High-Precision Numerical Solutions of the Heat Equation via Mimetic Operators (MOLE)

Rafael Hernandez-Walls and Jose Castillo

May 2026

Publication Number: CSRCR2026-14

Computational Science & Engineering Faculty and Students Research Articles

Database Powered by the Computational Science Research Center Computing Group & Visualization Lab

COMPUTATIONAL SCIENCE & ENGINEERING

SAN DIEGO STATE UNIVERSITY

Computational Science Research Center
College of Sciences
5500 Campanile Drive
San Diego, CA 92182-1245
(619) 594-3430

High-Precision Numerical Solutions of the Heat Equation via Mimetic Operators (MOLE)

Rafael Hernandez-Walls[1] and Jose Castillo[2]

[1]ICOA, Mexico
[2]SDSU, USA

May 2026

Abstract

This study explores the application of mimetic discretization techniques for the numerical solution of 1D and 2D non-stationary diffusion equations. Utilizing the Mimetic Operators Library Enhanced (MOLE), we develop a framework that preserves the structural properties of the continuum operators. To optimize performance, we propose an integration strategy where the matrix exponential is computed only once, facilitating efficient long-term simulations with constant time steps. Numerical experiments demonstrate that the methodology achieves high-order convergence and reaches precision levels near machine epsilon (10^{-15}) even on coarse grids. The results confirm that the approach is unconditionally stable and computationally superior to traditional iterative schemes for parabolic systems.

1 Introduction

In the field of computational heat transfer and fluid dynamics, the fidelity of numerical simulations is governed by the ability of discrete operators to respect the underlying physical symmetries of the continuum. Traditional numerical models, predominantly those derived from standard Taylor-expansion-based finite difference methods, frequently suffer from consistency errors at the boundaries and a failure to preserve global conservation laws. The work of Hernandez-Walls [1] represents a significant paradigm shift by introducing mimetic discretization techniques to the solution of the 1D non-stationary heat equation.

The Mimetic philosophy is fundamentally distinct from local point-wise approximations. It aims to "mimic" the structural properties of the original partial differential equations (PDEs) by ensuring that discrete versions of fundamental integral theorems—specifically those of Gauss, Stokes, and Green—remain exactly satisfied in the discrete domain [2, 3]. By preserving these identities, mimetic operators maintain mathematical integrity even in the presence of complex boundary conditions. This work details the implementation, validation, and strategic advantages of utilizing the Mimetic Operators Library Enhanced (MOLE) for high-precision thermal modeling [4].

2 Mathematical Foundation and Problem Definition

This analysis centers, it will first be done in 1D and then in 2D, non-stationary diffusion (heat) equation:

$$\frac{\partial u}{\partial t} = \alpha \nabla^2 u \tag{1}$$

where $u(x,t)$ represents the temperature field and α is the thermal diffusivity coefficient. Physically, α is defined as the ratio of thermal conductivity to the product of density and specific calorific capacity,

$$\alpha = \frac{\kappa}{\rho C_p} \tag{2}$$

To establish a rigorous ground truth for validation, we employ a case where the initial condition $u(x, t = 0) = \sin(\pi x)$ acts as an eigenfunction of the Laplacian operator. This provides a distinct advantage for verifying the Principal Eigenvalue, applying the Laplacian to this function yields the function itself multiplied by the eigenvalue $\lambda = -\pi^2 \approx -9.8696$.

Empirical results using the MOLE library yield a principal eigenvalue of -9.8666. This close proximity (within 0.03%) demonstrates the library's ability to capture the operator's spectrum with high fidelity.

The resulting exact solution is $u(x, t) = \sin(\pi x) \exp(-\alpha \pi^2 t)$, allowing for a direct calculation of the absolute error across the spatial and temporal domains.

Transitioning from this continuous mathematical elegant to a computable model requires a robust spatial discretization that does not degrade this inherent precision.

Spatial discretization is the primary bottleneck in achieving high-order accuracy. Traditional finite difference schemes often lose precision at the boundaries where the stencil becomes one-sided. The MOLE library addresses this through a Staggered Grid architecture.

In this framework, the Laplacian (L) is a matrix operator of order k. MOLE generates these matrices such that the divergence and gradient satisfy the discrete Green's theorem. This matrix-based approach transforms the spatial problem into a system of equations perfectly suited for high-performance temporal integration.

3 Semi-Discrete Systems

The methodology shifts from fully discrete space-time schemes to a semi-discrete approach. By discretizing only the spatial component, the PDE is transformed into a system of linear ordinary differential equations (ODEs):

$$\frac{du}{dt} = \alpha L u \tag{3}$$

Unlike traditional iterative schemes (such as Forward Euler or Crank-Nicolson) which introduce temporal truncation errors, the approach by Hernandez-Walls (2014) treats the semi-discrete system as a system of linear ODEs with constant coefficients. The Matrix Exponential gives the exact solution to this system at any time:

$$u(x, t) = \exp(\alpha L t) u_0 \tag{4}$$

where u_0 represents the discretized initial condition (In vector form, updated at each time step), or

$$u(x, t + \Delta t) = \exp(\alpha L \Delta t) u_0 \tag{5}$$

where Δt is the time step.

3.1 Strategic Implementation Advantages

Unlike the Forward-Time Central-Space (FTCS) scheme, which is constrained by the stability limit $\Delta t \leqslant \|\Delta x\|^2 / 2\alpha$, the use of exponential integrators provides unconditional stability. The implementation utilizes the MATLAB expm function, which employs a Padé approximation to compute the matrix exponential (Moler and Van Loan, 2003).

For simulations with a constant time step Δt, the operator $\exp(\alpha L \Delta t)$ is calculated only once at the initialization phase. Subsequent temporal evolution is reduced to a single matrix-vector multiplication per step, providing a significant computational speedup while maintaining analytic level temporal precision.

From a computational complexity perspective, the proposed method offers a structural advantage over traditional iterative schemes. The initial calculation of the transition matrix $M = \exp(\alpha L \Delta t)$ using Padé's approximation has an asymptotic cost of $O(N^3)$, where N is the order of the matrix. However, since this calculation is performed only once at the beginning of the simulation, the operational cost per time step is drastically reduced to a matrix-vector multiplication of $O(N^2)$. For long-term simulations where $n_t \geqslant N$, this approach exceeds the efficiency of implicit methods that require the solution of a linear system at each iteration.

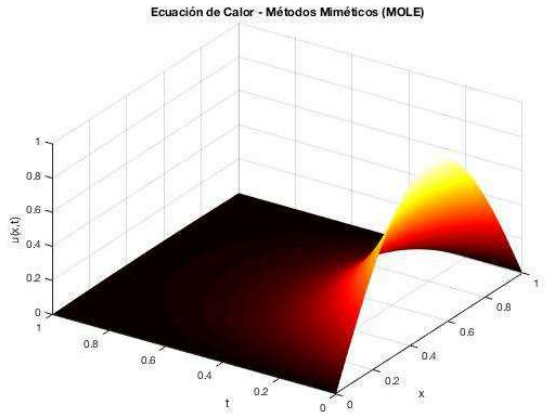


Figure 1: Numerical solution for the diffusion equation.

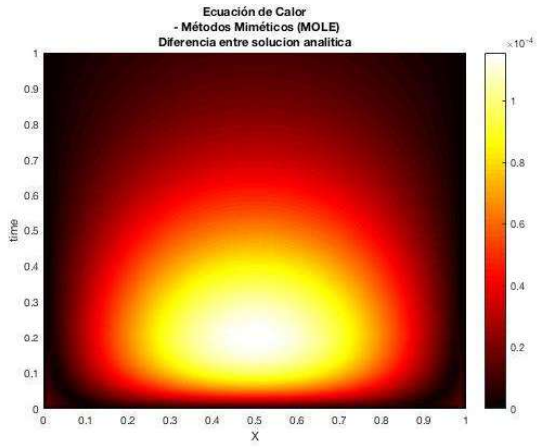


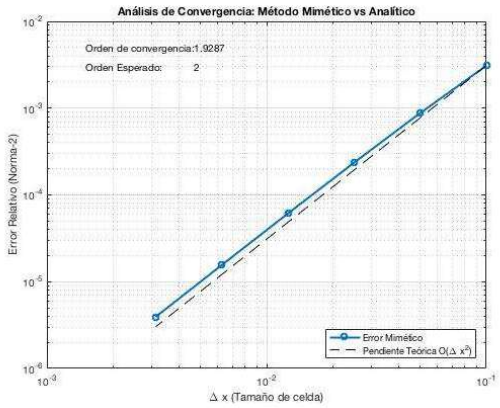
Figure 2: Error between theoretical solution and numerical solution for $k = 2$, using $\Delta x = 0.02$.

4 Convergence Analysis and Experimental Results

The ultimate validation of the MOLE library is its performance under mesh refinement. By observing the rate at which the relative error decreases as the grid spacing Δx is reduced, we verify the theoretical order of accuracy.

The experimental slopes for $k = 2$ and $k = 4$ closely match theoretical expectations. For $k = 6$ and $k = 8$, the experimental slopes appear higher than theoretical values; this is a result of selective regression. To maintain technical integrity, the slopes were calculated using only the thickest mesh points. In these high-order schemes, the error reduces so rapidly that it reaches the machine's precision floor (10^{-15}) before a full refinement study can be completed on fine meshes.

When evaluating high-order schemes ($k = 6$ and $k = 8$), a numerical saturation phenomenon is observed where the relative error reaches the limit of double-precision arithmetic (10^{-15}). This saturation defines a threshold of strategic utility for high orders, the truncation error of the mimetic operator becomes negligible compared to the machine's rounding error. The practical implication is significant: using the MOLE library allows achieving precisions of 10^{-12} with extremely coarse meshes, resulting in massive savings of memory and computation time compared to second-order schemes that would require much denser meshes to approach that level of error.



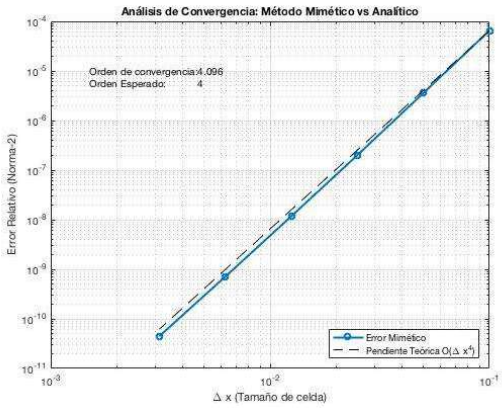
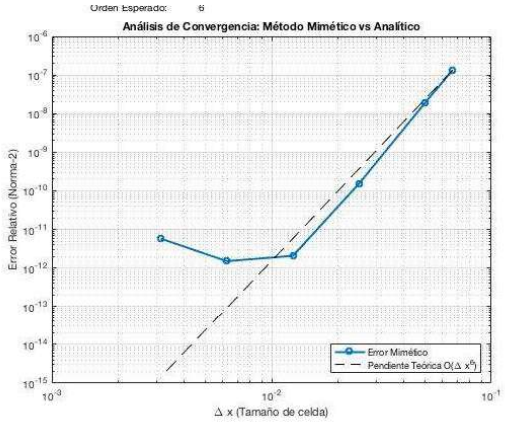


Figure 3: Comparison of the order of the mimetic operator and the number of cells, for two orders k=2 (left) and k=4 (right).



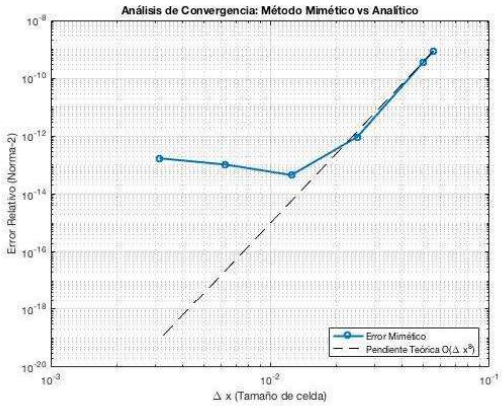


Figure 4: Comparison of the order of the mimetic operator and the number of cells, for two orders k=6 (left) and k=8 (right).

Table 1: Convergence Performance of Mimetic Orders (k). Values marked with * are calculated via selective regression on coarse mesh points to avoid saturation.

Mimetic Order (k)	Theoretical Slope	Experimental Slope
2	2.0	1.9287
4	4.0	4.0960
6	6.0	6.6343*
8	8.0	8.5651*

5 Numerical Saturation and Precision Limits

There exists a strategic limit of utility in numerical refinement. As the order of the mimetic operators ($k = 6, 8$) increases, the system encounters Error Saturation. This occurs when the truncation error of the method becomes smaller than the Double-Precision Floating Point limit (10^{-15} of the computing environment).

Furthermore, as the order k increases, the matrix bandwidth grows. While high-order mimetic operators are less sparse than second-order FD operators —increasing the cost of the initial matrix exponential calculation- the ability to achieve errors of 10^{-12} using extremely coarse meshes results in a massive net gain in memory and compute efficiency. The deviation from the theoretical slope at fine scales simply defines the threshold of machine precision and does not invalidate the consistency of the MOLE library.

6 Extension to the two-dimensional case: Rectangular thermal plate

The evolution of the temperature in a square metal plate of dimensions is analyzed, $L_x = 1$ ans $L_y = 1$.

Unlike the one-dimensional case, this experiment seeks to model heat diffusion in two simultaneous directions, maintaining constant Dirichlet boundary conditions on the four edges of the plate to observe the heat flow towards the steady state.

Boundary Conditions and Parameters

- Top: 100^oC
- Botoom 0^oC
- Left/Right 50^oC
- • Initial Condition: Uniform plate to 20^oC (except at the edges)
- • Diffusivity $\alpha = 0.01$

6.1 2D spatial discretization

To construct the Laplacian operator 2D (∇^2) Using the MOLE library, the Kronecker product, ($\otimes$), is employed. This method allows extending one-dimensional operators (L_x, L_y) to the two-dimensional domain as follows (Steeb, 2006):

$$L2D = (I_y \otimes L_x) + (L_y \otimes I_x) \tag{6}$$

Where I represents the identity matrix of corresponding dimensions. This formulation preserves the mimetic structure in every direction of space, ensuring that the conservation properties are maintained across the entire surface of the plate.

6.2 Temporary integration and stability

Following the methodology of Hernández-Walls (2014), the resulting system of ordinary differential equations: $du/dt = \alpha L2Du$.

This is solved using the exponential of the matrix. A critical aspect of the implementation is the treatment of the boundaries. To ensure that the temperatures at the edges remain constant over time ($du/dt = 0$), the rows of the system matrix A corresponding to the boundary nodes are made zero:

$$A_{i,j} = 0 \, \forall \, i \; \in \; border \; nodes \tag{7}$$

This adjustment allows the use of the transition matrix $M = \exp(A\Delta t)$.

iteratively, allowing an unconditionally stable simulation without the time-step constraints (CFL) common in explicit finite difference methods, This technique does not destroy the overall consistency of the mimetic method.

6.3 Pre-calculated Transition Matrix vs. Iterative Solvers

A key advantage of the method described by Hernández-Walls (2014), when implemented with the MOLE library, is the use of a pre-calculated Transition Matrix (M).

The most computationally expensive operation is the calculation of the matrix exponential $\exp(A\Delta t)$ via Padé approximation. However, since the system matrix A and the time step Δt are constant, this operation is performed only once at the beginning of the simulation.

In each subsequent time step, the update of the temperature vector u is reduced to a simple Matrix-Vector Multiplication: $u_{t+1} = Mu_t$.

6.4 Comparison with Standard Solvers

The Explicit Schemes (Euler) require very small Δt (CFL condition) to avoid instability, leading to thousands of iterations.

Implicit Schemes (Crank-Nicolson) require solving a linear system,

$$(I - \Theta\Delta tA)x = b$$

at every single step, which is numerically taxing for large 2D grids.

Exponential Approach provides unconditional stability (like implicit methods) but with the low per-step cost of an explicit method.

For a 30×30 grid (900 nodes), the pre-calculated matrix allows for real-time visualization of the heat flow. As the grid size increases, the sparsity of the mimetic operators in MOLE ensures that the memory footprint remains manageable, while the exponential transition maintains a high level of temporal precision that exceeds traditional finite difference approximations.

6.5 Analysis of expected results

The temperature distribution is expected to evolve from the initial condition of 20^oC towards a smooth profile where isotherms connect the upper heat sources with the lower heat sinks (figure 5). MOLE's second-order accuracy (k=2) ensures that the temperature gradient at the plate corners is resolved without spurious numerical oscillations.

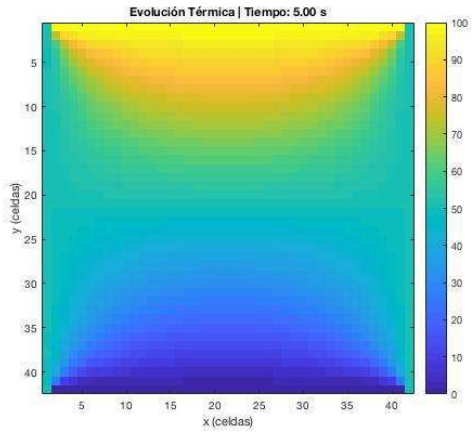


Figure 5: Simulation of the 2D diffusion equation for a metal plate, t=5 seconds.

6.6 Comparative Summary: 1D VS. 2D MIimetic Implementations

The following table summarizes the key technical differences and similarities between the two experiments conducted using the MOLE library and the Matrix Exponential approach The same fundamental MATLAB function (expm) handles both cases, proving the robustness of the linear ODE system approach.

While the 2D case involves a significantly larger matrix, the Kronecker Product implementation in MOLE keeps the code clean and mathematically identical to the continuous operator definitions.

In both cases, using $k = 2$ or higher provides a superior resolution of the thermal gradients near the boundaries compared to standard finite differences.

7 Conclusion

The developed numerical framework demonstrates that integrating the MOLE library with exponential integration schemes offers a dual advantage in terms of precision and performance. On the one hand, while the initial calculation of the matrix exponential involves a $O(N^3)$ computational cost, the unconditional stability of the method allows efficient amortization of this effort in long-term simulations, where the cost per time step is reduced to an operation $O(N^2)$. On the other hand, the ability of high-order schemes $k \geqslant 6$ to reach the machine precision threshold (10^{-15}) even on coarse grids redefines spatial refinement strategies. This error saturation phenomenon does not represent a limitation, but rather proof of the method's efficiency, enabling high-fidelity solutions with significantly fewer memory resources compared to second-order finite difference schemes. Collectively, these findings position the proposed methodology as a robust and computationally economical alternative to solve complex parabolic systems.

References

[1] R. Hernández-Walls, J. Castillo, E. Rojas-Mayoral, and J. Girón-Nava, "Semi-discrete numeric solution for the non-stationary heat equation using mimetic techniques," *European Journal of Physics*, vol. 35, no. 6, p. 065013, 2014.

[2] J. E. Castillo and R. Grone, "A matrix analysis approach to higher-order approximations for divergence and gradients satisfying a global conservation law," *SIAM Journal on Matrix Analysis and Applications*, vol. 25, no. 1, pp. 128–142, 2003.

[3] J. Corbino and J. E. Castillo, "High-order mimetic finite-difference operators satisfying the extended gauss divergence theorem," *Journal of Computational and Applied Mathematics*, vol. 364, p. 112326, 2020.

[4] J. Corbino, M. A. Dumett, and J. E. Castillo, "Mole: Mimetic operators library enhanced," *Journal of Open Source Software*, vol. 9, no. 99, p. 6288, 2024. [Online]. Available: https://doi.org/10.21105/joss.06288

Appendix A

Listing 1: MATLAB code

```
clear all, clc, close all
% Author: RHW-2026 1D
% Parametros del problema
Lx = 1;              % Length
nx = 50;             % Cells (m)
k = 2;               % Order of mimetic precision (second order)
alpha = 0.5;         % Diffusion coefficient
tmax = 1.0;
dx = Lx / nx;
nt = 100;
dt = tmax / nt;
% Mimetic operator ( Lap) using MOLE library
Lap = lap(k, nx, dx);   % Mimetic Laplacian
% Set initial condition on staggered grid
xc = (dx/2 : dx : Lx-dx/2)';
xgrid = [0, xc', Lx];
```

```
U_total(:, 1) = [0; sin(pi * xc); 0]; % Boundary nodes included
% Compute Matrix Exponential via Pade approximation, only once
EM = expm(alpha * full(Lap) * dt);
% Solve for subsequent time steps
for i = 1:nt
  U_total(:, i+1) = EM * U_total(:, i);
end
% Graphic section
[T, X] = meshgrid(0:dt:tmax, xgrid);
surf(X, T, U_total);
shading interp; colormap hot;
xlabel('x-space'); ylabel('t-time'); zlabel('u(x,t)');
title('Heat␣eq.␣Mimetic␣Methods␣(MOLE)');
```