t^2} \left(t + \frac{\Delta t}{2}, \mathbf{p} \right) + O(\Delta \alpha_i^2). \quad (20)$$

By combining (19) and (18), (16) is obtained.

2.4. Multivariate Newton–Raphson Method

Newton–Raphson is an iterative method to approximate a root $\boldsymbol{\beta}$ of a system of n non-linear equations and n unknowns $f(\mathbf{x}) = \mathbf{0}$. The iterative algorithm is as follows:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{h}_k \quad k \in \{0, 1, \dots\} \quad (21)$$

where $\mathbf{h}_k$ is the solution of the linear system of equations $\mathbf{J}_f(\mathbf{x}_k)\mathbf{h}_k = f(\mathbf{x}_k)$ and $\mathbf{J}_f(\mathbf{x}_k)$ is the Jacobian matrix of f at $\mathbf{x}_k$. Under certain conditions of f and a good enough initial guess $\mathbf{x}_0$, the iterative algorithm (21) converges to the sought root β . When the Newton–Raphson method converges, it can be proved that

$$\|\mathbf{x}_{k+1} - \beta\| = O(\|\mathbf{h}_k\|)$$

where $\|\cdot\|$ denotes the Euclidean norm (see, e.g., Ref. [35] (p. 249))

3. Problem Statement

In order to solve the non-linear Diffusion Equation (1), it is necessary to specify the coordinate system that will be used. In this paper, without loss of generality, we will use the cylindrical coordinate system, but everything that is said for cylindrical coordinates can be extrapolated directly to any other coordinate system. In cylindrical coordinates, $\mathbf{p} = (r, \phi, z)$; therefore, the non-linear Diffusion Equation (1) can be rewritten as follows:

$$\frac{\partial u}{\partial t} = \frac{1}{r} \frac{\partial}{\partial r} \left(r \lambda(u) \frac{\partial u}{\partial r} \right) + \frac{1}{r} \frac{\partial}{\partial \phi} \left(\frac{\lambda(u)}{r} \frac{\partial u}{\partial \phi} \right) + \frac{\partial}{\partial z} \left(\lambda(u) \frac{\partial u}{\partial z} \right), \quad (22)$$

where $u = u(t, r, \phi, z)$.

In order to simplify the expressions, from now on, we consider that u only depends on t and r . Based on this, Equation (22) can be written as follows:

$$\begin{aligned} \frac{\partial u}{\partial t} &= \frac{\lambda(u)}{r} \frac{\partial u}{\partial r} + \frac{\partial \lambda(u)}{\partial r} \frac{\partial u}{\partial r} + \lambda(u) \frac{\partial^2 u}{\partial r^2} \\ &= \frac{\lambda(u)}{r} \frac{\partial u}{\partial r} + \frac{\partial \lambda(u)}{\partial u} \left(\frac{\partial u}{\partial r} \right)^2 + \lambda(u) \frac{\partial^2 u}{\partial r^2}. \end{aligned} \quad (23)$$

Observe that since the diffusion coefficient $\lambda(u)$ is not constant, the resulting partial differential Equation (23) is not linear.

We aim to solve the Diffusion equation for $t \geq 0$ and $r \in [a, b]$. At this point, we present the standard approach and our alternative approach, both based on the Crank–Nicolson method, to numerically solve Equation (23). In both approaches, we discretize the time domain $[0, \infty)$ into subintervals of length Δt , and we obtain the following temporal partition:

$$t_i = i\Delta t, \quad i \in \{0, 1, \dots\}$$

It should be mentioned that although the Crank–Nicolson method is known to be unconditionally stable, when cylindrical coordinates are employed, a suitable choice of the integration step Δt is required [36,37].

Similarly, we discretize the space domain $[0, R]$ into N subintervals of length $\Delta r = \frac{R}{N}$, and we obtain the following spatial partition:

$$r_j = j\Delta r, \quad j \in \{0, 1, \dots, N\}$$

3.1. Standard Approach

We first introduce the standard approach. This approach combines the Crank–Nicolson method and the Newton–Raphson method to numerically solve Equation (23). Using Equations (14)–(16) to approximate the partial derivatives in Equation (23) at point $(t + \frac{\Delta t}{2}, r)$, we obtain the following:

$$\begin{aligned}
& \frac{u(t + \Delta t, r) - u(t, r)}{\Delta t} \\
&= \frac{\lambda(u(t, r))}{2r} \left(\frac{u(t + \Delta t, r + \Delta r) - u(t + \Delta t, r - \Delta r)}{2\Delta r} + \frac{u(t, r + \Delta r) - u(t, r - \Delta r)}{2\Delta r} \right) \\
&+ \frac{1}{4} \frac{\partial \lambda}{\partial u}(u(t, r)) \left(\frac{u(t + \Delta t, r + \Delta r) - u(t + \Delta t, r - \Delta r)}{2\Delta r} + \frac{u(t, r + \Delta r) - u(t, r - \Delta r)}{2\Delta r} \right)^2 \\
&+ \frac{\lambda(u(t, r))}{2} \left(\frac{u(t + \Delta t, r + \Delta r) - 2u(t + \Delta t, r) + u(t + \Delta t, r - \Delta r)}{\Delta r^2} \right. \\
&\quad \left. + \frac{u(t, r + \Delta r) - 2u(t, r) + u(t, r - \Delta r)}{\Delta r^2} \right) + O(\Delta t^2 + \Delta r^2) \quad (24)
\end{aligned}$$

We now apply the finite difference approximation given in Equation (24) to all the points (t_i, r_j) of the considered partition. To simplify the notation, we denote $u(t_i, r_j)$ by $u_{i,j}$, $\lambda(u(t_i, r_j))$ by $\lambda_{i,j}$, and $\frac{\partial \lambda}{\partial u}(u(t_i, r_j))$ by $\gamma_{i,j}$. We obtain the following system of non-linear equations:

$$\begin{aligned}
& u_{i,j} - u_{i+1,j} + \frac{\lambda_{i,j}}{4j} \frac{\Delta t}{\Delta r^2} (u_{i+1,j+1} - u_{i+1,j-1} + u_{i,j+1} - u_{i,j-1}) \\
&+ \frac{\gamma_{i,j}}{16} \frac{\Delta t}{\Delta r^2} (u_{i+1,j+1} - u_{i+1,j-1} + u_{i,j+1} - u_{i,j-1})^2 \\
&+ \frac{\lambda_{i,j}}{2} \frac{\Delta t}{\Delta r^2} (u_{i+1,j+1} - 2u_{i+1,j} + u_{i+1,j-1} + u_{i,j+1} - 2u_{i,j} + u_{i,j-1}) = 0 \quad (25)
\end{aligned}$$

for $i \in \{0, 1, \dots\}$ and $j \in \{1, \dots, N-1\}$. The initial condition provides the values of $u_{0,j}$ for all $j \in \{0, 1, \dots, N\}$. Therefore, for $i = 0$, we have a system of $N-1$ equations and $N+1$ unknowns ($u_{1,j}$ with $j \in \{0, 1, \dots, N\}$). The boundary conditions provide the remaining two equations that allow us to solve the system. By solving this system recursively for each i , we can obtain the approximation of the solution in all the points (t_i, r_j) of the partition.

Observe that the system of non-linear equations (25) cannot be solved using basic matrix algebra. The multivariate Newton–Raphson method is a well-known iterative method that allows us to solve the system of non-linear equations (25) (see Multivariate Newton–Raphson method). The most appealing characteristic of this strategy is its high accuracy, namely $O(\Delta t^2 + \Delta r^2)$, although this accuracy comes at the expense of increased computational complexity because for each time step, several systems of linear equations have to be solved.

3.2. Alternative Approach

In certain practical scenarios, it is interesting to reduce the computational complexity of the resolution method, even if it implies a slight accuracy loss. In this subsection, we propose an alternative approach to numerically solve the non-linear Diffusion Equation (1) with much lower complexity than using the Newton–Raphson method. To that end, we consider a different finite difference approximation for $\frac{\partial u}{\partial \alpha_i}$ at point $(t + \frac{\Delta t}{2}, r)$ than the one given in Equation (15). Specifically,

$$\frac{\partial u}{\partial \alpha_i} \left(t + \frac{\Delta t}{2}, p \right) = \left(\frac{u(t, p + \Delta \alpha_i e_i) - u(t, p - \Delta \alpha_i e_i)}{2\Delta \alpha_i} \right) + O(\Delta \alpha_i). \quad (26)$$

Using Equations (14), (26), and (16) to approximate the partial derivatives in Equation (23) at point $(t + \frac{\Delta t}{2}, r)$, we obtain the following:

$$\begin{aligned}
\frac{u(t + \Delta t, r) - u(t, r)}{\Delta t} &= \frac{\lambda(u(t, r))}{r} \left(\frac{u(t, r + \Delta r) - u(t, r - \Delta r)}{2\Delta r} \right) \\
&+ \frac{\partial \lambda}{\partial u}(u(t, r)) \left(\frac{u(t, r + \Delta r) - u(t, r - \Delta r)}{2\Delta r} \right)^2 \\
&+ \frac{\lambda(u(t, r))}{2} \left(\frac{u(t + \Delta t, r + \Delta r) - 2u(t + \Delta t, r) + u(t + \Delta t, r - \Delta r)}{\Delta r^2} \right. \\
&\left. + \frac{u(t, r + \Delta r) - 2u(t, r) + u(t, r - \Delta r)}{\Delta r^2} \right) + O(\Delta t^2 + \Delta r)
\end{aligned} \quad (27)$$

We now apply the finite difference approximation given in Equation (27) to all the points (t_i, r_j) of the considered partition, and we obtain the following system of equations:

$$\begin{aligned}
u_{i,j} - u_{i+1,j} + \frac{\lambda_{i,j}}{2j} \frac{\Delta t}{\Delta r^2} (u_{i,j+1} - u_{i,j-1}) + \frac{\gamma_{i,j}}{4} \frac{\Delta t}{\Delta r^2} (u_{i,j+1} - u_{i,j-1})^2 \\
+ \frac{\lambda_{i,j}}{2} \frac{\Delta t}{\Delta r^2} (u_{i+1,j+1} - 2u_{i+1,j} + u_{i+1,j-1} + u_{i,j+1} - 2u_{i,j} + u_{i,j-1}) = 0
\end{aligned} \quad (28)$$

for $i \in \{0, 1, \dots\}$ and $j \in \{1, \dots, N-1\}$. Given the initial and the boundary conditions for $i = 0$, (28) is a system of $N+1$ equations and $N+1$ unknowns ($u_{1,j}$ with $j \in \{0, \dots, N\}$). Observe that the system of equations (28) is now linear; therefore, it can be solved using basic matrix algebra. By solving this system recursively for each i , we can obtain the approximation of the solution in all the points (t_i, r_j) of the partition.

The computational complexity of this alternative approach is much lower than the approach based on the Newton–Raphson method because for each time step, we only need to solve a single system of linear equations. However, this approach yields an accuracy of $O(\Delta t^2 + \Delta r)$. Observe that the accuracy remains the same with respect to the time step but decreases with respect to the spatial step. Hence, this alternative approach is appropriate for performing long-term simulations with very low complexity.

4. Application Example

In this section, we provide an application example to clarify the implementation of the approaches presented in this paper to numerically solve the non-linear diffusion equation. We also compare the computational complexity of these approaches. As an example, we consider the following problem:

$$\begin{aligned}
\frac{\partial u}{\partial t} &= \frac{\lambda(u)}{r} \frac{\partial u}{\partial r} + \frac{\partial \lambda(u)}{\partial u} \left(\frac{\partial u}{\partial r} \right)^2 + \lambda(u) \frac{\partial^2 u}{\partial r^2} \quad t > 0, \quad 0 \leq r \leq R, \\
\text{subject to:} \\
u(0, r) &= u_0 \quad 0 \leq r \leq R, \\
\frac{\partial u}{\partial r}(t, 0) &= 0 \quad t > 0, \\
\frac{\partial u}{\partial r}(t, R) &= -H(u(t, R) - u_\infty) \quad t > 0
\end{aligned} \quad (29)$$

where $u = u(t, r)$ is the sought function and u_0, R, H, u_∞ are positive constants.

4.1. Implementation of the Standard Approach

We aim to approximate the solution of (29) in all the points (t_i, r_j) of the considered partition using the standard approach. For a fixed i , from (25), we have a system of $N-1$ non-linear equations and $N+1$ unknowns. The boundary conditions in (29) provide the following two additional equations:

$$u_{i,0} - u_{i+1,0} + \lambda_{i,0} \frac{\Delta t}{\Delta r^2} (u_{i+1,1} + u_{i,1} - u_{i+1,0} - u_{i,0}) = 0, \quad (30)$$

$$u_{i,N} - u_{i+1,N} - \frac{H\lambda_{i,N}\Delta t}{2R} (u_{i+1,N} + u_{i,N} - 2u_\infty) + \frac{H^2\gamma_{i,N}\Delta t}{4} (u_{i+1,N} + u_{i,N} - 2u_\infty)^2 \\ + \lambda_{i,N} \frac{\Delta t}{\Delta r^2} (u_{i+1,N-1} + u_{i,N-1} - (1 + \Delta r H)(u_{i+1,N} + u_{i,N}) + 2\Delta r H u_\infty) = 0, \quad (31)$$

for $i \in \{0, 1, \dots\}$. Therefore, for a fixed i , we have a system of $N + 1$ non-linear equations that can be written in vector form as $f_i(u_{i+1,0}, \dots, u_{i+1,N}) = \mathbf{0}$, where $[f_i]_0$ is the left-hand side of (30), $[f_i]_j$ is the left-hand side of (25) for $j \in \{1, \dots, N - 1\}$, and $[f_i]_N$ is the left-hand side of (31). In order to apply the multivariate Newton–Raphson method, we need to compute the Jacobian matrix of f_i at $\mathbf{x} = (x_0, \dots, x_N)$; that is, we need to compute

$$[J_{f_i}(\mathbf{x})]_{h,\ell} = \frac{\partial [f_i]_{h-1}}{\partial x_{\ell-1}}(\mathbf{x}) \quad \forall h, \ell \in \{1, \dots, N + 1\}$$

From (25), (30) and (31), we obtain a tridiagonal matrix whose non-zero entries are given as follows:

$$\begin{aligned} [J_{f_i}(\mathbf{x})]_{1,1} &= -1 - \lambda_{i,0} \frac{\Delta t}{\Delta r^2} \\ [J_{f_i}(\mathbf{x})]_{1,2} &= \lambda_{i,0} \frac{\Delta t}{\Delta r^2} \\ [J_{f_i}(\mathbf{x})]_{j+1,j} &= -\frac{\lambda_{i,j}}{4j} \frac{\Delta t}{\Delta r^2} - \frac{\gamma_{i,j}}{8} \frac{\Delta t}{\Delta r^2} (x_{j+1} - x_{j-1} + u_{i,j+1} - u_{i,j-1}) + \frac{\lambda_{i,j}}{2} \frac{\Delta t}{\Delta r^2} \\ [J_{f_i}(\mathbf{x})]_{j+1,j+1} &= -\lambda_{i,j} \frac{\Delta t}{\Delta r^2} \\ [J_{f_i}(\mathbf{x})]_{j+1,j+2} &= \frac{\lambda_{i,j}}{4j} \frac{\Delta t}{\Delta r^2} + \frac{\gamma_{i,j}}{8} \frac{\Delta t}{\Delta r^2} (x_{j+1} - x_{j-1} + u_{i,j+1} - u_{i,j-1}) + \frac{\lambda_{i,j}}{2} \frac{\Delta t}{\Delta r^2} \\ [J_{f_i}(\mathbf{x})]_{N+1,N} &= \lambda_{i,N} \frac{\Delta t}{\Delta r^2} \\ [J_{f_i}(\mathbf{x})]_{N+1,N+1} &= -1 - \frac{H\lambda_{i,N}\Delta t}{2R} + \frac{H^2\gamma_{i,N}\Delta t}{2} (x_N + u_{i,N} - 2u_\infty) - (1 + \Delta r H)\lambda_{i,N} \frac{\Delta t}{\Delta r^2} \end{aligned} \quad (32)$$

for $j \in \{1, \dots, N - 1\}$.

By taking $\mathbf{x}_0 = (u_{i,0}, \dots, u_{i,N})$ and applying the Newton–Raphson iterative algorithm given in (21), we approximate the values of $(u_{i+1,0}, \dots, u_{i+1,N})$. Using the initial condition, we have that $u_{0,j} = u_0$ for all $j \in \{0, \dots, N\}$. Hence, repeating the aforementioned process for each $i \in \{0, 1, \dots\}$, we obtain the approximation of the solution of (29) in all the points (t_i, r_j) of the partition. This is summarized in Algorithm 1.

Algorithm 1 Algorithm for solving (29) with the standard approach

```

1:  $u_{0,:} \leftarrow u_0$ 
2: for  $i = 0 : i_{\max}$  do  $\triangleright i_{\max}$  is the maximum number of time steps
3:    $\mathbf{x} \leftarrow u_{i,:}$ 
4:   while 1 do
5:     Compute  $f_i(\mathbf{x})$  using (25), (30) and (31)
6:     Compute  $J_{f_i}(\mathbf{x})$  using (32)
7:     Solve  $J_{f_i}(\mathbf{x})\mathbf{h} = f_i(\mathbf{x})$ 
8:      $\mathbf{x} \leftarrow \mathbf{x} - \mathbf{h}$ 
9:     if  $\|\mathbf{h}\| < \epsilon$  then  $\triangleright \epsilon$  sets the accuracy of the Newton–Raphson algorithm
10:       $u_{i+1,:} \leftarrow \mathbf{x}$ 
11:      break
12:    end if
13:  end while
14: end for
```

4.2. Implementation of the Alternative Approach

We here aim to approximate the solution of (29) in all the points (t_i, r_j) of the considered partition using the alternative approach. For a fixed i , from (28), we have a system of $N - 1$ (linear) equations and $N + 1$ unknowns. The boundary conditions in (29) provide the following two additional equations:

$$u_{i,0} - u_{i+1,0} + 2\lambda_{i,0} \frac{\Delta t}{\Delta r^2} (u_{i+1,1} - u_{i+1,0} + u_{i,1} - u_{i,0}) = 0, \quad (33)$$

$$u_{i,N} - u_{i+1,N} - \frac{H\lambda_{i,N}\Delta t}{R} (u_{i,N} - u_\infty) + H^2\gamma_{i,N}\Delta t (u_{i,N} - u_\infty)^2 + \lambda_{i,N} \frac{\Delta t}{\Delta r^2} (u_{i+1,N-1} + u_{i,N-1} - (1 + \Delta r H)(u_{i+1,N} + u_{i,N}) + 2\Delta r H u_\infty) = 0, \quad (34)$$

for $i \in \{0, 1, \dots\}$. Therefore, for a fixed i value, we have a system of $N + 1$ linear equations that can be written in matrix form as $A_i(u_{i+1,0}, \dots, u_{i+1,N})^\top = \mathbf{b}_i$, where $\top$ denotes transpose and A_i is the $(N + 1) \times (N + 1)$ tridiagonal matrix whose non-zero entries are given as follows:

$$\begin{aligned} [A_i]_{1,1} &= 1 + 2\lambda_{i,0} \frac{\Delta t}{\Delta r^2} \\ [A_i]_{1,2} &= -2\lambda_{i,0} \frac{\Delta t}{\Delta r^2} \\ [A_i]_{j+1,j} &= -\frac{\lambda_{i,j}}{2} \frac{\Delta t}{\Delta r^2} \\ [A_i]_{j+1,j+1} &= 1 + \lambda_{i,j} \frac{\Delta t}{\Delta r^2} \\ [A_i]_{j+1,j+2} &= -\frac{\lambda_{i,j}}{2} \frac{\Delta t}{\Delta r^2} \\ [A_i]_{N+1,N} &= -\lambda_{i,N} \frac{\Delta t}{\Delta r^2} \\ [A_i]_{N+1,N+1} &= 1 + \lambda_{i,N} \frac{\Delta t}{\Delta r^2} (1 + \Delta r H) \end{aligned} \quad (35)$$

for $j \in \{1, \dots, N - 1\}$, and $\mathbf{b}_i$ is the $(N + 1) \times 1$ matrix given by

$$\begin{aligned} [\mathbf{b}_i]_{1,1} &= u_{i,0} + 2\lambda_{i,0} \frac{\Delta t}{\Delta r^2} (u_{i,1} - u_{i,0}) \\ [\mathbf{b}_i]_{j+1,1} &= u_{i,j} + \frac{\lambda_{i,j}}{2j} \frac{\Delta t}{\Delta r^2} (u_{i,j+1} - u_{i,j-1}) + \frac{\gamma_{i,j}}{4} \frac{\Delta t}{\Delta r^2} (u_{i,j+1} - u_{i,j-1})^2 \\ &\quad + \frac{\lambda_{i,j}}{2} \frac{\Delta t}{\Delta r^2} (u_{i,j+1} - 2u_{i,j} + u_{i,j-1}) \\ [\mathbf{b}_i]_{N+1,1} &= u_{i,N} - \frac{H\lambda_{i,N}\Delta t}{R} (u_{i,N} - u_\infty) + H^2\gamma_{i,N}\Delta t (u_{i,N} - u_\infty)^2 \\ &\quad + \lambda_{i,N} \frac{\Delta t}{\Delta r^2} (u_{i,N-1} - (1 + \Delta r H)u_{i,N} + 2\Delta r H u_\infty) \end{aligned} \quad (36)$$

for $j \in \{1, \dots, N - 1\}$.

Using the initial condition, we have that $u_{0,j} = u_0$ for all $j \in \{0, \dots, N\}$. Hence, solving this system recursively for each $i \in \{0, 1, \dots\}$, we obtain the approximation of the solution of (29) in all the points (t_i, r_j) of the partition. This is summarized in Algorithm 2.

Algorithm 2 Algorithm for solving (29) with the alternative approach

- 1: $u_{0,:} \leftarrow u_0$
 - 2: **for** $i = 0 : i_{\max}$ **do** $\triangleright i_{\max}$ is the maximum number of time steps
 - 3: Compute A_i using (35)
 - 4: Compute $\mathbf{b}_i$ using (36)
 - 5: Solve $A_i(u_{i+1,0}, \dots, u_{i+1,N})^\top = \mathbf{b}_i$
 - 6: **end for**
-

4.3. Computational Complexity

In this subsection, we compare the computational complexity of the standard and alternative approaches. This analysis focuses on the operations required per time step and the overall computational cost associated with each method.

The standard approach requires solving a system of (non-linear) equations. For each time step i , the Newton–Raphson method should be used, which involves the following components:

- Jacobian Matrix assembly: for each iteration of the Newton–Raphson method, constructing the $(N + 1) \times (N + 1)$ tridiagonal Jacobian coefficient matrix $J_{f_i}(\mathbf{x})$ using (32) involves $3N + 1$ operations.
- For each iteration of the Newton–Raphson method, the computation of $f_i(\mathbf{x})$ using (25), (30) and (31) requires $N + 1$ operations.
- For each iteration of the Newton–Raphson method, solving $J_{f_i}(\mathbf{x})\mathbf{h} = f_i(\mathbf{x})$ requires $5(N + 1)$ multiplications using Thomas' algorithm for tridiagonal matrices.

We consider that s iterations of the Newton–Raphson are required until the desired precision is obtained. Assuming that the initial guess is reasonable, the method converges quadratically, and for typical engineering or scientific problems, $s = 10$ iterations might be enough. Therefore, the total number of operations required for each time step is $s(9N + 7)$ operations.

The alternative approach requires solving a single system of linear equations for each time step i , which involves the following:

- Assembly of the $(N + 1) \times (N + 1)$ tridiagonal coefficient matrix A_i using (35) involves $3N + 1$ operations.
- The computation of $\mathbf{b}_i$ using (36) requires $N + 1$ operations.
- Solving $A_i(u_{i+1,0}, \dots, u_{i+1,N})^\top = \mathbf{b}_i$ requires $5(N + 1)$ multiplications using Thomas' algorithm for tridiagonal matrices.

Therefore, the total number of operations required for each time step is $9N + 7$ operations.

The conclusion is that the alternative approach requires s times fewer operations than the standard approach for each time step. Given that $s = 10$ is a typical number of iterations for the Newton–Raphson method, the alternative approach would require 10 times fewer operations.

5. Numerical Results

In this section, we present a numerical example in which we obtain the approximation of the solution of (29) using both the standard and the alternative approaches presented in this paper. Specifically, we consider the problem of solving the diffusion equation on an infinite length cylindrical antimony bar with Neumann boundary conditions. The diffusion coefficient is described by the Arrhenius equation [38]:

$$\lambda(u) = D_0 e^{-\frac{Q}{\rho u}}$$

where $D_0 = 56 \text{ cm}^2/\text{s}$, $Q = 2.07; 2, 10^5 \text{ J/mol}$, and $\rho = 8.31 \text{ J/K/mol}$ (see (Appendix A in [39])).

We discretize the time domain $[0, 4000]$ into subintervals of length $\Delta t = 1$, and we discretize the space domain $[0, 0.1]$ into subintervals of length $\Delta r = 0.004 \text{ cm}$. Figure 1 shows the approximation of the solution obtained, and Figure 2 shows the relative difference between the approximations provided by the standard and the alternative approach.

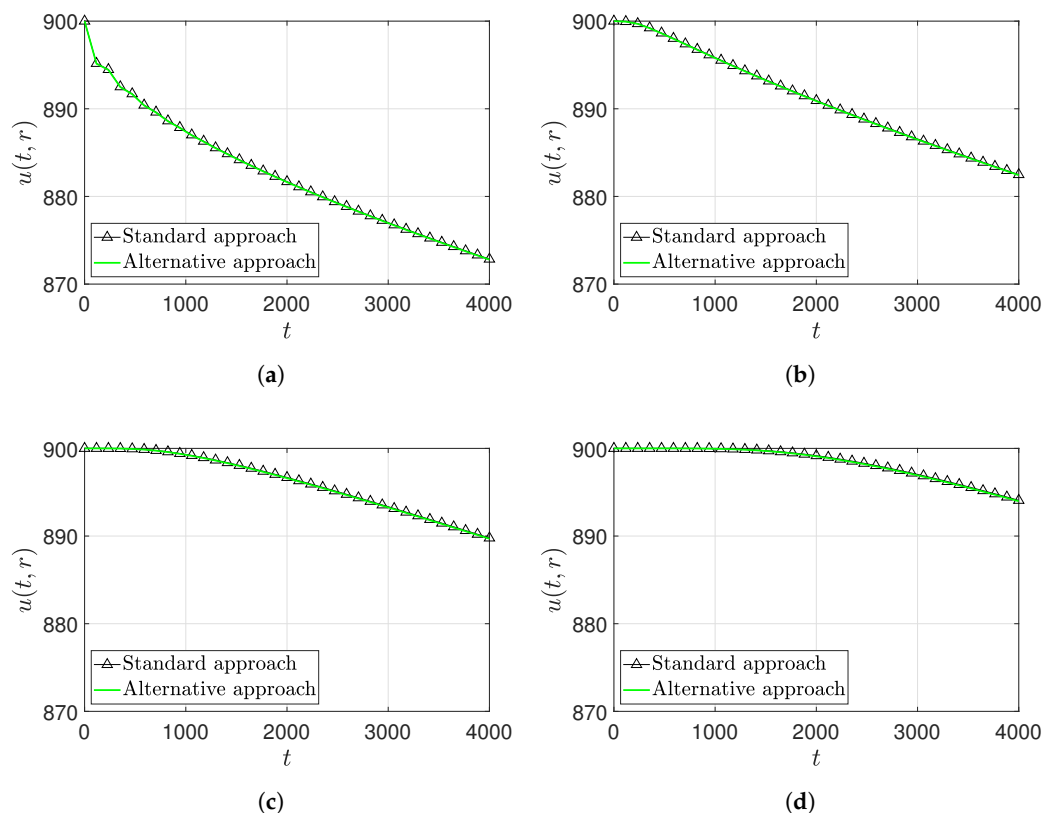


Figure 1. Approximation of the solution of (29) using both the standard and the alternative approaches, with $u_0 = 900$, $u_\infty = 400$, $H = 1$, for $r = 0.1$ (a), $r = 0.08$ (b), $r = 0.05$ (c) and $r = 0$ (d), respectively.

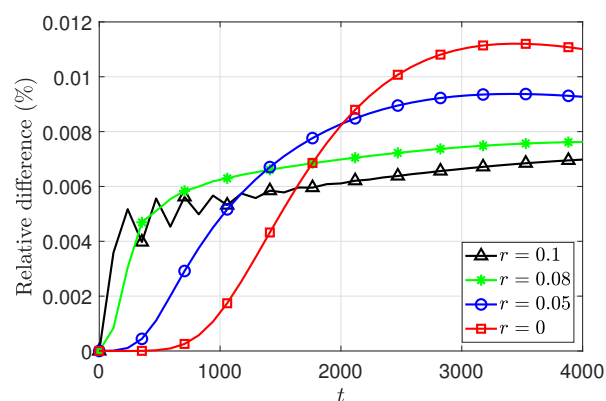


Figure 2. Relative difference between the approximations provided by the standard and the alternative approach.

6. Conclusions

When the diffusion equation has a non-constant diffusion coefficient, numerical methods are the most appropriate to address its resolution. There are many finite difference methods to solve the diffusion equation. In this article, we focused on the Crank–Nicolson method, as it allows for a good approximation. However, the finite difference methods that appear in the literature are used with the assumption that the diffusion coefficient is constant; therefore, the solution is approximated by solving systems of linear equations. In this article, we considered that the diffusion coefficient is not constant; therefore, to approximate the solution, we numerically solved systems of non-linear equations (this is referred to in the article as the standard approach). To reduce the computational complexity involved

in solving non-linear systems of equations, we also proposed an alternative approach that allows for working with systems of linear equations, even though the diffusion coefficient is not constant. This alternative approach considerably reduces the computational complexity with a slight accuracy loss with respect to the standard approach. Specifically, the accuracy of the alternative approach remains the same with respect to the time step but decreases with respect to the spatial step. Hence, this contribution fills a gap in numerical methods for non-linear diffusion problems by offering a solution that combines simplicity and accuracy, which is particularly relevant for large-scale or real-time simulations. Finally, the performance metrics of the standard and alternative approaches are summarized in Table 1, highlighting the differences in computational efficiency and accuracy.

Table 1. Performance metrics of the considered approaches.

Method	Standard Approach	Alternative Approach
Number of operations per time step	$10(9N + 7)$	$9N + 7$
Temporal accuracy	$O(\Delta t^2)$	$O(\Delta t^2)$
Spatial accuracy	$O(\Delta r^2)$	$O(\Delta r)$

Author Contributions: Conceptualization, M.Z.-R. and X.I.; methodology, M.Z.-R. and X.I.; resources, P.F.; writing—original draft preparation, M.Z.-R. and X.I.; writing—review and editing, M.Z.-R., X.I., and P.F.; visualization, M.Z.-R. and X.I.; supervision, M.Z.-R. and X.I.; project administration, M.Z.-R.; funding acquisition, M.Z.-R. and X.I. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Basque Government through the DATAPRIN project (KK-2020/00060).

Data Availability Statement: No new data were created or analyzed in this study.

Conflicts of Interest: Author Patricio Fuentes was employed by the company Photonic Inc. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Cunningham, R.E.; Williams, R.J.J. *Diffusion in Gases and Porous Solids*; Plenum Press: New York, NY, USA, 1980.
2. Reyes, S.C.; Sinfelt, J.H.; DeMartin, G.J. Diffusion in Porous Solids: The Parallel Contribution of Gas and Surface Diffusion Processes in Pores Extending from the Mesoporous Region into the Microporous Region. *J. Phys. Chem. B* **2000**, *104*, 5750–5761. [\[CrossRef\]](#)
3. Balogh, Z.; Schmitz, G. Diffusion in Metals and Alloys. In *Physical Metallurgy*, 5th ed.; Elsevier: Amsterdam, The Netherlands, 2014; pp. 387–559. [\[CrossRef\]](#)
4. Nagasawa, M. *Schrodinger Equations and Diffusion Theory*; Birkhauser Verlag: Basel, Switzerland, 1993. [\[CrossRef\]](#)
5. Dynkin, E.B. *Diffusions, Superdiffusions and Partial Differential Equations*; American Mathematical Society: Providence, RI, USA, 2002.
6. Möller, M.N.; Cuevasanta, E.; Orrico, F.; Lopez, A.C.; Thomson, L.; Denicola, A. Diffusion and Transport of Reactive Species Across Cell Membranes. *Adv. Exp. Med. Biol.* **2019**, *1127*, 3–19. [\[CrossRef\]](#) [\[PubMed\]](#)
7. Willoughby, A.F.W. Atomic diffusion in semiconductors. *Rep. Prog. Phys.* **1978**, *41*, 1665. [\[CrossRef\]](#)
8. Giddings, J. ‘Eddy’ Diffusion in Chromatography. *Nature* **1959**, *184*, 357–358. [\[CrossRef\]](#)
9. Karaiskakis, G.; Gavril, D. Determination of diffusion coefficients by gas chromatography. *J. Chromatogr. A* **2004**, *1037*, 147–189. [\[CrossRef\]](#)
10. Abbasbandy, S. Approximate solution for the nonlinear model of diffusion and reaction in porous catalysts by means of the homotopy analysis method. *Chem. Eng. J.* **2008**, *136*, 144–150. [\[CrossRef\]](#)
11. Potter, M.E.; O’Malley, A.J.; Chapman, S.; Kezina, J.; Newland, S.H.; Silverwood, I.P.; Mukhopadhyay, S.; Carravetta, M.; Mezza, T.M.; Parker, S.F.; et al. Understanding the Role of Molecular Diffusion and Catalytic Selectivity in Liquid-Phase Beckmann Rearrangement. *ACS Catal* **2017**, *7*, 2926–2934. [\[CrossRef\]](#)

12. Liu, Z.H.; Zhang, D.Q.; Sing, S.L.; Chua, C.K.; Loh, L.E. Interfacial characterization of SLM parts in multi-material processing: Metallurgical diffusion between 316L stainless steel and C18400 copper alloy. *Mater. Charact.* **2014**, *94*, 116–125. [\[CrossRef\]](#)
13. Hussain, A.; Uddin, M.; Haq, S.; Jan, H.U. Numerical Solution of Heat Equation in Polar Cylindrical Coordinates by the Meshless Method of Lines. *J. Math.* **2021**, *2021*, 8862139. [\[CrossRef\]](#)
14. Munier, A.; Burgan, J.R.; Gutierrez, J.; Fijalkow, E.; Feix, M.R. Group Transformations and the Nonlinear Heat Diffusion Equation. *SIAM J. Appl. Math.* **1981**, *40*, 191–207. [\[CrossRef\]](#)
15. Paripour, M.; Babolian, E.; Saeidian, J. Analytic solutions to diffusion equations. *Math. Comput. Model.* **2010**, *51*, 649–657. [\[CrossRef\]](#)
16. Suárez-Carreño, F.; Rosales-Romero, L. Convergency and Stability of Explicit and Implicit Schemes in the Simulation of the Heat Equation. *Appl. Sci.* **2021**, *11*, 4468. [\[CrossRef\]](#)
17. Singh, S.; Jain, P.K.; Rizwan-uddin. Analytical solution to transient heat conduction in polar coordinates with multiple layers in radial direction. *Int. J. Therm. Sci.* **2007**, *47*, 261–273. [\[CrossRef\]](#)
18. Santos, L.P.; Junior, J.O.M.; Campos, M.D.; Romao, E.C. A study about one-dimensional steady state heat transfer in cylindrical and spherical coordinates. *Appl. Math. Sci.* **2013**, *7*, 6227–6233. [\[CrossRef\]](#)
19. Crank, J. *The Mathematics of Diffusion*; Oxford University Press: Oxford, UK, 1979.
20. Ivanova, N.M. Exact solutions of diffusion–convection equations. *Dyn. Partial Differ. Equ.* **2008**, *5*, 139–171. [\[CrossRef\]](#)
21. Hayek, M. An exact solution for a nonlinear diffusion equation in a radially symmetric inhomogeneous medium. *Comput. Math. Appl.* **2014**, *68*, 1751–1757. [\[CrossRef\]](#)
22. Chang, M.J.; Chow, L.C.; Chang, W.S. Improved alternating-direction implicit method for solving transient three-dimensional heat diffusion problems. *Numer. Heat Transf. Part B Fundam.* **1991**, *19*, 69–84. [\[CrossRef\]](#)
23. Thibault, J. Comparison of nine three-dimensional numerical methods for the solution of the heat diffusion equation. *Numer. Heat Transf.* **1985**, *8*, 281–298. [\[CrossRef\]](#)
24. Crank, J.; Nicolson, P. A Practical Method for Numerical Evaluation of Solutions of Partial Differential Equations of the Heat Conduction Type. *Adv. Comput. Math.* **1996**, *6*, 207–226. [\[CrossRef\]](#)
25. Liu, Y.; Bi, H. Stability analysis and error estimation of variable coefficient convection-diffusion equation: Generalized numerical fluxes. *J. Phys. Conf. Ser.* **2021**, *1903*, 012016. [\[CrossRef\]](#)
26. Baranovskii, E.S.; Brizitskii, R.V.; Saritskaia, Z.Y. Optimal control problems for the reaction-diffusion-convection equation with variable coefficients. *Nonlinear Anal. Real World Appl.* **2024**, *75*, 103979. [\[CrossRef\]](#)
27. Nikan, O.; Avazzadeh, Z. An efficient localized meshless technique for approximating nonlinear sinh-Gordon equation arising in surface theory. *Eng. Anal. Bound. Elem.* **2021**, *130*, 268–285. [\[CrossRef\]](#)
28. Joy, D.; Kumar, S.D. Computational techniques for singularly perturbed reaction-diffusion delay differential equations: A second-order approach. *J. Math. Comput. Sci.* **2024**, *35*, 304–318. [\[CrossRef\]](#)
29. Balsa-Canto, E.; Vázquez, C.; López-Núñez, A. Numerical methods for a nonlinear reaction–diffusion system modelling a batch culture of biofilm. *Appl. Math. Model.* **2017**, *41*, 164–179. [\[CrossRef\]](#)
30. Papanicolaou, A.N.; Diplas, P. Numerical solution of a non-linear model for self-weight solids settlement. *Appl. Math. Model.* **1999**, *23*, 345–362. [\[CrossRef\]](#)
31. Apostol, T.M. *Mathematical Analysis*; Addison-Wesley: Boston, MA, USA, 1974.
32. Smith, G.D. *Numerical Solution of Partial Differential Equations: Finite Difference Methods*; Oxford University Press: Oxford, UK, 1978; ISBN 0-19-859625-1.
33. Sharma, T.C.; Pathak, S.P.; Trivedi, G. Comparative Study of Crank-Nicolson and Modified Crank-Nicolson Numerical Methods to Solve Linear Partial Differential Equations. *Indian J. Sci. Technol.* **2024**, *17*, 924–931. [\[CrossRef\]](#)
34. Wang, J.; Li, H.; Ren, X.; Chang, X. The Crank-Nicolson Mixed Finite Element Scheme and Its Reduced-Order Extrapolation Model for the Fourth-Order Nonlinear Diffusion Equations with Temporal Fractional Derivative. *Fractal Fract.* **2025**, *9*, 789. [\[CrossRef\]](#)
35. Dahlquist, G.; Björck, A. *Numerical Methods*; Dover: Kent, UK, 2003; ISBN 0-486-42807-9.
36. Burger, J.; Machbub, C. Comparison of numerical solutions of a one-dimensional non-linear heat equation. *Commun. Appl. Numer. Methods* **1991**, *7*, 233–240. [\[CrossRef\]](#)
37. Albasiny, E.L. On the numerical solution of a cylindrical heat-conduction problem. *Q. J. Mech. Appl. Math.* **1960**, *13*, 374–384. [\[CrossRef\]](#)
38. Peterson, N.L. Self-diffusion in pure metals. *J. Nucl. Mater.* **1978**, *69–70*, 3–37. [\[CrossRef\]](#)
39. François, D.; Pineau, A.; Zaoui, A. *Mechanical Behaviour of Materials*; Springer: Dordrecht, The Netherlands, 2013. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.