Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A meshfree CSRBF–ETDRDP scheme for solving systems of semilinear parabolic partial differential equations

Ibraheem Abiodun Yahayah
Clarkson University
Emmanuel Asante-Asamani
easantea@clarkson.edu
Clarkson University
Guangming Yao
Clarkson University

Research Article

Keywords: Compactly supported radial basis function, Exponential time differencing with real and distinct poles, Parabolic PDEs, Support radius, Irregular domains

Posted Date: November 17th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11692072/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

A meshfree CSRBF–ETDRDP scheme for solving systems of semilinear parabolic partial differential equations

Ibraheem Abiodun Yahayah[a], Emmanuel Asante-Asamani[a,*], Guangming Yao[a]

[a] *Clarkson University, 8 Clarkson Avenue, Potsdam, 13699, NY, USA*

Abstract

A second-order meshfree compactly supported radial basis function is combined with a real and distinct poles exponential time differencing scheme (CSRBF–ETDRDP) to efficiently solve one- and two-dimensional, nonlinear parabolic PDEs with irregular domains. The method employs meshless collocation method for spatial discretization, enabling flexible treatment of irregular geometries while maintaining good accuracy. The ETDRDP framework provides an efficient L-stable time integration for stiff problems without requiring nonlinear iterative solvers. This combination effectively handles nonlinear source terms and complex boundary geometries. We evaluate the performance of the proposed CSRBF–ETDRDP scheme by comparing its accuracy, convergence rate, and computational efficiency against other time integration methods, including the implicit Crank–Nicolson (CSRBF-CN) and explicit Runge-Kutta (CSRBF-ERK2) methods. We consider several linear and non-linear parabolic problems with Dirichlet and Neumann boundary conditions. The CSRBF-ETDRDP method achieves second order temporal accuracy while offering improved computational efficiency compared with the methods considered.

Keywords: Compactly supported radial basis function, Exponential time differencing with real and distinct poles, Parabolic PDEs, Support radius, Irregular domains
2020 MSC: 65M06, 65M20, 65M22

1. Introduction

Semilinear parabolic PDEs arise in numerous applications, such as the Fisher equation for population dynamics [1], the Allen-Cahn equation for phase transitions [2], and pattern formation in biology [3]. Many existing numerical methods for solving these types of PDEs rely heavily on implicit time-stepping schemes such as Backward Euler or Crank–Nicolson and the nonlinear reaction term is typically handled with nonlinear solvers such as Newton's method or Picard iteration. For stiff problems, these schemes are stable and robust but require multiple nonlinear iterations at each time step which increases the computational cost. For problems in high-dimensional, non-rectangular domains, traditional mesh-based methods such as finite difference and finite element pose several challenges in imposing boundary conditions or constructing computational meshes. Consequently, the development of numerical schemes that are both spatially flexible and computationally efficient remains an active area of research.

Radial basis function (RBF) approximation has become one of the most successful meshfree techniques for scattered data interpolation and the numerical solution of partial differential equations [4, 5]. Unlike mesh-based methods, RBF collocation constructs multidimensional approximations directly from scattered centers without requiring element connectivity, making it particularly

*easantea@clarkson.edu

well suited for problems posed on irregular domains or in higher dimensions [6, 7]. Over the past three decades, RBF methods have been successfully applied to elliptic, parabolic, hyperbolic and integro-differential equations, where they offer high-order accuracy together with geometric flexibility. Classical RBF methods often employ globally supported kernels, meaning that every basis function has nonzero influence throughout the computational domain. Consequently, the approximation at any evaluation point depends on all collocation centers. Common globally supported kernels include the Gaussian, multiquadric (MQ), inverse multiquadric (IMQ), inverse quadratic (IQ), polyharmonic spline (PHS), and Matérn families, all of which have been successfully applied to scattered data interpolation and the numerical solution of partial differential equations.

The principal advantage of globally supported kernels lies in their excellent approximation properties. Depending on the choice of kernel and the regularity of the underlying function, they often exhibit high-order or spectral convergence and can achieve remarkable accuracy with relatively few degrees of freedom. These favorable approximation properties have led to their successful application in a broad range of meshfree numerical methods. They have been combined with a number of time-discretization schemes to solve several nonlinear PDEs [8, 9, 10, 11, 12, 13, 14]. The global nature of these basis functions, however, also gives rise to their main computational limitation. Since every basis function interacts with every collocation point, the interpolation matrix is dense. As the number of centers increases, the storage requirements and computational cost grow rapidly, while the interpolation matrix may become increasingly ill-conditioned [15]. These difficulties become particularly pronounced in large-scale simulations, where repeated matrix operations dominate the overall computational expense.

For time-dependent partial differential equations, the dense differentiation matrices produced by globally supported RBFs significantly increase the cost of each time step, making large-scale computations less attractive. Consequently, considerable effort has been devoted to developing localized RBF approximations that preserve the geometric flexibility of meshfree collocation while reducing computational complexity. To generate sparse matrices, researchers have also developed localized methods such as the modified localized method of approximated particular solutions (MLMAPS), which employs integrated polyharmonic splines (PS) augmented with polynomial bases [16, 17], RBF-generated finite differences (RBF-FD) using MQ, and hybrid Gaussian–cubic kernels [18, 19], and the partition of unity method (PUM) [20, 21], which restrict interactions to local stencils or subdomains. However, these methods may involve additional preprocessing, such as nearest-neighbor searches, [19] patch construction and boundary tracking [22].

Compactly Supported Radial Basis Functions (CSRBFs) provide an alternative approach to generate sparse interpolation matrices directly at the basis level. They were introduced through the work of Wu, Wendland, and others [5, 23, 24, 25]. Unlike globally supported kernels, CSRBFs vanish outside a finite support, producing sparse interpolation matrices while retaining the meshfree character of RBF approximation. Consequently, CSRBF methods are particularly attractive for time-dependent partial differential equations, where sparse spatial discretization matrices can substantially reduce storage requirements and computational cost [26, 27]. Among the various compactly supported kernels, Wendland's functions are particularly well suited for meshfree collocation because they possess minimal polynomial degree for a prescribed level of smoothness while remaining strictly positive definite in the appropriate spatial dimensions. These properties ensure a well-posed interpolation problem for distinct collocation centers. Combined with their compact support, they yield sparse interpolation matrices suitable for large-scale meshfree computations [28, 29]. CSRBFs have been combined with various numerical schemes, including finite difference methods to solve shallow water hydrodynamics equations [30] and Stokes flow problems [31], explicit Euler methods for certain inverse problems [32], and more recently, RK4 for fluid dynamics PDEs [33]. Additionally, CSRBFs have been used for the spatial discretization of Poisson's equation [34] and

higher-order PDEs [35] in 3D.

Exponential Time Differencing (ETD) methods represent a powerful class of time integrators for solving stiff differential equations. Their main advantage lies in treating the stiff linear part through matrix exponential actions while handling the nonlinear part explicitly, often through predictor–corrector formulations. Unlike traditional explicit methods with stringent time-step restrictions, this approach can permit larger time steps and improve numerical stability. Consequently, exponential integrators have become increasingly popular for the numerical simulation of complex physical and biological systems, where computational efficiency and numerical accuracy are equally important. An extensive overview of their theoretical framework, implementation, and applications can be found in [36]. Existing studies have successfully combined RBF methods with ETD schemes using different spatial formulations and specific applications. For instance, [37] combined an MQ-based RBF-FD discretization with an ETDRK4 scheme for a two-dimensional damped Kuramoto–Sivashinsky equation, while [38] employed a compact local integrated RBF formulation with ETDRK4 for a nonlinear advection-diffusion-reaction system. More recently, localized RBF collocation methods (LRBFCM) have been coupled with first- and second-order ETD schemes to solve multidimensional neural-field models with transmission delays [39].

The closest work combining CSRBF with an ETD scheme employed a third-order exponential Rosenbrock method (exprb32) with adaptive spatial and temporal discretizations [40]. In their adaptive spatial construction, the authors noted that the Delaunay-based checkpoint strategy can require substantial memory, particularly in higher dimensions. Furthermore, the exponential Rosenbrock implementation requires repeated evaluations of the Jacobian operator during Newton interpolation at Leja points, making matrix-function evaluations the dominant computational cost. ETD schemes of Runge–Kutta type (ETDRK), introduced by Cox and Matthews [41] and later improved by Kassam and Trefethen [42], do not require the evaluation of the Jacobian operator and thus tend to be more computationally efficient and often avoid repeated Jacobian evaluations for problems with a fixed linear operator, potentially reducing computational cost. Recently, a new class of ETDRK schemes has been developed [43, 44, 45] that uses L-acceptable rational approximations with real and distinct poles (RDP) to simplify matrix exponential evaluations in ETDRK schemes. These schemes have several advantages. Their use of rational functions to approximate the matrix exponential reduces their implementation to a sequence of linear solves. The L-acceptability of the rational approximations provides strong damping of stiff modes, which can help suppress spurious oscillations associated with nonsmooth data. Finally, the use of RDP rational approximations eliminates complex arithmetic in the scheme, potentially reducing computational cost and simplifying implementation in lower-level programming languages compared with approaches involving complex poles [46, 47, 48, 49].

Our work evaluates the performance of CSRBF in combination with a second-order ETDRK scheme based on an RDP rational approximation (ETDRDP) [43]. We show that the combined CSRBF–ETDRDP scheme is computationally efficient and takes advantage of the sparsity of CSRBF discretization matrices, reducing memory requirements and enabling efficient sparse linear solves. The method transforms the PDE into a system of ordinary differential equations (ODEs) using CSRBF and then integrates the resulting semidiscrete system using ETDRDP. We compare the proposed scheme with other hybrid schemes combining CSRBF spatial discretization with established time-stepping methods, including ERK2 and Crank–Nicolson (CN). The convergence rates are examined numerically with various test problems.

The remainder of the paper is organized as follows. Section 2 introduces CSRBF interpolation, while Section 3 describes the spatial discretization of the PDE and the resulting system of ODEs. The ETDRDP scheme is presented in Section 4, and numerical comparisons are presented in Section 5.

2. Radial Basis Function Interpolation

Let $X = \{\mathbf{x}_j\}_{j=1}^N \subset \Omega$ be a set of distinct centers in the computational domain $\Omega \subset \mathbb{R}^d$. The RBF approximation of a sufficiently smooth function u is given by

$$\hat{u}(\mathbf{x}) = \sum_{j=1}^{N} \alpha_j \phi(\|\mathbf{x} - \mathbf{x}_j\|),$$

where ϕ is a radial kernel and the coefficients $\{\alpha_j\}_{j=1}^N$ are determined by interpolation or collocation. Different choices of the radial kernel lead to different approximation properties and computational characteristics [28, 4, 5]. For the numerical solution of large-scale PDEs, the distinction between globally and compactly supported kernels, discussed in the introduction, is particularly important and forms the basis of the present work.

2.1. Compactly Supported Radial Basis Functions

In this work, we employ the Wendland C^2 function

$$\phi(r) = (1-r)_+^4(4r+1),$$

where $(1-r)_+ = \max(1-r, 0)$. The Wendland C^2 function possesses two continuous derivatives and is strictly positive definite in spatial dimensions up to three [5]. Since the governing equations considered in this work involve second-order spatial derivatives, the C^2 kernel provides sufficient smoothness while maintaining compact support, sparse interpolation and derivative evaluation matrices.

2.2. Support Radius

In practical implementations, the Wendland basis is scaled by a positive parameter δ according to $\phi_\delta(r) = \phi(\delta r)$. The corresponding support radius is $\varepsilon = \frac{1}{\delta}$. For the Wendland C^2 function used in this work, the scaled kernel may be written as

$$\phi_\delta(r) = v^4(5-4v), \qquad v = \max(1-\delta r, 0).$$

The support radius determines the locality of the approximation by controlling the number of neighboring centers that contribute to the interpolant at a given point [5, 50]. Since $\phi_\delta(r) = 0$ for $r > \varepsilon$, only centers lying within the support radius contribute to the local approximation. Consequently, the interpolation and derivative evaluation matrices are sparse, although their density increases as ε increases.

The choice of support radius plays a fundamental role in the performance of CSRBF methods. Smaller support radii increase the sparsity of the interpolation matrix and thus reduce computational cost, but may lead to poor approximation accuracy, rank deficiency, or numerical instability if too few neighboring centers are included. Conversely, a larger support radius increases the overlap among basis functions and may improve approximation accuracy, but it also produces denser matrices which increase the computational cost [28, 29]. The support radius therefore governs the balance among locality, accuracy, conditioning, and efficiency. The influence of ε on the accuracy, conditioning, convergence, and computational efficiency of the proposed CSRBF–ETDRDP method is therefore examined systematically in the numerical experiments.

3. CSRBF Spatial Discretization

Consider the general system of q coupled semilinear PDEs

$$\frac{\partial u_p}{\partial t} = \mathcal{L}u_p + \mathcal{N}_p(\mathbf{u}, t), \quad \mathbf{x} \in \Omega \subset \mathbb{R}^d, \ t \in (0, T] \tag{1}$$

for $p = 1, \ldots, q$ and $d \geq 2$, subject to the boundary condition $\mathbf{u}(\mathbf{x}, t) = \mathbf{g}(\mathbf{x}, t)$ for $\mathbf{x} \in \partial\Omega$, with some initial condition $\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x})$. Here, $\mathcal{L}$ is a linear differential operator, $\mathcal{N}$ is the nonlinear source term and $\mathbf{u} = (u_1, u_2, ..., , u_q)^T$ is the vector of unknowns. Given N_I interior nodes in Ω and N_B nodes on its boundary $\partial\Omega$, the solution $\mathbf{u}(\mathbf{x}, t) = \begin{pmatrix} u_1(\mathbf{x}, t) & \cdots & u_q(\mathbf{x}, t) \end{pmatrix}^\top \in \mathbb{R}^q$ is first approximated using CSRBFs centered at scattered nodes $\mathbf{x}_j$, $j = 1, \ldots, N$, where $N = N_I + N_B$. Hence, we can write

$$\mathbf{u}(\mathbf{x}, t) \approx \sum_{j=1}^{N} \boldsymbol{\alpha}_j(t) \phi \left(\|\mathbf{x} - \mathbf{x}_j\| \right),$$

where $\boldsymbol{\alpha}_j(t) \in \mathbb{R}^q$ is a q-component vector of coefficients for each node.

3.1. Collocation Equations

For a clearer presentation of our formulation, we first present the discretization for a scalar equation ($q = 1$) in d spatial dimensions. This is extended to multiple components in a later section. To begin, we enforce the PDE at the interior collocation points $\mathbf{x}_i$, $i = 1, \ldots, N_I$, and the boundary conditions at the boundary points $\mathbf{x}_i$, $i = N_I + 1, \ldots, N$ as follows. At each interior point $\mathbf{x}_i$, we require

$$\frac{\partial u(\mathbf{x}_i, t)}{\partial t} = \mathcal{L}u(\mathbf{x}_i, t) + \mathcal{N}(u(\mathbf{x}_i, t), t). \tag{2}$$

Substituting the RBF interpolant,

$$u(\mathbf{x}_i, t) \approx \sum_{j=1}^{N} \alpha_j(t) \phi_{ij}, \tag{3}$$

where $\phi_{ij} = \phi \left(\|\mathbf{x}_i - \mathbf{x}_j\| \right)$, its temporal derivative,

$$\frac{\partial u(\mathbf{x}_i, t)}{\partial t} \approx \sum_{j=1}^{N} \dot{\alpha}_j(t) \phi_{ij}. \tag{4}$$

and the corresponding approximation of the differential operator,

$$\mathcal{L}u(\mathbf{x}_i, t) \approx \sum_{j=1}^{N} \alpha_j(t) \mathcal{L}\phi_{ij}, \tag{5}$$

with $\mathcal{L}\phi_{ij} = \mathcal{L}\phi \left(\|\mathbf{x}_i - \mathbf{x}_j\| \right)$ into the PDE equation (2), we obtain the collocation equation

$$\sum_{j=1}^{N} \dot{\alpha}_j(t) \phi_{ij} = \sum_{j=1}^{N} \alpha_j(t) \mathcal{L}\phi_{ij} + \mathcal{N} \left(\sum_{j=1}^{N} \alpha_j(t) \phi_{ij}, t \right), \quad i = 1, \ldots, N_I. \tag{6}$$

It remains to derive the collocation equations for the boundary conditions. First, we consider Dirichlet boundary conditions, followed by Neumann boundary conditions. In each case, our resulting collocation equations will define a system of ODEs which can be solved with an appropriate time discretization scheme.

3.2. Dirichlet Boundary Condition

Consider the Dirichlet boundary condition $u(\mathbf{x}, t) = g(\mathbf{x}, t)$. At each boundary point $\mathbf{x}_i$, we impose

$$u(\mathbf{x}_i, t) = g(\mathbf{x}_i, t),$$

which can be approximated using the boundary collocation equations

$$\sum_{j=1}^{N} \alpha_j(t)\, \phi_{ij} = g(\mathbf{x}_i, t), \quad i = N_I + 1, \ldots, N. \tag{7}$$

We will make extensive use of the following matrices and vectors in deriving our system of ODEs,

$$\alpha(t) = \begin{bmatrix} \alpha_1(t) \\ \alpha_2(t) \\ \vdots \\ \alpha_N(t) \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1N} \\ \phi_{21} & \phi_{22} & \cdots & \phi_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{N1} & \phi_{N2} & \cdots & \phi_{NN} \end{bmatrix}, \quad \mathbf{L_I} = \begin{bmatrix} \mathcal{L}\phi_{11} & \mathcal{L}\phi_{12} & \cdots & \mathcal{L}\phi_{1N} \\ \mathcal{L}\phi_{21} & \mathcal{L}\phi_{22} & \cdots & \mathcal{L}\phi_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{L}\phi_{N_I I} & \mathcal{L}\phi_{N_I 2} & \cdots & \mathcal{L}\phi_{N_I N} \end{bmatrix}, \quad \mathbf{N}(u_I) = \begin{bmatrix} \mathcal{N}(u(\mathbf{x}_1, t), t) \\ \mathcal{N}(u(\mathbf{x}_2, t), t) \\ \vdots \\ \mathcal{N}(u(\mathbf{x}_{N_I}, t), t) \end{bmatrix}$$

Here, $\alpha(t) \in \mathbb{R}^N$ is the vector of unknown coefficients, $\mathbf{R} \in \mathbb{R}^{N \times N}$ is the CSRBF interpolation matrix for both interior and boundary nodes, and $\mathbf{L_I} \in \mathbb{R}^{N_I \times N}$ and $\mathbf{N}(u_I) \in \mathbb{R}^{N_I}$ are the differentiation operator matrix and the nonlinear source vector, both evaluated at the interior nodes, respectively.

3.2.1. ODE system for Dirichlet Boundary

We can express equation (3) more compactly by defining the nodal solution vector

$$\hat{u}(t) = [\hat{u}(\mathbf{x}_1, t), \ldots, \hat{u}(\mathbf{x}_N, t)]^T = \sum_{j=1}^{N} \alpha_j(t)\phi_{ij} = \mathbf{R}\alpha(t)$$

Since $\mathbf{R}$ is the CSRBF interpolation matrix, and the Wendland C^2 kernel is strictly positive definite in spatial dimensions up to three [5], $\mathbf{R}$ is invertible for distinct collocation centers. Thus, we can write

$$\alpha(t) = \mathbf{R}^{-1}\hat{u}(t)$$

Similarly, we can rewrite equation (5) in terms of the differential operator matrix $\mathbf{L_I}$ as

$$\sum_{j=1}^{N} \alpha_j(t)\mathcal{L}\phi_{ij} = \mathbf{L_I}\alpha(t) = \mathbf{L_I}\mathbf{R}^{-1}\hat{u}(t)$$

Define $\mathbf{L} = \mathbf{L_I}\mathbf{R}^{-1}$ so that $\mathbf{L} \in \mathbb{R}^{N_I \times N}$. If we partition the solution vector as $\hat{u} = [\hat{u}_I; \hat{u}_B]$, we can partition $\mathbf{L}$ into interior and boundary blocks as

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_{II} & \mathbf{L}_{IB} \end{bmatrix} \tag{8}$$

where $\mathbf{L}_{II} \in \mathbb{R}^{N_I \times N_I}$ and $\mathbf{L}_{IB} \in \mathbb{R}^{N_I \times N_B}$ map the interior nodal values and the boundary nodal values to the interior differential operator respectively. With the Dirichlet boundary condition (7), the boundary values $\hat{u}_B$ are explicitly prescribed by the boundary data:

$$\hat{u}_B(t) = g(t),$$

where $g(t) = [g(\mathbf{x}_{N_I+1}, t), \ldots, g(\mathbf{x}_N, t)]^T$.

3.2.2. Reduction to Interior Variables

Applying the operator partitioning in (8), the interior collocation equation (6) can be expressed in the semidiscrete form

$$\dot{\hat{u}}_I = \mathbf{L}_{II}\hat{u}_I + \mathbf{L}_{IB}\hat{u}_B + \mathbf{N}(\hat{u}_I).$$

Substituting the prescribed boundary values $\hat{u}_B = g(t)$ into the interior equation, we obtain

$$\dot{\hat{u}}_I = \mathbf{L}_{II}\hat{u}_I + \mathbf{L}_{IB}g(t) + \mathbf{N}(\hat{u}_I).$$

Setting $\mathbf{A} = \mathbf{L}_{II}$ and $\mathbf{F}(\hat{u}_I, t) = \mathbf{L}_{IB}g(t) + \mathbf{N}(\hat{u}_I)$, we obtain the following reduced ODE system for the interior nodes

$$\frac{d\hat{u}_I}{dt} = \mathbf{A}\hat{u}_I + \mathbf{F}(\hat{u}_I, t). \tag{9}$$

3.3. Neumann Boundary Condition

Consider the Neumann boundary condition $\nabla u \cdot \mathbf{n} = h(\mathbf{x}, t)$. We enforce this at all boundary nodes as

$$\sum_{j=1}^{N} \alpha_j(t)\, \nabla\phi_{ij} \cdot \mathbf{n} = h(\mathbf{x}_i, t), \quad i = N_{I+1}, \cdots, N \tag{10}$$

By stacking the spatial differential operator rows from the collocation equation (6) for the interior nodes over the directional derivative rows from (10) for the boundary nodes, we define the matrix $\mathbf{Z} \in \mathbb{R}^{N\times N}$ as

$$\mathbf{Z} = \begin{bmatrix} \mathcal{L}\phi_{11} & \mathcal{L}\phi_{12} & \cdots & \mathcal{L}\phi_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{L}\phi_{N_{I1}} & \mathcal{L}\phi_{N_{I2}} & \cdots & \mathcal{L}\phi_{N_{IN}} \\ \nabla^{\mathbf{n}}\phi_{N_{I+1,\,1}} & \nabla^{\mathbf{n}}\phi_{N_{I+1,\,2}} & \cdots & \nabla^{\mathbf{n}}\phi_{N_{I+1,\,N}} \\ \vdots & \vdots & \ddots & \vdots \\ \nabla^{\mathbf{n}}\phi_{N,1} & \nabla^{\mathbf{n}}\phi_{N,2} & \cdots & \nabla^{\mathbf{n}}\phi_{N,N} \end{bmatrix},$$

where $\nabla^{\mathbf{n}}\phi_{ij} = \nabla\phi(\|\mathbf{x}_i - \mathbf{x}_j\|) \cdot \mathbf{n}(\mathbf{x}_i)$. Using $\alpha(t) = \mathbf{R}^{-1}\hat{u}(t)$, we can write (5) in terms of $\mathbf{Z}$ as:

$$\sum_{j=1}^{N} \alpha_j(t)\mathcal{L}\phi_{ij} = \mathbf{Z}\alpha(t) = \mathbf{Z}\mathbf{R}^{-1}\hat{u}(t).$$

Define $\mathbf{K} = \mathbf{Z}\mathbf{R}^{-1}$ so that $\mathbf{K} \in \mathbb{R}^{N\times N}$. With the partitioned solution vector as $\hat{u} = [\hat{u}_I; \hat{u}_B]$, we can partition $\mathbf{K}$ into interior and boundary blocks as

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{II} & \mathbf{K}_{IB} \\ \mathbf{K}_{BI} & \mathbf{K}_{BB} \end{bmatrix}, \tag{11}$$

where $\mathbf{K}_{II} \in \mathbb{R}^{N_I\times N_I}$ and $\mathbf{K}_{IB} \in \mathbb{R}^{N_I\times N_B}$ map the nodal values to the interior differential operator, while $\mathbf{K}_{BI} \in \mathbb{R}^{N_B\times N_I}$ and $\mathbf{K}_{BB} \in \mathbb{R}^{N_B\times N_B}$ map the nodal values to the boundary normal derivatives. The Neumann boundary condition at any time t provides the algebraic constraint

$$\mathbf{K}_{BI}\hat{u}_I + \mathbf{K}_{BB}\hat{u}_B = h(t),$$

where $h(t) = [h(\mathbf{x}_{N_I+1}, t), \ldots, h(\mathbf{x}_N, t)]^T$. Solving for the boundary unknowns yields,

$$\hat{u}_B = \mathbf{K}_{BB}^{-1}\left(h(t) - \mathbf{K}_{BI}\hat{u}_I\right). \tag{12}$$

Solving for $\hat{u}_B$ requires an invertible matrix $\mathbf{K}_{BB}$. This matrix is non-singular when boundary collocation points are sufficiently well-distributed and support radius is properly selected.

3.3.1. Reduction to Interior Variables

The interior equation (6) results in the semidiscrete form:

$$\dot{\hat{u}}_I = \mathbf{K}_{II}\hat{u}_I + \mathbf{K}_{IB}\hat{u}_B + \mathbf{N}(\hat{u}_I). \tag{13}$$

Substituting the expression for $\hat{u}_B$ in (12) into the interior system (13) yields

$$\dot{\hat{u}}_I = \left(\mathbf{K}_{II} - \mathbf{K}_{IB}\mathbf{K}_{BB}^{-1}\mathbf{K}_{BI}\right)\hat{u}_I + \mathbf{K}_{IB}\mathbf{K}_{BB}^{-1}h(t) + \mathbf{N}(\hat{u}_I).$$

Now, define $\mathbf{A} = \mathbf{K}_{II} - \mathbf{K}_{IB}\mathbf{K}_{BB}^{-1}\mathbf{K}_{BI}$ and $\mathbf{F}(\hat{u}_I, t) = \mathbf{K}_{IB}\mathbf{K}_{BB}^{-1}h(t) + \mathbf{N}(\hat{u}_I)$, so that the resulting ODE system becomes

$$\frac{d\hat{u}_I}{dt} = \mathbf{A}\hat{u}_I + \mathbf{F}(\hat{u}_I, t). \tag{14}$$

Note that the expressions for $\mathbf{A}$ and $\mathbf{F}$ are different from the Dirichlet case. This scalar component formulation can be extended to a q-component coupled system by applying the RBF collocation discretization to each component equation and assembling the resulting semidiscrete equations into a single block matrix system. Specifically, by stacking the q-component coefficient vectors associated with all N collocation points, we form the system coefficient vector $\boldsymbol{\alpha}(t) \in \mathbb{R}^{qN}$ with $\alpha_j(t) \in \mathbb{R}^q, \quad j = 1, \ldots, N$. The scalar interpolation matrix $\mathbf{R}$ is extended to the block interpolation matrix $\mathbf{R}_q = \mathbf{R} \otimes \mathbf{I}_q \in \mathbb{R}^{qN \times qN}$ where $\mathbf{I}_q$ is the $q \times q$ identity matrix and $\otimes$ is the Kronecker product. If the spatial operator that acts on each component is the same, the corresponding differential operator matrix is extended similarly as $\mathbf{L}_{\mathbf{I}q} = \mathbf{L}_{\mathbf{I}} \otimes \mathbf{I}_q \in \mathbb{R}^{qN_I \times qN}$ or $\mathbf{Z}_q = \mathbf{Z} \otimes \mathbf{I}_q \in \mathbb{R}^{qN \times qN}$ depending on the boundary condition. Finally, we incorporate any coupling between the components through appropriate component-coupling operators or through the nonlinear term. Thus, the nonlinear source vector is assembled component-wise at the interior collocation points and has dimension $\mathbf{N}(\mathbf{u}_I) \in \mathbb{R}^{qN_I}$. This block formulation allows the components to be coupled without the need to modify the CSRBF spatial discretization.

4. ETD Time Discretization

What remains is to discretize (9) or (14) with respect to time. To do this, we consider a general semilinear ODE system of the form:

$$\frac{d\hat{u}_I}{dt} = \mathbf{A}\hat{u}_I + \mathbf{F}(\hat{u}_I, t) \tag{15}$$

Using Duhamel's principle, the exact solution of the IVP (15) is given by

$$\hat{u}_I(t) = e^{\mathbf{A}t}\hat{u}_I^0 + \int_0^t e^{\mathbf{A}*(t-s)}\,\mathbf{F}(s, \hat{u}_I(s))\,ds \quad \forall t \in [0, T] \tag{16}$$

Over the time interval $[t_n, t_{n+1}]$ with positive time step $k = t_{n+1} - t_n$, the variation-of-constants formula gives

$$\hat{u}_I(t_{n+1}) = e^{\mathbf{A}k}\hat{u}_I(t_n) + \int_{t_n}^{t_{n+1}} e^{\mathbf{A}*(t_{n+1}-s)}\,\mathbf{F}(s, \hat{u}_I(s))\,ds. \tag{17}$$

Letting $s = t_n + \tau k$ with $\tau \in [0, 1]$, $t_n = nk$, where $k \in [0, k_0]$ and $n \in [0, N]$, we get

$$\hat{u}_I(t_{n+1}) = e^{\mathbf{A}k}\hat{u}_I(t_n) + k\int_0^1 e^{\mathbf{A}*k(1-\tau)}\,\mathbf{F}(t_n + \tau k, \hat{u}_I(t_n + \tau k))\,d\tau \tag{18}$$

which serves as the foundation for deriving exponential time differencing (ETD) schemes. Integrating polynomial interpolants of the nonlinear function $\mathbf{F}$ yields the family of ETD Runge-Kutta schemes (ETDRK). In what follows, we describe the formulation of the second-order ETD-RDP scheme introduced in [43], beginning with a description of the general second-order ETDRK scheme.

4.1. ETD-RK2

To obtain the second-order ETD scheme, $\mathbf{F}$ is approximated by linear interpolation on each time interval $[t_n, t_{n+1}]$ as

$$\mathbf{F}(\hat{u}_I(t_n+s), t_n+s) \approx \mathbf{F}(t_n, \hat{u}_I^n) + \frac{s}{k}\Big(\mathbf{F}(t_{n+1}, \hat{u}_I^{n+1}) - \mathbf{F}(t_n, \hat{u}_I^n)\Big), \tag{19}$$

where $s = \tau k \in [0, k]$ and $\hat{u}_I^n = \hat{u}_I(t_n)$, $\hat{u}_I^{n+1} = \hat{u}_I(t_{n+1})$. Substituting into the integral and simplifying terms, the update formula in (18) becomes

$$\hat{u}_I^{n+1} = e^{\mathbf{A}k}\hat{u}_I^n + \mathbf{A}^{-1}\left(e^{\mathbf{A}k} - \mathbf{I}\right)\mathbf{F}(\hat{u}_I^n) + \frac{1}{k}\mathbf{A}^{-2}\left(e^{\mathbf{A}k} - \mathbf{I} - \mathbf{A}k\right)\left[\mathbf{F}(\hat{u}_I^{n+1}) - \mathbf{F}(\hat{u}_I^n)\right] \tag{20}$$

Here, the expressions involving inverse powers of $\mathbf{A}$ can equivalently be expressed using matrix φ-functions, without requiring $\mathbf{A}$ to be invertible. To avoid the implicit dependency on $\hat{u}_I^{n+1}$, we use a **predictor step** with a locally second-order approximation [41, 42], given by

$$\hat{u}_I^* = e^{\mathbf{A}k}\hat{u}_I^n + \mathbf{A}^{-1}(e^{\mathbf{A}k} - \mathbf{I})\mathbf{F}(\hat{u}_I^n), \tag{21}$$

and correct this estimate using a locally third-order **corrector step**

$$\hat{u}_I^{n+1} = e^{\mathbf{A}k}\hat{u}_I^n + \mathbf{A}^{-1}(e^{\mathbf{A}k} - \mathbf{I})\mathbf{F}(\hat{u}_I^n) + \frac{1}{k}\mathbf{A}^{-2}\left(e^{\mathbf{A}k} - \mathbf{I} - \mathbf{A}k\right)\left[\mathbf{F}(\hat{u}_I^*) - \mathbf{F}(\hat{u}_I^n)\right]. \tag{22}$$

The resulting globally second-order ETD-RK2 scheme is particularly attractive for stiff problems because it handles the linear term exactly. It also handles the nonlinear term explicitly, thus avoiding nonlinear iterations at each time step while providing second-order temporal accuracy under appropriate regularity assumptions. However, direct evaluation of the matrix-function expressions may suffer from numerical cancellation, particularly for small time steps. The scheme also involves the matrix exponential $e^{\mathbf{A}k}$ whose efficient evaluation can be computationally expensive. The ETD-RDP scheme addresses these computational difficulties by approximating the matrix exponential with a rational function having real and distinct poles and simplifying the resulting scheme.

4.2. ETD-RDP

The fully discrete ETD-RDP scheme is achieved by replacing the matrix exponentials in (22) with the RDP rational approximation

$$R_{\mathrm{RDP}}(\mathbf{A}k) = \left(\mathbf{I} + \frac{5}{12}\mathbf{A}k\right)\left[\left(\mathbf{I} - \frac{1}{3}\mathbf{A}k\right)\left(\mathbf{I} - \frac{1}{4}\mathbf{A}k\right)\right]^{-1}. \tag{23}$$

and using the Padé-(0,1) rational function

$$R_{01}(\mathbf{A}k) = \left(\mathbf{I} - \mathbf{A}k\right)^{-1} \tag{24}$$

in the predictor step (21). The resulting predictor-corrector scheme is of the form

$$\begin{aligned}\hat{u}_I^* &= R_{01}(\mathbf{A}k)\hat{u}_I^n + \mathbf{A}^{-1}\big[R_{01}(\mathbf{A}k) - \mathbf{I}\big]\mathbf{F}(\hat{u}_I^n),\\ \hat{u}_I^{n+1} &= R_{\mathrm{RDP}}(\mathbf{A}k)\hat{u}_I^n + \mathbf{A}^{-1}\big[R_{\mathrm{RDP}}(\mathbf{A}k) - \mathbf{I}\big]\mathbf{F}(\hat{u}_I^n)\\ &\quad + \frac{\mathbf{A}^{-2}}{k}\big[R_{\mathrm{RDP}}(\mathbf{A}k) - \mathbf{I} - \mathbf{A}k\big]\Big(\mathbf{F}(\hat{u}_I^*) - \mathbf{F}(\hat{u}_I^n)\Big).\end{aligned} \tag{25}$$

This can be simplified to the final efficient form

$$\begin{aligned}\hat{u}_I^* &= \big(\mathbf{I} - \mathbf{A}k\big)^{-1}\big[\hat{u}_I^n + k\mathbf{F}(\hat{u}_I^n)\big],\\ \hat{u}_I^{n+1} &= \big(\mathbf{I} - \tfrac{1}{3}\,\mathbf{A}\,k\big)^{-1}\Big[9\hat{u}_I^n + 2k\,\mathbf{F}(\hat{u}_I^n) + k\mathbf{F}(\hat{u}_I^*)\Big]\\ &\quad - \big(\mathbf{I} - \tfrac{1}{4}\,\mathbf{A}\,k\big)^{-1}\Big[8\hat{u}_I^n + \tfrac{3k}{2}\,\mathbf{F}(\hat{u}_I^n) + \tfrac{k}{2}\,\mathbf{F}(\hat{u}_I^*)\Big]\end{aligned} \tag{26}$$

This completes the spatial and temporal discretizations of the governing PDEs. In the next section, we investigate the accuracy and computational efficiency of the combined CSRBF-ETDRDP scheme for solving semilinear PDEs on regular and irregular domains. The algorithm is shown in **Fig.** 1.

Algorithm 1 CSRBF-ETDRDP Main Solver

Input: Points X, initial $\hat{u}_0$, boundary data, time T, N_t
Output: Solution vector $\hat{u}^n$ at each time level
Step 1 (Spatial Setup): Call Alg. 2 for $\mathbf{A}, \mathbf{F}$
Step 2 (Initialization): Set interior state $\hat{u}_I^0 = \hat{u}_0(X_I)$
Step 3 (Time Loop): Call Alg. 3 to advance $\hat{u}_I$
for $n = 1$ to N_t **do**
  if Dirichlet **then**
    Evaluate $\hat{u}_B^n = g(X_B, t_n)$
  else Neumann
    Solve $\hat{u}_B^n = \mathbf{K}_{BB}^{-1}\,(h_B^n - \mathbf{K}_{BI}\hat{u}_I^n)$ (Eq. (12))
  end if
end for

Step 1

Algorithm 2 Spatial Operator Construction

Input: Points X, indices I, B, shape ε, boundary type
Output: Operator $\mathbf{A}$, boundary coupling $\mathbf{F}$
Compute distance matrix $\mathbf{r}_{ij} = \|x_i - x_j\|$
Set $\delta = 1/\varepsilon$ and compute $v_{ij} = \max(1 - \delta\mathbf{r}_{ij}, 0)$
Build $\mathbf{R}_{ij}$, along with $\mathbf{L}_{I,ij}$ or $\mathbf{Z}_{ij}$
Form operator matrix ($\mathbf{L}$ or $\mathbf{K}$) (Eq. (8) or (11))
if Dirichlet **then**
  Extract $\mathbf{L}_{II}, \mathbf{L}_{IB}$
  Compute $\mathbf{A} = \mathbf{L}_{II}$ and coupling $\mathbf{F}(\hat{u}_I, t)$
else Neumann
  Extract $\mathbf{K}_{II}, \mathbf{K}_{IB}, \mathbf{K}_{BI}, \mathbf{K}_{BB}$
  Compute $\mathbf{A} = \mathbf{K}_{II} - \mathbf{K}_{IB}\mathbf{K}_{BB}^{-1}\mathbf{K}_{BI}$ and $\mathbf{F}(\hat{u}_I, t)$
end if

Step 3

Algorithm 3 ETD2-RDP Time-Stepping

Input: Initial $\hat{u}_I^0$, operator $\mathbf{A}$, forcing $\mathbf{F}$, step Δt, steps N_t
Output: Solution $\hat{u}_I^n$ at each time level
for $n = 1$ to $N_t - 1$ **do**
  Compute $\mathbf{F}^n = \mathbf{F}(\hat{u}_I^n, t_n)$
  Predictor: Compute $\hat{u}_I^*$ (Eq. (26))
  Compute $\mathbf{F}^* = \mathbf{F}(\hat{u}_I^*, t_{n+1})$
  Corrector: Compute $\hat{u}_I^{n+1}$ (Eq. (26))
end for

Fig. 1: Implementation of the CSRBF-ETDRDP Algorithm

5. Numerical Experiments

We consider both scalar reaction–diffusion equations and coupled reaction–diffusion systems and compare the CSRBF-based ETDRDP scheme with commonly used time discretization schemes such

as Crank–Nicolson (CN) and a second-order explicit Runge–Kutta (ERK2) scheme. In each test problem, we compare the accuracy and computational time of each scheme and discuss the effect of the support radius ϵ on performance. For problems with irregular domains, we use parametric curves to define the boundary. The entire implementation is carried out in MATLAB R2024b on a 14-inch MacBook Pro with an Apple M4 10-core chip and 16 GB RAM.

The accuracy and convergence of the schemes are evaluated using two discrete error norms, namely the maximum absolute error (E_∞) and the root mean square error (E_{RMS}), defined respectively as

$$E_\infty = \max_{1\le i\le N} |\hat{u}_i - u_i|, \qquad E_{\text{RMS}} = \left[\frac{1}{N}\sum_{i=1}^{N}(\hat{u}_i - u_i)^2\right]^{1/2}.$$

Here, N denotes the total number of collocation points, $\hat{u}_i$ represents the numerical solution, and u_i denotes the corresponding exact solution at the collocation point $\mathbf{x}_i$ and at the specified time t. We estimate the observed order of convergence using the formula

$$\text{EOC} = \frac{\log\left(\tilde{E}(k)/\tilde{E}(k/2)\right)}{\log(2)}$$

using both norms. When the spatial resolution and time step are refined simultaneously, the EOC measures the combined spatial–temporal convergence behavior.

Since our test problems are all second-order PDEs, we use the Wendland C^2 kernel, which is twice continuously differentiable and strictly positive definite in spatial dimensions up to three. The ETDRDP temporal scheme is known to be second-order accurate. In contrast, the spatial convergence order of CSRBF depends on the smoothness of the underlying Wendland function, the regularity of the analytical solution to the governing equation, and the support radius (ϵ). Hence, ϵ has to be carefully tuned to achieve the desired accuracy, since the observed order may become slightly higher or lower as ϵ is adjusted and the grid is refined.

5.1. Nonlinear Heat Problem

We consider a nonlinear heat problem formulated as a bistable reaction–diffusion equation describing a traveling wavefront on a unit square domain, given by

$$u_t = \Delta u - 4u^3 + 5u^2 - u. \tag{27}$$

over the time interval $t \in [0, 1]$. The initial and Dirichlet boundary conditions are chosen to be consistent with the analytical solution $u(x, y, t) = \frac{1}{1+e^{x+y-t}}$.

To begin, we partition the total collocation points N into $(pe-1)^2$ interior points and $4pe$ boundary points, where pe denotes the number of points per edge. As discussed in our Dirichlet boundary formulation, we perform the spatial discretization using the CSRBF interpolant to form the ODE system, which we then solve in time using each of our time discretization schemes. To examine how numerical error scales with the support radius ϵ in this problem and the remaining test problems in this work, we test a range of ϵ from 0.1 to 3.0 in increments of 0.1. The effect of the support radius on the sparsity of the interior collocation matrix is illustrated in **Fig.** 2. Observe that increasing ϵ reduces the sparsity of interior collocation matrix.

In **Fig.** 3A and **Fig.** 3B, we illustrate the exact solution and the error profile of the ETD-RDP scheme using a temporal step size of $\Delta t = 0.02$ and 900 collocation points. **Fig.** 3C displays the log-scale plot of the maximum absolute error against the support radius and the CPU time against the

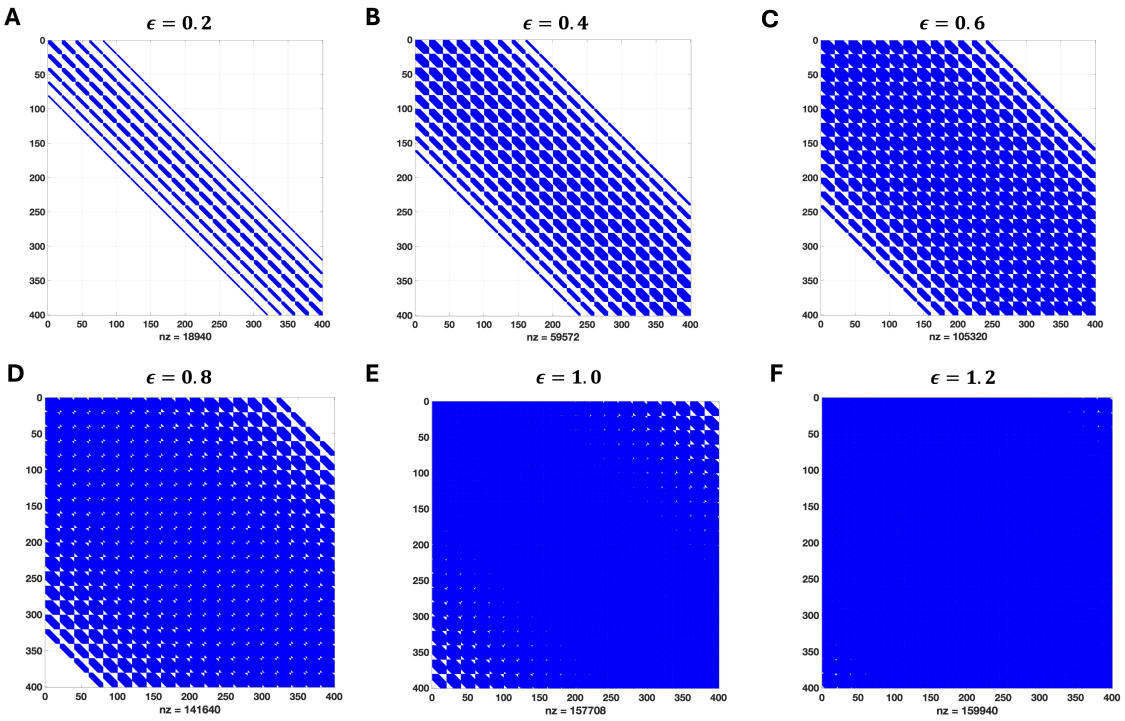


Fig. 2: **Sparsity patterns of the interior collocation matrix for different support radius (ϵ)** in Example 5.1 with ϵ increasing from 0.2 to 1.4 in increments of 0.2 and 400 interior points.

number of interior collocation points for all the three schemes. We observe that the error decreases monotonically as the support radius is increased, with the least error achieved at the largest support radius within the tested range.

Next, we compare the computational efficiency of ETDRDP with Crank–Nicolson (CN) and an explicit second-order Runge–Kutta scheme (ERK2). The convergence results are shown in Table 1 and illustrated in **Fig.** 3D. In the convergence test, we adjusted the range of time steps where necessary to obtain comparable errors across the schemes. This allowed us to use the CPU time as a measure of efficiency.

For a unit square domain, the maximum distance is its diameter ($\sqrt{2}$), so $\epsilon > \sqrt{2}$ will result in a dense matrix and hence make the basis function essentially global. For this reason, we set $\epsilon = 0.4$ for the convergence test, where the knee of the error versus support radius profile occurs in **Fig.** 3C. Other choices of ϵ yielded the same convergence behavior, with only differences in error magnitude.

Observe from Table 1 that all three schemes exhibit observed convergence orders approaching three on the finer discretizations. Since both the spatial resolution and the time step are refined simultaneously, these results reflect the combined spatial–temporal error rather than the temporal convergence order alone. A separate investigation of the spatial and temporal convergence contributions is left for future work. Overall, ETDRDP has comparable efficiency to CN and outperforms ERK2 across all tested spatial resolutions. The inefficiency of ERK2 for such a diffusion-dominated problem is expected, as evidenced by the relatively smaller time steps required to achieve comparable accuracy with CN and ETDRDP.

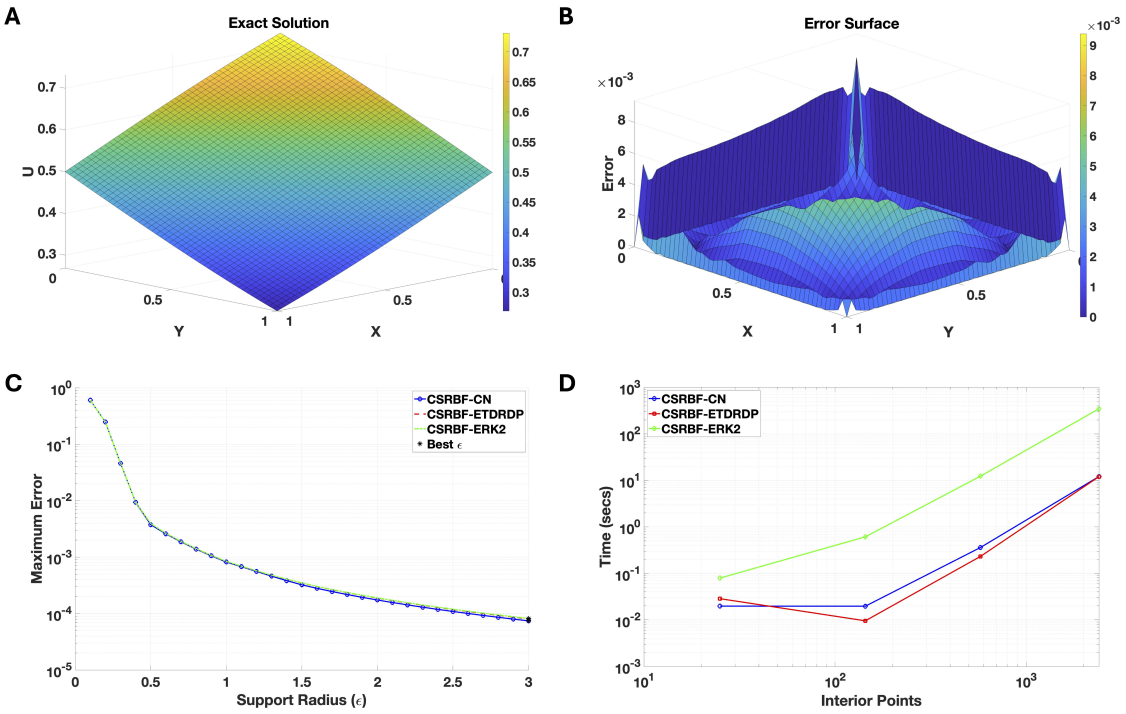


Fig. 3: **Numerical accuracy, support-radius sensitivity, and computational efficiency for Example 5.1** (A) Exact solution; (B) MAE; (C) Error versus support radius using $\Delta t = 10^{-5}$ for ERK2, $\Delta t = 0.02$ for ETDRDP and CN, and $N = 900$ collocation points; (D) CPU time versus the number of interior collocation points for the three schemes.

Method	N_{interior}	h	Δt	ϵ	E_∞		E_{RMS}		Time (s)
					Error	EOC	Error	EOC	
ETDRDP	25	0.1600	0.1000	0.40	4.01×10^{-1}	–	1.90×10^{-1}	–	0.0287
	144	0.0800	0.0500	0.40	1.02×10^{-1}	1.97	4.76×10^{-2}	2.00	0.0096
	576	0.0400	0.0250	0.40	1.47×10^{-2}	2.80	6.95×10^{-3}	2.78	0.2319
	2401	0.0200	0.0125	0.40	1.92×10^{-3}	2.94	7.99×10^{-4}	3.12	12.0351
ERK2	25	0.1600	0.0001	0.40	4.01×10^{-1}	–	1.90×10^{-1}	–	0.0795
	144	0.0800	0.00005	0.40	1.02×10^{-1}	1.97	4.75×10^{-2}	2.00	0.6125
	576	0.0400	0.000025	0.40	1.46×10^{-2}	2.81	6.91×10^{-3}	2.78	12.3290
	2401	0.0200	0.0000125	0.40	1.92×10^{-3}	2.93	7.92×10^{-4}	3.13	346.5374
CN	25	0.1600	0.1000	0.40	4.06×10^{-1}	–	1.90×10^{-1}	–	0.0198
	144	0.0800	0.0500	0.40	1.02×10^{-1}	1.99	4.75×10^{-2}	2.00	0.0198
	576	0.0400	0.0250	0.40	1.46×10^{-2}	2.81	6.90×10^{-3}	2.78	0.3609
	2401	0.0200	0.0125	0.40	1.87×10^{-3}	2.97	7.90×10^{-4}	3.13	12.1053

Table 1: Errors, observed convergence orders (EOC), and CPU times for the CSRBF-based ETDRDP, ERK2, and CN schemes applied to the nonlinear heat problem in Example 5.1 at final time $T = 1$.

5.2. Diffusion–Reaction Problem

Next, we consider the inhomogeneous semilinear diffusion–reaction problem [51] defined on the spatial domain $\Omega = [-0.5, 0.5]^2$ over the time interval $t \in [0, 1]$, given by

$$\frac{\partial u}{\partial t} = \Delta u + \frac{u}{u^2 + 1} + f(x, y, t).$$

It admits an exact solution $u(x, y, t) = e^{-t} \sin(\pi x) \sin(\pi y)$, which satisfies the homogeneous Neumann boundary condition $\frac{\partial u}{\partial n} = 0$ for $(x, y) \in \partial\Omega$. The source term $f(x, y, t)$ is given by $f(x, y, t) = \left(2\pi^2 - 1 - \frac{1}{1+e^{-2t}\sin^2(\pi x)\sin^2(\pi y)}\right) e^{-t} \sin(\pi x) \sin(\pi y)$.

For this problem, we varied the support radius ϵ from 0.4 to 3.0 in increments of 0.1, since values below this range yielded extremely large errors. Similar to the first test problem, we present in **Fig.** 4A, B the analytical solution and the error of the ETDRDP scheme. With parameters $\Delta t = 0.02$ and $h \approx 0.03$, an error estimate of the form $\mathcal{O}(h^2 + (\Delta t)^2)$ corresponds to a scale of approximately $c \cdot 1.3 \times 10^{-3}$, where $c > 0$ is an unspecified constant. This is consistent with the scale of the error plot. The MAE against support radius for both ETDRDP and CN schemes is presented in **Fig.** 4C. Observe that the minimum error for ETDRDP and CN occurs at a support radius of $\epsilon = 1.4$. ERK2 is not included in the plots or the subsequent analysis of this test problem because it failed to match the error levels of competing schemes and exhibited an observed convergence order close to one under the tested discretizations. For the range of h that we considered, we could not find time steps Δt for ERK2 that produced the expected second-order convergence, and the restrictive time-step requirements also increased the computational time, as shown in Table S1 of the supplementary material.

Observe from Table 2 that both ETDRDP and CN exhibit approximately second-order observed convergence when the optimal support radius of 1.4 is used. This optimal support radius did not change as the grid was refined. We note that the expected convergence behavior shown in Table 2 was only possible for a small subset of temporal and spatial step sizes. Most choices of h and k did not exhibit the expected convergence behavior. To achieve convergence, we had to shift away from the optimal support radius (see results in Table S3 of the supplementary material). This behavior is consistent with CSRBF based schemes that may attain minimum errors at intermediate support radii, as shown in **Fig.** 4C and discussed in [28]. However, it remains unclear why a mid-range optimal ϵ is attained for this problem whereas the largest tested ϵ was attained in Example 5.1. This will be examined in future studies.

To compare the performance of ETDRDP and CN, we selected different time steps to attain comparable error for both schemes. This allowed us to use the CPU time as a measure of computational efficiency, as was done in Example 5.1. Observe from the CPU times in Table 2 and **Fig.** 4D that ETDRDP is about 1.6 times faster than CN at the finest spatial grid. This lower computational cost can be attributed to the relatively large time steps that could be taken with ETD-RDP to achieve comparable accuracy to CN (see Table 2). Hence, ETD-RDP provides a better balance between computational cost and numerical accuracy for this test problem.

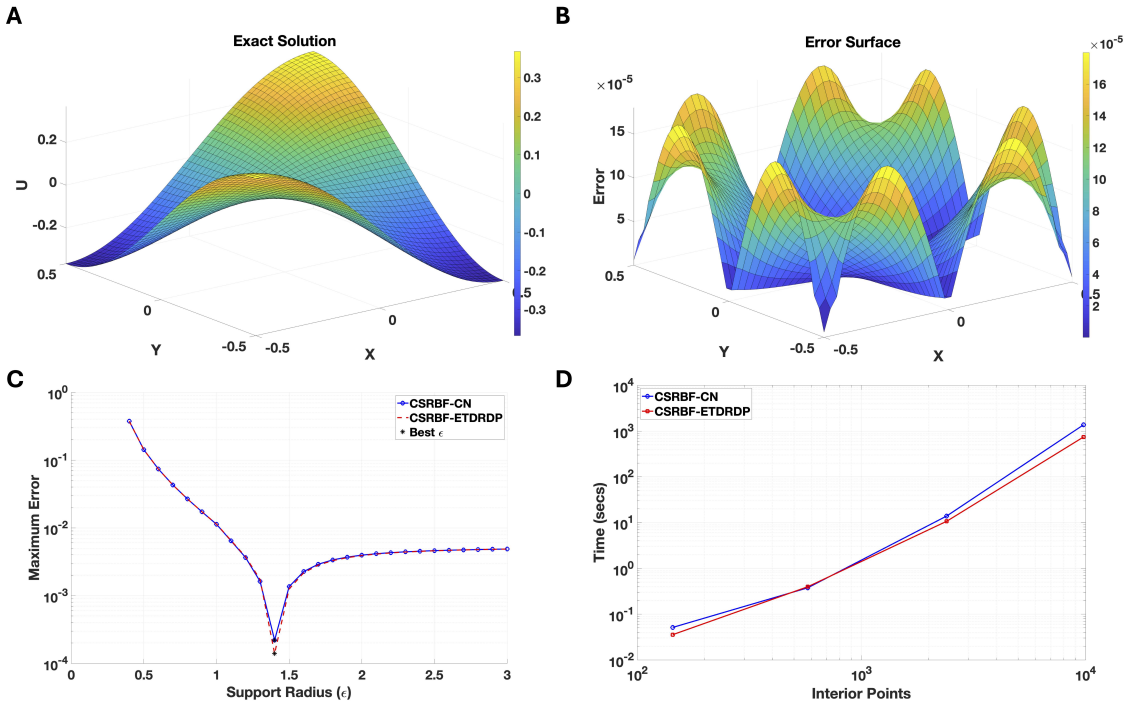


Fig. 4: **Numerical accuracy, support sensitivity and computational efficiency plots for** Example 5.2.(A) Exact solution, (B) MAE, (C) Error versus support radius using $\Delta t = 0.02$ for both ETDRDP and CN, with $N = 900$ collocation points, and (D) CPU time versus number of interior collocation points for both schemes.

Method	N_{interior}	h	Δt	Best ϵ	E_∞		E_{RMS}		Time (s)
					Error	EOC	Error	EOC	
ETDRDP	144	0.0800	0.2000	1.40	2.06×10^{-3}	–	7.64×10^{-4}	–	0.0698
	576	0.0400	0.1000	1.40	6.75×10^{-4}	1.61	2.33×10^{-4}	1.71	0.4116
	2401	0.0200	0.0500	1.40	1.79×10^{-4}	1.92	5.34×10^{-5}	2.12	10.7236
	9801	0.0100	0.0250	1.40	4.30×10^{-5}	2.05	9.02×10^{-6}	2.57	802.7427
CN	144	0.0800	0.0900	1.40	1.42×10^{-3}	–	8.68×10^{-4}	–	0.0415
	576	0.0400	0.0450	1.40	7.99×10^{-4}	0.83	4.50×10^{-4}	0.95	0.6953
	2401	0.0200	0.0225	1.40	3.50×10^{-4}	1.19	1.92×10^{-4}	1.23	14.1371
	9801	0.0100	0.01125	1.40	8.37×10^{-5}	2.06	4.37×10^{-5}	2.14	1315.0836

Table 2: Errors, observed convergence orders (EOC), and CPU times for the CSRBF-based ETDRDP and CN schemes applied to the diffusion–reaction problem in Example 5.2 at final time $T = 1$.

5.3. Burgers' Equation

Next, we evaluate the performance of the CSRBF-ETDRDP scheme against CSRBF-CN and CSRBF-ERK2 for a system of nonlinear convection–diffusion equations posed in regular and irregular domains. The test problem considered is the two-dimensional coupled viscous Burgers' equation [52]. This parabolic convection-diffusion system models the interaction between nonlinear advection and viscous diffusion in fluid dynamics. The two-dimensional viscous system is given by

$$\begin{cases} \dfrac{\partial u}{\partial t} + u\,\dfrac{\partial u}{\partial x} + v\,\dfrac{\partial u}{\partial y} = \kappa\,\Delta u, \\ \dfrac{\partial v}{\partial t} + u\,\dfrac{\partial v}{\partial x} + v\,\dfrac{\partial v}{\partial y} = \kappa\,\Delta v, \end{cases}$$

where $u(x, y, t)$, $v(x, y, t)$ are the velocity components, $\kappa = 1/Re > 0$ is the viscosity coefficient and Re is the Reynolds number. Fletcher [53] presented an exact travelling-wave-type solution of this system obtained via the Cole–Hopf transformation, which is given by

$$\begin{cases} u(x, y, t) = \dfrac{3}{4} - \dfrac{1}{4}\left[1 + \exp((4y - 4x - t)/32\,\kappa)\right]^{-1}, \\ v(x, y, t) = \dfrac{3}{4} + \dfrac{1}{4}\left[1 + \exp((4y - 4x - t)/32\,\kappa)\right]^{-1}. \end{cases}$$

The coupled system can also be expressed in terms of the velocity vector $\mathbf{w}(x, y, t) = \big(u(x, y, t),\ v(x, y, t)\big)$ as

$$\frac{\partial \mathbf{w}}{\partial t} + \mathbf{w} \cdot \nabla \mathbf{w} = \kappa\,\Delta \mathbf{w}, \qquad \textit{where} \quad \mathbf{w} \cdot \nabla \mathbf{w} = \begin{pmatrix} u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} \\ u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} \end{pmatrix}.$$

5.3.1. Regular domain

We solve the problem on the unit square $\Omega = [0, 1]^2$ over the time interval of $[0, 1]$. The initial and Dirichlet boundary conditions are chosen to be consistent with the exact solution. The exact solution and the numerical error obtained using CSRBF-ETDRDP at two different Reynolds numbers are presented in **Fig.** 5 and **Fig.** 6 respectively. As depicted in the plots, at $Re = 100$, the wave profiles are smooth with numerical errors on the order of 10^{-3}. As the Reynolds number increases to $Re = 1000$, the solution develops a steeper, shock-like transition, and the numerical error increases to the order of 10^{-2}. The smallest error occurs at the largest tested support radius as shown in **Fig.** 7A.

We present in Table 3 the convergence behavior of the schemes using a fixed support radius of 0.4, chosen for the same reason as in Example 5.1. Again, we adjust the time steps to obtain comparable errors, allowing computational efficiency to be assessed through CPU time. Observe that all the methods exhibit approximately second-order observed convergence in both error norms as the spatial and temporal resolution are refined. At the finest resolution, ETDRDP is faster than CN despite using a smaller time step. However, the ERK2 scheme is the most efficient for this test problem, at the specified Reynolds number.

We proceed to investigate the influence of the Reynolds number, support radius, and number of collocation points on the efficiency of the schemes. To do this, we compare the MAE against CPU time across variations in Reynolds number (Re), total collocation points (N), and support radius (ϵ). For a fixed support radius of $\epsilon = 0.4$, we obtain comparable errors at $N \in \{100, 400, 900, 1600\}$, by choosing appropriate time steps. This procedure is carried out for Reynolds numbers, Re $\in \{50, 100, 200, 400\}$. We observe from the results in **Fig.** 8 that ERK2 is consistently more efficient than ETDRDP across all tested Reynolds numbers (Re), whereas CN is the least efficient.

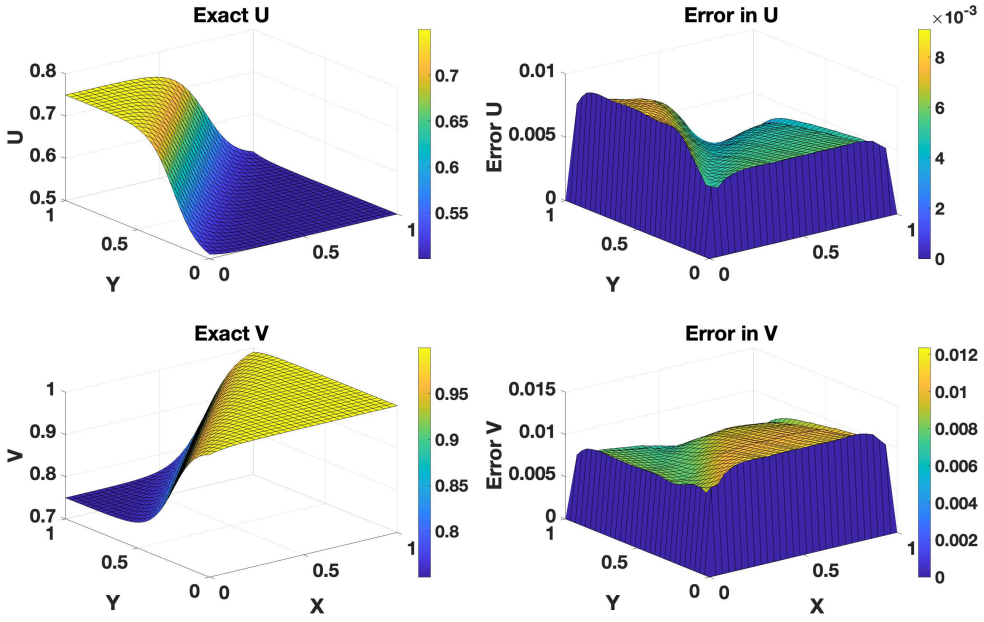


Fig. 5: **Numerical accuracy plots using ETDRDP for** Example 5.3 for $Re = 100$, $\Delta t = 0.01$ and $N = 900$. Solutions U and V and their corresponding errors are displayed.

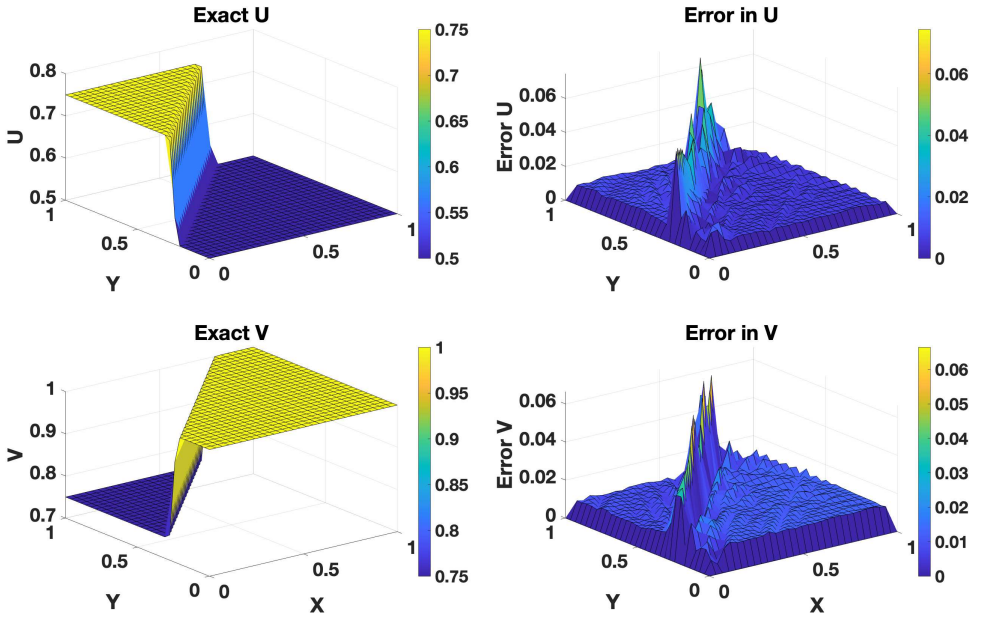


Fig. 6: **Numerical accuracy plots using ETDRDP for** Example 5.3 for $Re = 1000$, $\Delta t = 0.01$ and $N = 900$. Solutions U and V and their corresponding errors are displayed.

The difference in performance can be partly explained by the relative importance of diffusion and convection at different Reynolds numbers. Although the support radius is fixed, the number of neighboring collocation points within the support generally increases as the spatial grid is refined. At low Reynolds numbers, the solution is smooth, the linear diffusion and its associated stiffness is stronger while the influence of the convective term is less. In this case, the larger stable time steps permitted by ETDRDP can compensate for the computational cost of the linear solves, yielding performance comparable to ERK2. However, as Re increases, convection becomes more important relative to diffusion, and the stability advantage of treating the diffusion term implicitly may diminish. Consequently, the cost of the linear solves required by ETDRDP may outweigh the benefit of its treatment of the diffusive term. The explicit method, which avoids these linear solves, is thus more efficient for the tested parameter ranges.

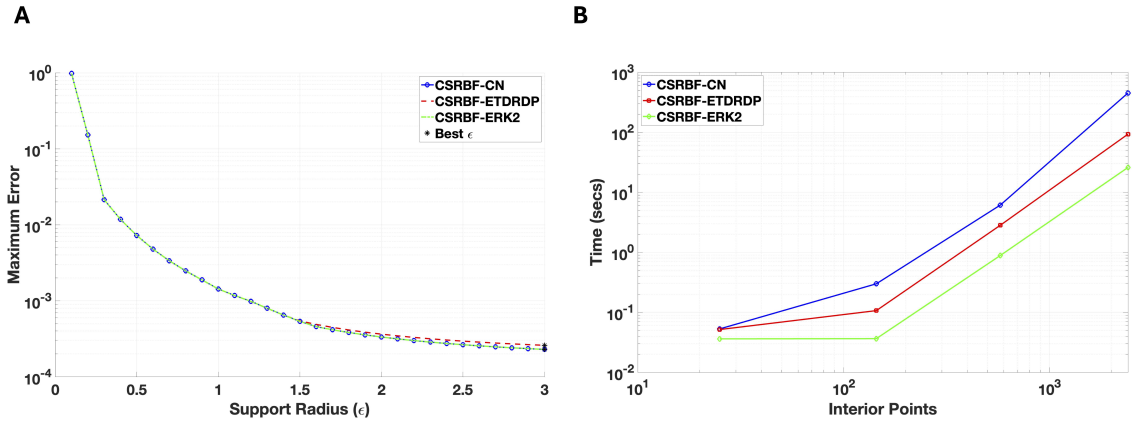


Fig. 7: Sensitivity and computational scaling plots for Example 5.3 for $Re = 50$, $N = 900$, $\Delta t = (0.01, 0.02, 0.004)$ for ETDRDP, CN and ERK2 respectively.

Method	N_{interior}	h	Δt	ϵ	E_∞		E_{RMS}		Time (s)
					Error	EOC	Error	EOC	
ETDRDP	25	0.1600	0.0040	0.40	4.76×10^{-1}	–	2.30×10^{-1}	–	0.0516
	144	0.0800	0.0020	0.40	6.37×10^{-2}	2.90	3.54×10^{-2}	2.70	0.1067
	576	0.0400	0.0010	0.40	1.64×10^{-2}	1.96	1.03×10^{-2}	1.79	2.8198
	2401	0.0200	0.0005	0.40	3.33×10^{-3}	2.30	2.20×10^{-3}	2.22	92.9228
ERK2	25	0.1600	0.0100	0.40	4.76×10^{-1}	–	2.30×10^{-1}	–	0.0361
	144	0.0800	0.0050	0.40	6.37×10^{-2}	2.90	3.54×10^{-2}	2.70	0.0364
	576	0.0400	0.0025	0.40	1.64×10^{-2}	1.96	1.03×10^{-2}	1.79	0.8837
	2401	0.0200	0.0013	0.40	3.33×10^{-3}	2.30	2.20×10^{-3}	2.22	25.8856
CN	25	0.1600	0.1000	0.40	4.78×10^{-1}	–	2.30×10^{-1}	–	0.0529
	144	0.0800	0.0500	0.40	6.37×10^{-2}	2.91	3.55×10^{-2}	2.70	0.2984
	576	0.0400	0.0250	0.40	1.64×10^{-2}	1.96	1.03×10^{-2}	1.79	6.1178
	2401	0.0200	0.0125	0.40	3.33×10^{-3}	2.30	2.20×10^{-3}	2.22	451.0212

Table 3: Convergence results for the CSRBF-based ETDRDP, ERK2, and CN schemes applied to Example 5.3 at $Re = 50$ and $T = 1$.

Next we investigate the comparative efficiency of the schemes as the support radius varies, $\epsilon \in \{0.175, 0.35, 0.7, 1.4\}$, at a fixed Reynolds number with spatial resolution set to $N = 900$. Errors are matched across schemes by adjusting the time step. The analysis is repeated for Reynolds

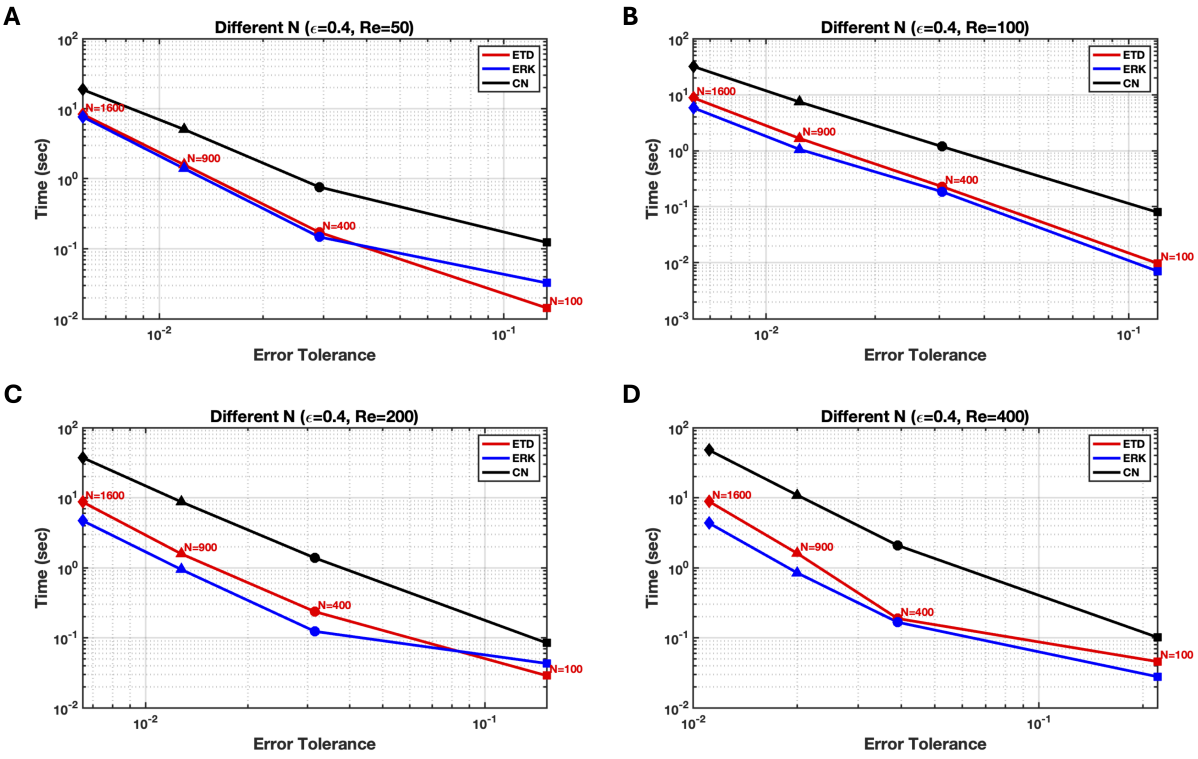


Fig. 8: **Efficiency plots for** Example 5.3 using a fixed support radius ($\epsilon = 0.4$): (A) varying the number of collocation points ($N = 100, 400, 900, 1600$); (B) varying the Reynolds number ($Re = 50, 100, 200, 400$).

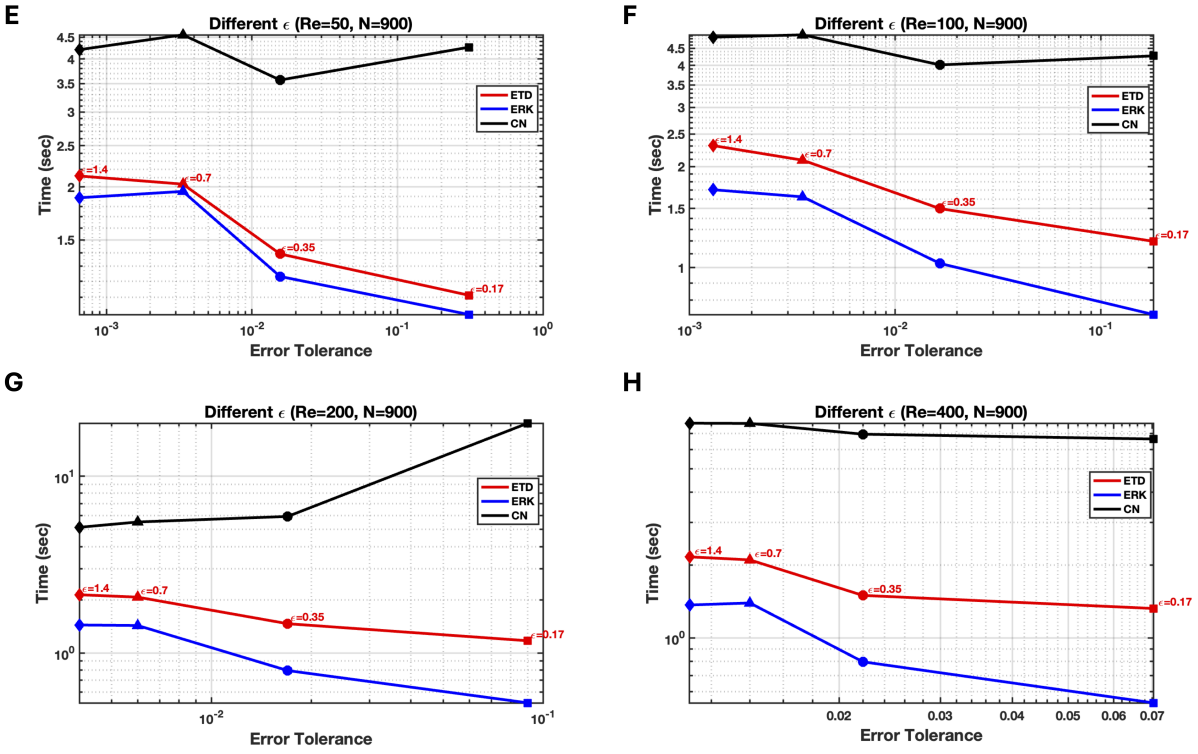


Fig. 9: **Efficiency plots for** Example 5.3 using a fixed number of collocation points ($N = 900$): (A) varying the support radius ($\epsilon = 0.175, 0.35, 0.7, 1.4$); (B) varying the Reynolds number ($Re = 50, 100, 200, 400$).

numbers Re $\in \{50, 100, 200, 400\}$. The results are shown in **Fig.** 9 respectively. At the lower Reynolds number (Re = 50), both ERK2 and ETDRDP have comparable CUP time but the gap widens as Re increases to 200 and 400, with ERK2 being more efficient.

Since N is fixed and the support radius grows, the CSRBF operator matrix becomes denser, requiring more computational work to solve the associate linear systems with ETDRDP. As in the fixed-ϵ case, we observe comparable CPU times at low Reynolds numbers, where diffusion is relatively stronger than convection. As ϵ and Re increases, the problem is dominated by convection and the influence of the diffusion term diminishes, reducing the need for a stiff solver. The larger computational cost per time step for ETDRDP compared with ERK2 leads to a more efficient solve with the explicit scheme.

Throughout this parameter sweep, CN scheme consistently remains less efficient compared to the other schemes. The numerical values corresponding to the MAE, the CPU time, and the time steps used for these simulations are presented in Tables S5–S8 of the supplementary material.

5.3.2. Irregular domain

Finally, we demonstrate that the CSRBF-ETDRDP scheme can solve Burger's equation on irregular domains. Specifically, we consider an amoeba-shaped domain and a peanut-shaped domain, whose boundaries are parametrically defined as

$$\partial\Omega = \{(x, y) \mid x = r(\theta)\cos\theta,\ y = r(\theta)\sin\theta,\ 0 \le \theta < 2\pi\}$$

Both domains are illustrated in **Fig.** 10A and 10B, where the blue points represent the interior nodes and the red points denote the boundary nodes. Choosing the radial function $r(\theta)$ to be

$$r(\theta) = 0.4e^{\sin\theta}\sin^2(2\theta) + 0.4e^{\cos\theta}\cos^2(2\theta),$$

we obtain an amoeba-shaped domain which we use in this test problem. For the choice of

$$r(\theta) = \sqrt{\cos(2\theta) + \sqrt{1.1 - \sin^2(2\theta)}}$$

we obtain a peanut-shaped domain, which is used for the next test problem (Example 5.4). In **Fig.** 11, we present the exact solution and the corresponding numerical errors. Observe that the accuracy is comparable to that obtained on the regular domain case.

5.4. Schrödinger Equation

Finally, we investigate the performance of our schemes on a partial differential equation that describes the evolution of a quantum-mechanical wave function. Unlike particles that follow Newton's second law, electrons follow the rules of waves evolving in space and time [54]. In this section, we consider its nonlinear extension, known as the *nonlinear Schrödinger (NLS) equation*. Specifically, we focus on the two-dimensional form

$$i\Psi_t + \Delta\Psi = q(|\Psi|^2)\Psi,$$

where $q(|\Psi|^2)$ represents a nonlinear potential depending on the wave intensity and is given as $q(|\Psi|^2) = B(x, y) + C(x, y)|\Psi|^2$ with

$$B(x, y) = (1 - 2\pi^2)\big(1 - \cos^2(\pi x)\cos^2(\pi y)\big), \quad C(x, y) = 1 - 2\pi^2.$$

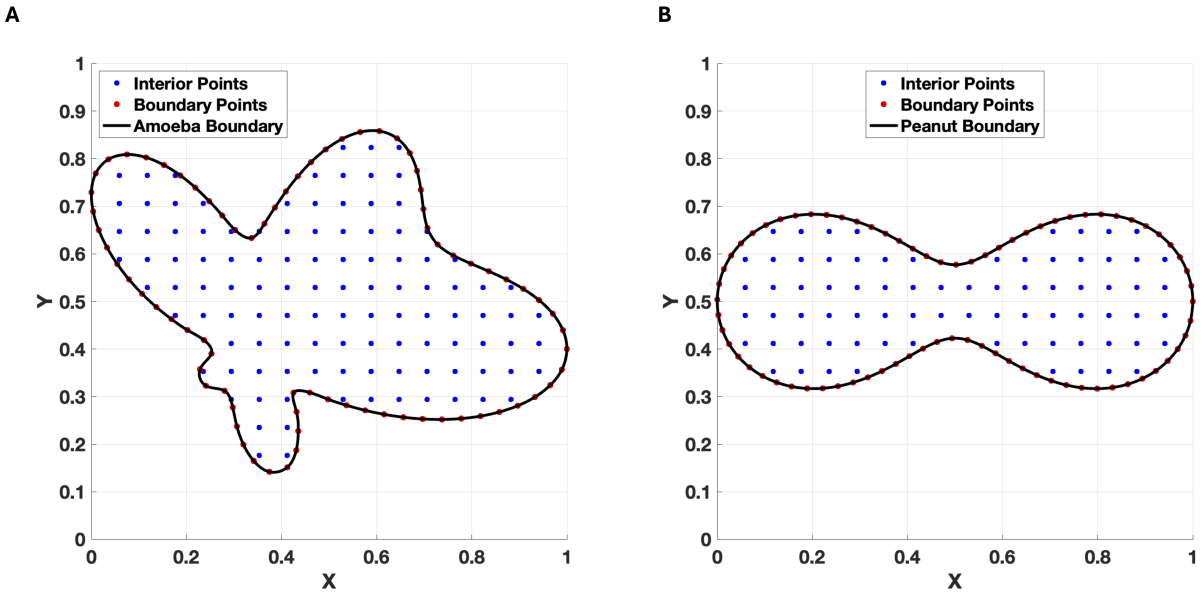


Fig. 10: Amoeba- and peanut-shaped domains with interior and boundary collocation points.

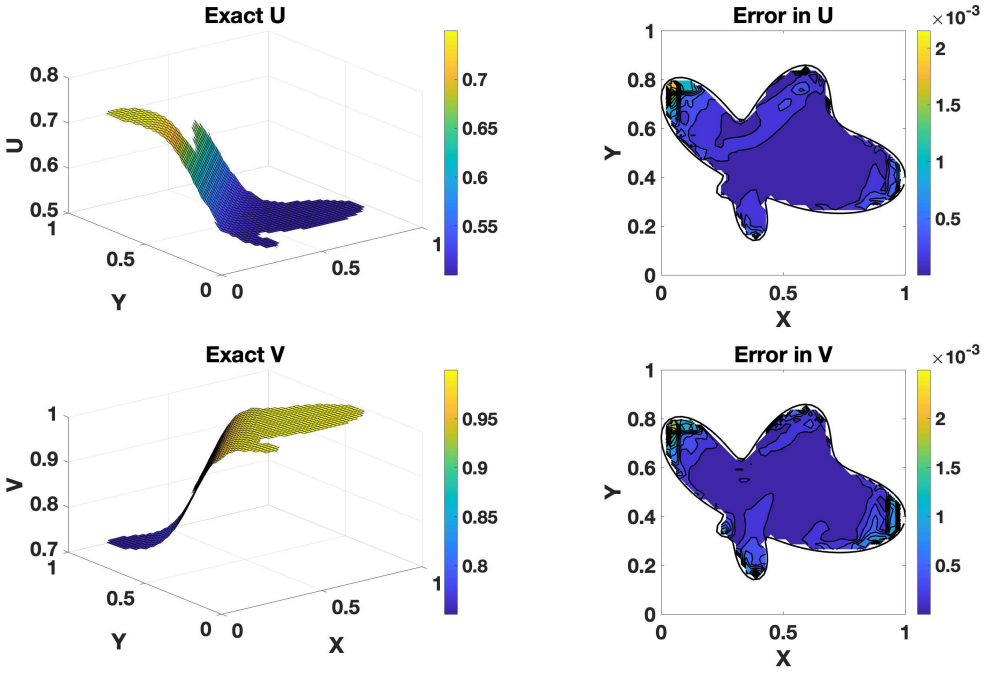


Fig. 11: **Numerical accuracy plots using ETDRDP for** Example 5.3 for $Re = 100$, $\epsilon = 1$, $\Delta t = 0.01$ and $N = 900$. The velocity components U and V and their corresponding errors are displayed on the amoeba-shaped domain.

Writing the complex-valued function as $\Psi(x,y,t) = v(x,y,t) + i\, w(x,y,t)$, we obtain the coupled real system

$$\begin{cases} v_t + \Delta w = q(v^2+w^2)w, \\ w_t - \Delta v = -\,q(v^2+w^2)v. \end{cases}$$

This system can equivalently be expressed in a compact vector–matrix form as

$$\begin{bmatrix} v_t \\ w_t \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \Delta v \\ \Delta w \end{bmatrix} = q(v^2+w^2) \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}.$$

Setting

$$\mathbf{u} = \begin{bmatrix} v \\ w \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix},$$

and noting that $\mathbf{D}$ is not a diagonal matrix, the system becomes

$$\mathbf{u}_t + \mathbf{D}\Delta\mathbf{u} = q(|\mathbf{u}|^2)\,\mathbf{D}\mathbf{u},$$

which fits into the general framework introduced in (1). The system admits the exact solution $\Psi(x,y,t) = e^{-it}\cos(\pi x)\cos(\pi y)$ [55] with initial and boundary conditions chosen to be consistent with this solution.

The total *mass* $M(t)$ and *energy* $E(t)$ of the wave function are defined, respectively, by

$$M(t) = \iint_\Omega |\Psi(x,y,t)|^2\, dx\, dy, \quad E(t) = \iint_\Omega \left[|\nabla\Psi(x,y,t)|^2 - \tfrac{1}{2}|\Psi(x,y,t)|^4\right] dx\, dy.$$

These quantities are conserved in the continuous setting. To numerically verify the conservation properties, the integrals can be approximated as in [56] using the trapezoidal rule as

$$M(t_n) \approx h^2 \sum_{i\in\text{interior}} |\Psi(x_i,y_i,t_n)|^2 + \frac{h^2}{2} \sum_{i\in\text{boundary}} |\Psi(x_i,y_i,t_n)|^2,$$

and for the energy at time t_n,

$$\begin{aligned} E(t_n) \approx\ & h^2 \sum_{i\in\text{interior}} \left[|\Psi_x(x_i,y_i,t_n)|^2 + |\Psi_y(x_i,y_i,t_n)|^2 - \tfrac{1}{2}|\Psi(x_i,y_i,t_n)|^4\right] \\ & + \frac{h^2}{2} \sum_{i\in\text{boundary}} \left[|\Psi_x(x_i,y_i,t_n)|^2 + |\Psi_y(x_i,y_i,t_n)|^2 - \tfrac{1}{2}|\Psi(x_i,y_i,t_n)|^4\right]. \end{aligned}$$

5.4.1. Regular domain

We first solve this problem on the regular unit square domain with initial and boundary conditions chosen from the exact solution. We present in **Fig.** 12 the exact solution of the wave function Ψ, expressed in terms of its real and imaginary components v and w as well as the error of the problem in a unit square domain using ETDRDP scheme. To assess the accuracy and conservation properties of both schemes, we compare the evolution of mass and energy over time with their exact analytical values. In **Fig.** 13A and 13B, we display the evolution of these two conserved quantities. The exact mass and energy are respectively $M = 0.25$ and $E = (8\pi+3)(8\pi-3)/128$. Both quantities are preserved numerically by all schemes. Specifically, at the final time $T = 1$, the ETDRDP scheme yields mass and energy error of about 8.02×10^{-4} and 6.43×10^{-4}, respectively, while CN method yields 8.12×10^{-4} and 4.89×10^{-4}.

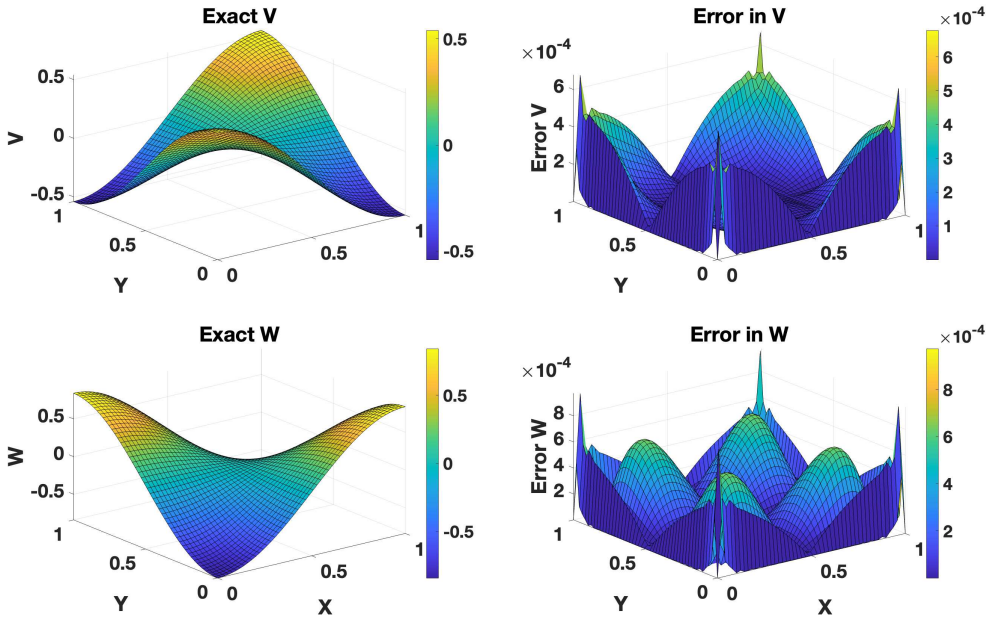


Fig. 12: **Numerical accuracy plots for** Example 5.4 for $\Delta t = 0.01$ and $N = 900$. Solutions V and W and their corresponding errors are displayed.

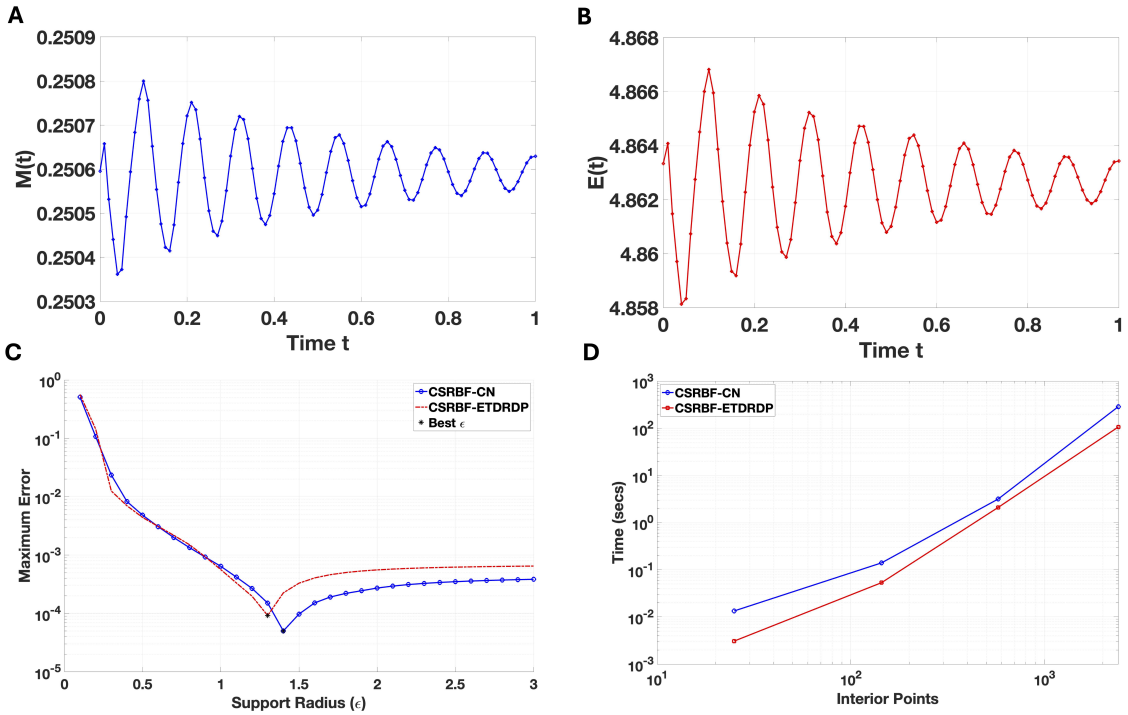


Fig. 13: **Conservation, support sensitivity and efficiency plots for** Example 5.4.(A) Mass conservation, (B) Energy Conservation, (C) Error across different support radius using $\Delta t = 0.03$ for both ETDRDP and CN and $N = 900$, and (D) comparative profiles for CPU time versus interior nodes.

Both ETDRDP and CN achieved a mid-range optimal support radius that alternated between 1.3 and 1.4 depending on the size of N. See **Fig.** 13C for the error versus support radius profile. As was previously observed in Example 5.2, ERK2 failed to achieve second-order accuracy as shown in Table S2 of the supplementary material. In Table 4, we report the convergence results for ETDRDP and CN schemes, using the optimal support radius for each nodal size N. As was done in previous examples, we chose the range of time steps to obtain comparable error across schemes. Observe from the table that both schemes converge with approximately second-order accuracy; however, ETDRDP is more efficient across all domain node sizes (**Fig.** 13C) and about 2.7 times faster than CN for a comparable level of accuracy at the finest spatial grid. It should also be noted here that fixing ϵ to either 1.3 or 1.4 during grid refinement also yields convergent schemes as shown Table S4 of the supplementary, however the accuracy is lower than adjusting the support radius as we vary the space and time steps. As we noted in Example 5.2, convergence for both schemes is sensitive to the range of grid refinements used. This is suspected to be a consequence of having a mid-range optimal support radius, as was noted in [28].

5.4.2. Irregular domain

To demonstrate the spatial flexibility of the CSRBF-ETDRDP scheme, we solved the Schrödinger equation on the irregular peanut-shaped domain illustrated in **Fig.** 10B. The exact solution and error plot for this problem are presented in **Fig.** 14. The observed accuracy is comparable to that obtained on the regular domain.

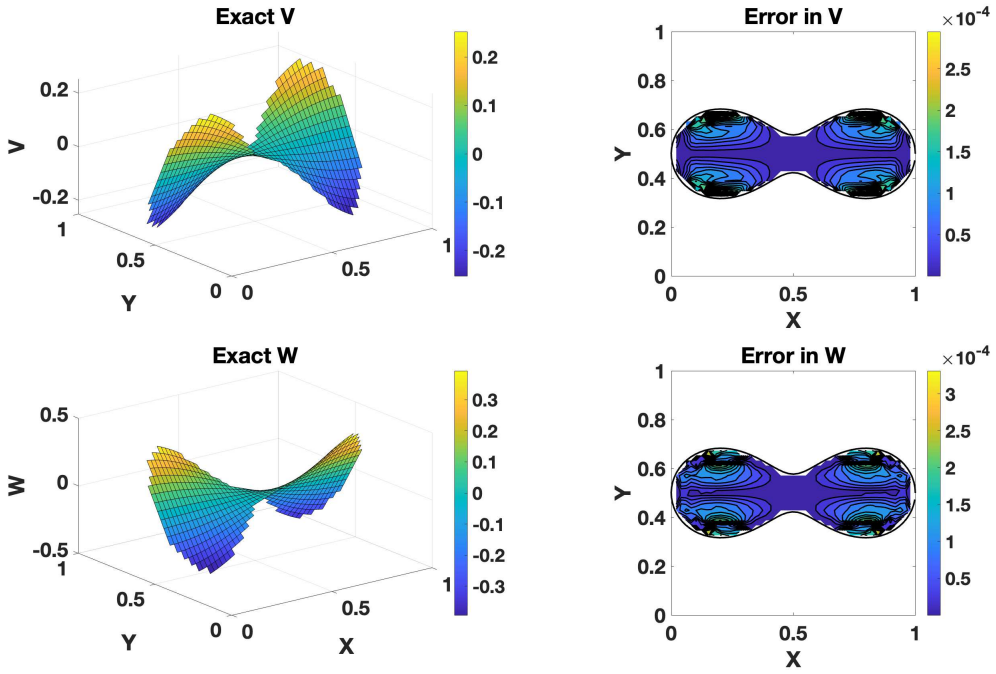


Fig. 14: **Numerical accuracy plots for** Example 5.4 with $\epsilon = 1$, $\Delta t = 0.01$ and $N = 900$. Solutions V and W and their corresponding errors are displayed in the Peanut shape.

From the plots of support radius against the MAE in our various test problems, we observe both a monotonic curve where the error continues to decrease with increasing ϵ and a case where a mid-range optimal support radius is achieved. For the monotonic case, we observe convergence of the numerical schemes for the tested values of ϵ over a reasonable range of spatial and temporal

Method	N_{interior}	h	Δt	E_∞			E_{RMS}			Time (s)
				Best ϵ	Error	EOC	Best ϵ	Error	EOC	
ETDRDP	25	0.1600	0.0080	1.30	1.07×10^{-3}	–	1.30	5.99×10^{-4}	–	0.0030
	144	0.0800	0.0040	1.40	3.94×10^{-4}	1.44	1.30	1.36×10^{-4}	2.14	0.0533
	576	0.0400	0.0020	1.30	1.33×10^{-4}	1.57	1.30	3.81×10^{-5}	1.84	2.1036
	2401	0.0200	0.0010	1.30	2.96×10^{-5}	2.17	1.20	1.11×10^{-5}	1.78	106.7156
CN	25	0.1600	0.0500	1.30	6.53×10^{-4}	–	1.30	3.64×10^{-4}	–	0.0134
	144	0.0800	0.0250	1.40	3.10×10^{-4}	1.07	1.40	1.59×10^{-4}	1.19	0.1403
	576	0.0400	0.0125	1.40	9.16×10^{-5}	1.76	1.30	4.56×10^{-5}	1.80	3.1709
	2401	0.0200	0.0063	1.40	1.93×10^{-5}	2.25	1.40	1.03×10^{-5}	2.14	288.0282

Table 4: Convergence results for the CSRBF-based ETDRDP and CN schemes for Example 5.4 at final time $T = 1$.

grid refinements. On the other hand, test problems with a mid-range optimal support radius only converged at the optimal support radius under a very restrictive choice of spatial and temporal step sizes or when ϵ is chosen away from the optimal value, which compromises accuracy. This sensitivity of convergence to the nature of the optimal support radius is not unique to CSRBF, but has also been observed in some global RBFs. The details of this uncertainty or trade-off principle can be found in [28] where the concepts of stationary and non-stationary interpolation approaches and safe and unsafe ϵ are discussed.

6. Conclusions

In this article, we have developed and investigated the performance of a novel CSRBF-ETDRDP scheme that is able to solve stiff problems on regular and irregular domains. The scheme uses the Wendland C^2 function for spatial discretization in all test problems and an exponential time differencing method with real and distinct poles (ETDRDP) that is L-stable for temporal discretization. The scheme eliminates the need for structured grids and can be implemented using direct linear solvers and performs competitively with schemes combining CSRBF with CN and ERK2. CSRBF-ETDRDP achieved approximately second-order observed convergence in both the maximum absolute error (MAE) and root mean square error (RMSE) norms for most test problems, with higher observed convergence orders in some cases. Test problems were chosen to have Dirichlet or Neumann boundary conditions and to include both scalar equations and coupled systems.

Across all test problems, the CSRBF-ETDRDP scheme was more efficient than the CSRBF-CN scheme, requiring less computational time than CN while maintaining comparable accuracy. In the bistable diffusion-reaction problem where the main source of stiffness is the Laplacian, ETDRDP was more efficient than ERK2. However, ERK2 was more efficient than ETDRDP for Burgers' equation, where nonlinear convection becomes increasingly important relative to diffusion at higher Reynolds numbers. These results suggest that CSRBF-ETDRDP is particularly effective for the tested stiff, diffusion-dominated problems, while CSRBF-ERK2 can be more efficient for convection-dominated problems. The performance advantage of CSRBF-ETDRDP was particularly evident in the inhomogeneous semilinear diffusion–reaction problem and Schrödinger equation where mid-range optimal support radii ϵ were observed. Here ERK2 failed to exhibit the expected convergence behavior and required significantly smaller time steps to produce accurate solutions. In some cases we could not find a reasonable time step to match the error of ERK2 to that of ETDRDP.

Preprocessing still consumes a substantial portion of the computational time in the proposed scheme. The sensitivity of convergence to the shape of the error-versus-support-radius profile still requires further theoretical investigation. We will address these concerns in future work and evaluate the performance of the proposed scheme for problems in higher dimensions ($d > 2$).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding Declaration

This project did not receive funding.

References

[1] M. Hussain, A. Ghafoor, A. Hussain, S. Haq, I. Ali, S. U. Arifeen, A hybrid kernel-based meshless method for numerical approximation of multidimensional fisher's equation, Mathematics and Computers in Simulation 223 (2024) 130–157.

[2] H. Zhang, J. Yan, X. Qian, S. Song, Up to fourth-order unconditionally structure-preserving parametric single-step methods for semilinear parabolic equations, Computer Methods in Applied Mechanics and Engineering 393 (2022) 114817.

[3] Z. Wang, X. Huan, K. Garikipati, Variational system identification of the partial differential equations governing the physics of pattern-formation: Inference under varying fidelity and noise, Computer Methods in Applied Mechanics and Engineering 356 (2019) 44–74.

[4] B. Fornberg, N. Flyer, Solving pdes with radial basis functions, Acta Numerica 24 (2015) 215–258.

[5] H. Wendland, Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, Advances in computational Mathematics 4 (1) (1995) 389–396.

[6] W. Chen, Z.-J. Fu, C. Chen, Radial basis functions, in: Recent Advances in Radial Basis Function Collocation Methods, Springer, 2013, pp. 5–28.

[7] C.-M. Fan, C.-S. Chien, H.-F. Chan, C.-L. Chiu, The local rbf collocation method for solving the double-diffusive natural convection in fluid-saturated porous media, International Journal of Heat and Mass Transfer 57 (2) (2013) 500–503.

[8] J. Rashidinia, M. Khasi, G. E. Fasshauer, A stable gaussian radial basis function method for solving nonlinear unsteady convection–diffusion–reaction equations, Computers & Mathematics with Applications 75 (5) (2018) 1831–1850.

[9] H. B. Jebreen, A gaussian radial basis function-finite difference technique to simulate the hcir equation, Journal of Computational and Applied Mathematics 347 (2019) 181–195.

[10] M. UDDIN, A. HUSSAIN, et al., Rbf meshless method of lines for the numerical solution of nonlinear sine-gordon equation, Walailak Journal of Science and Technology (WJST) 11 (4) (2014) 349–360.

[11] A. Ali, S. Mukhtar, I. Hussain, A numerical meshless technique for the solution of some burgers' type equations, World Appl. Sci. J 14 (12) (2011) 1792–1798.

[12] B. Šarler, R. Vertnik, G. Kosec, et al., Radial basis function collocation method for the numerical solution of the two-dimensional transient nonlinear coupled burgers' equations, Applied Mathematical Modelling 36 (3) (2012) 1148–1160.

[13] S. Haq, A. Hussain, M. Uddin, Rbfs meshless method of lines for the numerical solution of time-dependent nonlinear coupled partial differential equations, Applied Mathematics 2 (4) (2011) 414.

[14] F. Soleymani, M. Barfeie, F. K. Haghani, Inverse multi-quadric rbf for computing the weights of fd method: Application to american options, Communications in Nonlinear Science and Numerical Simulation 64 (2018) 74–88.

[15] G. Yao, B. Šarler, et al., A comparative study of global and local meshless methods for diffusion-reaction equation, Computer Modeling in Engineering & Sciences 59 (2) (2010) 127.

[16] G. Yao, C.-S. Chen, H. Zheng, A modified method of approximate particular solutions for solving linear and nonlinear pdes, Numerical Methods for Partial Differential Equations 33 (6) (2017) 1839–1858.

[17] W. Li, K. Rubasinghe, G. Yao, L. Kuo, The modified localized method of approximated particular solutions for linear and nonlinear convection-diffusion-reaction pdes, Advances in Applied Mathematics and Mechanics 12 (5) (2020) 1113–1136.

[18] P. K. Mishra, G. E. Fasshauer, M. K. Sen, L. Ling, A stabilized radial basis-finite difference (rbf-fd) method with hybrid kernels, Computers & Mathematics with Applications 77 (9) (2019) 2354–2368.

[19] V. Bayona, An insight into rbf-fd approximations augmented with polynomials, Computers & Mathematics with Applications 77 (9) (2019) 2337–2353.

[20] R. Cavoretto, A. De Rossi, E. Perracchione, Efficient computation of partition of unity interpolants through a block-based searching technique, Computers & Mathematics with Applications 71 (12) (2016) 2568–2584.

[21] V. Shcherbakov, E. Larsson, Radial basis function partition of unity methods for pricing vanilla basket options, Computers & Mathematics with Applications 71 (1) (2016) 185–200.

[22] R. Tian, G. Yagawa, H. Terasaka, Linear dependence problems of partition of unity-based generalized fems, Computer Methods in Applied Mechanics and Engineering 195 (37-40) (2006) 4768–4782.

[23] Z. Wu, Compactly supported positive definite radial functions, Advances in computational mathematics 4 (1) (1995) 283–292.

[24] M. D. Buhmann, Radial functions on compact support, Proceedings of the Edinburgh Mathematical Society 41 (1) (1998) 33–46.

[25] T. Gneiting, Compactly supported correlation functions, Journal of Multivariate Analysis 83 (2) (2002) 493–508.

[26] R. Schaback, H. Wendland, Using compactly supported radial basis functions to solve partial differential equations, WIT Transactions on Modelling and Simulation 23 (2025).

[27] S. Zhu, Compactly supported radial basis functions: how and why? (2012).

[28] G. E. Fasshauer, Meshfree approximation methods with Matlab (With Cd-rom), Vol. 6, World Scientific Publishing Company, 2007.

[29] H. Wendland, Scattered data approximation, Vol. 17, Cambridge university press, 2004.

[30] S. Wong, Y. Hon, M. A. Golberg, Compactly supported radial basis functions for shallow water equations, Applied mathematics and computation 127 (1) (2002) 79–101.

[31] D. Young, C.-C. Tsai, T. Eldho, A.-D. Cheng, Solution of stokes flow using an iterative drbem based on compactly-supported, positive-definite radial basis function, Computers & Mathematics with Applications 43 (3-5) (2002) 607–619.

[32] W. Li, X. Liu, G. Yao, A local meshless collocation method for solving certain inverse problems, Engineering Analysis with Boundary Elements 57 (2015) 9–15.

[33] L. Keita, L. Azrar, A. A. Al-Ghamedi, A local meshfree approach based on compactly supported radial basis functions for 2d coupled fluid dynamics pdes on regular and irregular domains, Engineering Analysis with Boundary Elements 177 (2025) 106246.

[34] C. Chen, M. Marcozzi, S. Choi, The method of fundamental solutions and compactly supported radial basis functions-a meshless approach in 3d problems, boundary element methods xxi, eds. ca brebbia and h, Power, WIT Press, Boston, Southampton (1999) 313–322.

[35] W. Li, M. Li, C. Chen, X. Liu, Compactly supported radial basis functions for solving certain high order partial differential equations in 3d, Engineering Analysis with Boundary Elements 55 (2015) 2–9.

[36] M. Hochbruck, A. Ostermann, Exponential integrators, Acta Numerica 19 (2010) 209–286.

[37] M. Dehghan, V. Mohammadi, Two-dimensional simulation of the damped kuramoto–sivashinsky equation via radial basis function-generated finite difference scheme combined with an exponential time discretization, Engineering Analysis with Boundary Elements 107 (2019) 168–184.

[38] A. Ebrahimijahan, M. Dehghan, M. Abbaszadeh, Compact local integrated radial basis functions (integrated rbf) method for solving system of non–linear advection-diffusion-reaction equations to prevent the groundwater contamination, Engineering Analysis with Boundary Elements 121 (2020) 50–64.

[39] W. Zhao, Y. Hon, An accurate and efficient numerical method for neural field models with transmission delays, Engineering Analysis with Boundary Elements 135 (2022) 206–216.

[40] M. Caliari, A. Ostermann, S. Rainer, Meshfree exponential integrators, SIAM Journal on Scientific Computing 35 (1) (2013) A431–A452.

[41] S. M. Cox, P. C. Matthews, Exponential time differencing for stiff systems, Journal of Computational Physics 176 (2) (2002) 430–455.

[42] A.-K. Kassam, L. N. Trefethen, Fourth-order time-stepping for stiff PDEs, SIAM J. Sci. Comput. 26 (4) (2005) 1214–1233.

[43] E. Asante-Asamani, A. Khaliq, B. A. Wade, A real distinct poles exponential time differencing scheme for reaction–diffusion systems, Journal of Computational and Applied Mathematics 299 (2016) 24–34.

[44] E. Asante-Asamani, A. Kleefeld, B. A. Wade, A second-order exponential time differencing scheme for non-linear reaction-diffusion systems with dimensional splitting, J. Comput. Phys. 415 (2020) 109490.

[45] W. K. Attipoe, A. Kleefeld, E. Asante-Asamani, A fourth-order exponential time differencing scheme with real and distinct poles rational approximation for solving non-linear reaction-diffusion systems, Journal of computational and applied mathematics (2026) 117587.

[46] A. Khaliq, J. Martin-Vaquero, B. Wade, M. Yousuf, Smoothing schemes for reaction-diffusion systems with nonsmooth data, J. Comput. Appl. Math. 223 (1) (2009) 374–386.

[47] B. Kleefeld, A. Khaliq, B. Wade, An ETD Crank-Nicolson method for reaction-diffusion systems, Numer. Methods Partial Differ. Equ. 28 (4) (2012) 1309–1335.

[48] E. Asante-Asamani, A. Kleefeld, B. A. Wade, A fourth-order exponential time differencing scheme with dimensional splitting for non-linear reaction-diffusion systems, J. Comput. Appl. Math. (2025) 116568.

[49] E. Asante-Asamani, B. A. Wade, A dimensional splitting of etd schemes for reaction-diffusion systems, Communications in Computational Physics 19 (5) (2016) 1343–1356.

[50] E. Larsson, R. Schaback, Scaling of radial basis functions, IMA Journal of Numerical Analysis 44 (2) (2024) 1130–1152.

[51] S. Murphy, O. Ogunleye, K. Rubasinghe, G. Yao, Comparison between rbf and fdm for solving nonlinear time-dependent pdes using newton's method, in: Journal of Physics: Conference Series, Vol. 2766, IOP Publishing, 2024, p. 012166.

[52] M. Yağmurlu, A. Gagir, Numerical simulation of two dimensional coupled burgers equations by rubin-graves type linearization, Mathematical Sciences and Applications E-Notes 9 (4) (2021) 158–169.

[53] C. A. Fletcher, Generating exact solutions of the two-dimensional burgers' equations, International Journal for Numerical Methods in Fluids 3 (1983) 213–216.

[54] D. J. Griffiths, D. F. Schroeter, Introduction to quantum mechanics, Cambridge university press, 2018.

[55] M. Dehghan, D. Mirzaei, The meshless local petrov–galerkin (mlpg) method for the generalized two-dimensional non-linear schrödinger equation, Engineering Analysis with Boundary Elements 32 (9) (2008) 747–756.

[56] A. G. Bratsos, A. Khaliq, A conservative exponential time differencing method for the nonlinear cubic schrödinger equation, International Journal of Computer Mathematics 94 (2) (2017) 230–251.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- RBFETDnonlinearsupp.pdf