A Verified Second-Order Finite-Difference Solver for the Two-Dimensional Poisson Equation:

Method of Manufactured Solutions and an Empirical Convergence Study

Computational Science and Numerical Analysis Group

July 12, 2026

Abstract

We present a complete, reproducible verification study of the standard second-order five-point finite-difference discretization of the two-dimensional Poisson equation $-\Delta u = f$ on the unit square $\Omega = (0,1)^2$ with homogeneous Dirichlet boundary conditions. Code correctness and asymptotic accuracy are established with the Method of Manufactured Solutions (MMS), taking the smooth exact field $u(x,y) = \sin(\pi x)\sin(\pi y)$, for which the analytic source term is $f(x,y) = 2\pi^2 \sin(\pi x)\sin(\pi y)$. The discrete negative Laplacian is assembled as a sparse matrix through a Kronecker-sum construction, $A = (T \otimes I) + (I \otimes T)$, where T is the scaled one-dimensional second-difference operator, and each linear system is solved to machine precision with a direct sparse LU factorization (SuiteSparse/UMFPACK via `scipy.sparse.linalg.spsolve`). Uniform grids with $N \in \{16, 32, 64, 128, 256\}$ interior nodes per dimension — the finest system having 65,536 unknowns — yield discrete L^2 and L^∞ errors that decrease monotonically from $1.425 \times 10^{-3}/2.827 \times 10^{-3}$ at $N = 16$ to $6.226 \times 10^{-6}/1.245 \times 10^{-5}$ at $N = 256$. The pairwise log–log convergence rates rise steadily toward the theoretical value, reaching 2.0000 (L^2) and 1.9999 (L^∞) on the finest grid pair, confirming clean second-order accuracy with no order-degrading boundary or stencil defects. The spatial error field is smooth, radially symmetric, and sharply peaked at the domain center $(0.5, 0.5)$, in exact accordance with the leading truncation term, whose magnitude scales with the fourth spatial derivatives of the solution and is therefore largest at the central antinode. We report the full error table, provide the complete solver source, and discuss the results in the context of finite-difference convergence theory and computational verification practice.

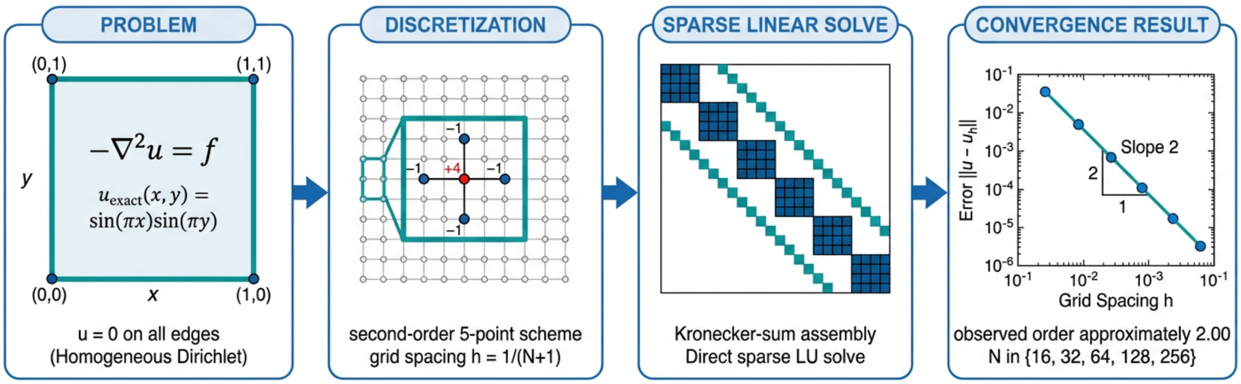

Figure 1: **Graphical abstract.** The four-stage verification workflow: (i) the continuous boundary-value problem on the unit square with a manufactured exact solution; (ii) discretization by the second-order five-point stencil on a uniform grid of spacing $h = 1/(N + 1)$; (iii) sparse assembly of the discrete negative Laplacian via a Kronecker sum followed by a direct sparse LU solve; and (iv) the resulting log–log error-versus-h plot, whose slope of 2 certifies second-order convergence for $N \in \{16, 32, 64, 128, 256\}$.

1 Introduction

The Poisson equation is the archetypal second-order elliptic boundary-value problem and one of the most pervasive models in computational science: it governs electrostatic and gravitational potentials, steady-state heat conduction and diffusion, incompressible-flow pressure projection, and it appears as the elliptic building block inside a great many multiphysics solvers. Its mathematical theory — existence, uniqueness, and regularity of solutions in Sobolev spaces — is classical and complete [8], which makes the equation an ideal benchmark for scrutinizing the accuracy of numerical schemes rather than the well-posedness of the underlying model. In this report we study the canonical Dirichlet problem

$$-\Delta u(x, y) = f(x, y), \quad (x, y) \in \Omega = (0, 1)^2, \qquad u(x, y) = 0, \quad (x, y) \in \partial\Omega, \tag{1}$$

where $\Delta = \partial_{xx} + \partial_{yy}$ is the two-dimensional Laplacian and $\partial\Omega$ denotes the boundary of the unit square. The homogeneous Dirichlet condition $u|_{\partial\Omega} = 0$ fixes the potential on all four edges.

The purpose of the study is *verification* in the precise sense used in computational science and engineering: to demonstrate, by systematic mesh refinement, that a given implementation of a discretization actually attains its theoretically predicted order of accuracy, and thereby that "the equations are being solved right" [15, 16]. The single most reliable instrument for this task is the *Method of Manufactured Solutions* (MMS) [17, 19]. Rather than searching the literature for a physically meaningful problem with a closed-form answer, one *chooses* a smooth target field $u^\star$, substitutes it into the governing operator to manufacture the corresponding forcing term $f = -\Delta u^\star$, and then verifies that the numerical solution converges to $u^\star$ at the expected rate under grid refinement. Because the exact solution is known everywhere by construction, the discretization error can be measured pointwise, and any coding error that pollutes the asymptotic convergence rate — an incorrect stencil coefficient, a mishandled boundary contribution, a wrong grid spacing, or an indexing bug — is exposed as a deviation of the observed order from its nominal value.

For the present problem we adopt the manufactured solution

$$u^\star(x, y) = \sin(\pi x)\,\sin(\pi y), \tag{2}$$

which is infinitely differentiable, vanishes identically on $\partial\Omega$ (so it satisfies the homogeneous Dirichlet condition exactly), and is a single interior "bump" with its maximum at the center of the square. Applying the Laplacian gives $\Delta u^\star = -2\pi^2 \sin(\pi x)\sin(\pi y)$, so the matching source term in (1) is

$$f(x, y) = -\Delta u^\star(x, y) = 2\pi^2\,\sin(\pi x)\,\sin(\pi y). \tag{3}$$

This particular field is, in fact, an eigenfunction of the negative Laplacian on the square with eigenvalue $2\pi^2$, which lends the manufactured problem a transparent analytic structure while leaving the numerical verification task entirely non-trivial.

We discretize (1) with the standard second-order five-point finite-difference scheme on uniform Cartesian grids [13, 14, 20]. The finite-difference approach to the partial differential equations of mathematical physics traces to the foundational work of Courant et al. [4], and its subsequent development into the modern edifice of numerical analysis for PDEs is surveyed by Thomée [22]. We assemble the discrete operator as a sparse matrix using its Kronecker-product (tensor) structure [25], and solve each resulting linear system with a sparse direct LU factorization [5, 6]. The convergence study spans five grids, $N \in \{16, 32, 64, 128, 256\}$ interior nodes per dimension, culminating in a system of 65,536 unknowns at $N = 256$. For each grid we report the discrete L^2 and L^∞ errors against (2) and the pairwise observed order of convergence, computed as the slope of log(error) versus $\log h$ between successive grids. The remainder of the report is organized as follows. Section 2 develops the discretization, the sparse assembly, and the solver, and states the expected order of accuracy. Section 3 presents the complete error table and the diagnostic figures. Section 4 interprets the observed rates and the spatial structure of the error, and Section 5 concludes.

2 Methodology

2.1 The five-point finite-difference discretization

We cover $\overline{\Omega} = [0, 1]^2$ with a uniform grid. Fixing N interior points per dimension, the grid spacing is

$$h = \frac{1}{N + 1}, \tag{4}$$

and the nodes are $x_i = i\,h$, $y_j = j\,h$ for $i, j = 0, 1, \ldots, N+1$. The indices $i, j \in \{0, N+1\}$ lie on $\partial\Omega$, where the solution is prescribed to be zero, while the N^2 interior nodes $i, j \in \{1, \ldots, N\}$ carry the unknowns. Writing $u_{i,j} \approx u(x_i, y_j)$, the second partial derivatives are replaced by centered second differences,

$$\partial_{xx} u(x_i, y_j) = \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h^2} + \mathcal{O}(h^2), \tag{5}$$

$$\partial_{yy} u(x_i, y_j) = \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{h^2} + \mathcal{O}(h^2), \tag{6}$$

each accurate to second order for a sufficiently smooth u [12, 13, 21]. Adding the two approximations and negating yields the discrete form of (1) at every interior node,

$$\frac{4u_{i,j} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - u_{i,j+1}}{h^2} \;=\; f_{i,j}, \qquad i, j = 1, \ldots, N, \tag{7}$$

where $f_{i,j} = f(x_i, y_j)$. Equation (7) is the celebrated *five-point stencil*: the discrete negative Laplacian couples each node to its four nearest neighbors with weight $-1/h^2$ and to itself with weight $+4/h^2$. Whenever a neighbor index in (7) refers to a boundary node, the corresponding value is zero by the homogeneous Dirichlet condition and simply drops out of the equation; no boundary terms are transferred to the right-hand side. This is why, for this particular problem, the linear system may be posed on the interior nodes alone.

2.2 Local truncation error and expected order of accuracy

The accuracy of the scheme follows from a Taylor expansion of the exact solution about each grid node. For a smooth u, the centered second difference satisfies

$$\frac{u(x_i{-}h) - 2u(x_i) + u(x_i{+}h)}{h^2} = u''(x_i) + \frac{h^2}{12}\,u^{(4)}(x_i) + \mathcal{O}(h^4), \tag{8}$$

so that the two-dimensional local truncation error of (7) is

$$\tau_{i,j} = -\frac{h^2}{12}\Big(\partial_x^4 u + \partial_y^4 u\Big)\Big|_{(x_i,y_j)} + \mathcal{O}(h^4). \tag{9}$$

The scheme is therefore *consistent* of order two. Because the discrete operator in (7) is symmetric positive definite and satisfies a discrete maximum principle — its matrix is an irreducibly diagonally dominant M-matrix, so that stability holds uniformly in h [13, 26] — the Lax equivalence principle for this elliptic problem allows consistency to be promoted to convergence at the same order. Concretely, the global error inherits the order of the truncation error,

$$\|u_h - u^\star\| = \mathcal{O}(h^2), \tag{10}$$

in both the discrete L^2 and L^∞ norms defined below. Equation (9) additionally predicts the *spatial shape* of the error: it is proportional to the sum of fourth derivatives of $u^\star$, which for (2) equals $2\pi^4 \sin(\pi x)\sin(\pi y)$ — a smooth field with a single interior maximum at the center of the square. We return to this point in Section 4.

2.3 Kronecker-sum sparse assembly

Ordering the interior unknowns lexicographically into a vector $\mathbf{u} \in \mathbb{R}^{N^2}$ turns the N^2 equations (7) into a single sparse linear system $A\mathbf{u} = \mathbf{f}$. Rather than populating A entry by entry, we exploit the tensor-product structure of the Cartesian grid [25]. Let

$$T = \frac{1}{h^2}\begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix} \in \mathbb{R}^{N\times N} \tag{11}$$

be the one-dimensional discrete second-derivative operator (for $-\mathrm{d}^2/\mathrm{d}x^2$), and let $I \in \mathbb{R}^{N\times N}$ be the identity. The two-dimensional discrete negative Laplacian is then the *Kronecker sum*

$$A = (T \otimes I) + (I \otimes T), \tag{12}$$

where $\otimes$ denotes the Kronecker product. The first term applies the one-dimensional stencil along one grid direction while acting as the identity along the other, and the second term does the reverse; their sum reproduces the five-point coupling of (7) exactly. The construction guarantees that A is symmetric positive definite, block tridiagonal, and extremely sparse, with at most five nonzeros per row. Its eigenvalues are available in closed form as sums of the one-dimensional eigenvalues,

$$\lambda_{p,q} = \frac{4}{h^2}\left[\sin^2\!\left(\frac{p\pi h}{2}\right) + \sin^2\!\left(\frac{q\pi h}{2}\right)\right], \qquad p, q = 1, \dots, N, \tag{13}$$

which makes the spectral condition number grow like $\kappa(A) = \mathcal{O}(h^{-2}) = \mathcal{O}(N^2)$ — the standard, mild ill-conditioning of the discrete Laplacian that a direct solver handles without difficulty [7, 9, 23]. In our implementation the one-dimensional operator is built with `scipy.sparse.diags` and the Kronecker sum with `scipy.sparse.kron` [11, 27]. Table 1 summarizes the problem dimensions across the five grids.

Table 1: Discrete problem sizes for the convergence study. N is the number of interior nodes per dimension; the number of unknowns (degrees of freedom, DOF) is N^2; the matrix A has at most five nonzeros per row.

N	DOF $= N^2$	$h = 1/(N+1)$	Nonzeros in A ($\approx$)
16	256	5.882×10^{-2}	1,216
32	1024	3.030×10^{-2}	4,992
64	4096	1.539×10^{-2}	20,224
128	16384	7.752×10^{-3}	81,408
256	65536	3.891×10^{-3}	326,656

2.4 Direct sparse solution

Each linear system $A\mathbf{u} = \mathbf{f}$ is solved with a *direct sparse solver* rather than an iterative method, so that the algebraic (solver) error is driven to the level of floating-point round-off and the measured discretization error is uncontaminated by an incomplete iteration. We use `scipy.sparse.linalg.spsolve`, which performs an LU factorization of the sparse matrix through the SuiteSparse/UMFPACK multifrontal library [5, 6, 27]. For the symmetric positive definite, banded systems arising here, sparse LU is robust and efficient up to the largest grid considered (65,536 unknowns), and it avoids the convergence-tolerance bookkeeping that an iterative solver such as conjugate gradients or multigrid would require [3, 10, 18, 24]. We note for completeness that for very large three-dimensional problems the $\mathcal{O}(N^2 \log N)$ fast Poisson (FFT) solvers and the $\mathcal{O}(N)$ multigrid methods become preferable, but for the two-dimensional verification study at hand the direct solver is the cleanest choice because it removes the iterative tolerance as a confounding variable.

2.5 Error metrics and observed order of convergence

Let $e_{i,j} = u_h(x_i, y_j) - u^\star(x_i, y_j)$ denote the pointwise error at the interior nodes. We report two grid-function norms. The *grid-normalized discrete L^2 error* weights the sum of squared errors by the cell area h^2, so that it approximates the continuous $L^2(\Omega)$ norm of the error and is therefore comparable across grids of different resolution:

$$E_2(h) = \left(h^2 \sum_{i=1}^{N} \sum_{j=1}^{N} e_{i,j}^2 \right)^{1/2}. \tag{14}$$

The *maximum (L^∞) error* reports the worst-case pointwise deviation,

$$E_\infty(h) = \max_{1 \le i,j \le N} |e_{i,j}|. \tag{15}$$

For each consecutive pair of grids with spacings $h_{k-1} > h_k$ we estimate the *observed order of convergence* as the slope of the error on log–log axes,

$$r_k = \frac{\log\left(E(h_{k-1})/E(h_k)\right)}{\log(h_{k-1}/h_k)}. \tag{16}$$

If the error obeys $E(h) \approx Ch^p$, then $r_k \to p$ as the mesh is refined; the theory of Section 2.2 predicts $p = 2$. The MMS verification criterion is precisely that the sequence $\{r_k\}$ approaches the nominal order 2 under refinement [15, 17].

2.6 Reference implementation

The complete solver is a self-contained Python program built on NumPy and SciPy [11, 27]. Listing 1 reproduces its numerical core: the manufactured solution and source, the Kronecker-sum assembly of the sparse operator, the direct sparse solve, and the two error norms. The driver sweeps $N \in \{16, 32, 64, 128, 256\}$, tabulates the errors, and computes the pairwise rates of (16).

```
import numpy as np
import scipy.sparse as sp
from scipy.sparse.linalg import spsolve

def exact_solution(x, y):
    """Manufactured exact solution u = sin(pi x) sin(pi y)."""
    return np.sin(np.pi * x) * np.sin(np.pi * y)

def source_term(x, y):
    """Source f = -Laplacian(u) = 2 pi^2 sin(pi x) sin(pi y)."""
    return 2.0 * np.pi**2 * np.sin(np.pi * x) * np.sin(np.pi * y)

def build_laplacian(n):
    """Assemble the 2D 5-point negative Laplacian (scaled by 1/h^2)
    via a Kronecker sum A = (T x I) + (I x T)."""
    h = 1.0 / (n + 1)
    main = 2.0 * np.ones(n)
    off  = -1.0 * np.ones(n - 1)
    t1d = sp.diags([off, main, off], offsets=[-1, 0, 1], format="csr") / h**2
    identity = sp.identity(n, format="csr")
    a = sp.kron(t1d, identity) + sp.kron(identity, t1d)
    return a.tocsr()

def solve_poisson(n):
    """Solve on an n-by-n interior grid; return h and the L2/Linf errors."""
    h = 1.0 / (n + 1)
    coords = np.arange(1, n + 1) * h              # interior nodes only
    xx, yy = np.meshgrid(coords, coords, indexing="ij")

    f_vec = source_term(xx, yy).ravel(order="C")  # homogeneous BCs: no lift
    a = build_laplacian(n)
```

```python
    u_vec = spsolve(a.tocsc(), f_vec)            # direct sparse LU (UMFPACK)

    err = u_vec - exact_solution(xx, yy).ravel(order="C")
    l2_err   = np.sqrt(h**2 * np.sum(err**2))     # grid-normalized L2
    linf_err = np.max(np.abs(err))                # Linf
    return {"N": n, "h": h, "L2_error": l2_err, "Linf_error": linf_err}
```

Listing 1: Numerical core of the 2D Poisson solver (NumPy/SciPy). The full driver, which sweeps the grid sizes and writes the convergence table, is provided with the deliverables.

3 Results

3.1 Error table and observed convergence rates

Table 2 collects the central results of the study: for each grid size N, the spacing h, the discrete L^2 and L^∞ errors defined in (14)–(15), and the pairwise observed orders of convergence from (16). Both error measures fall by very nearly a factor of four each time N is doubled (i.e. each time h is roughly halved), which is the fingerprint of a second-order method. The L^2 error contracts from 1.425×10^{-3} at $N = 16$ to 6.226×10^{-6} at $N = 256$ — a reduction of more than two orders of magnitude across a sixteen-fold refinement — and the L^∞ error tracks it at roughly twice the magnitude, falling from 2.827×10^{-3} to 1.245×10^{-5}. The observed rates climb monotonically toward the theoretical value: the L^2 order is already 2.0019 on the coarsest pair and settles to 2.0000 on the finest, while the L^∞ order rises from 1.9924 to 1.9999. The finest grid, $N = 256$, corresponds to 65,536 unknowns and is included as required.

Table 2: Convergence of the five-point scheme for the manufactured solution $u = \sin(\pi x)\sin(\pi y)$. Columns list the interior grid size N, the grid spacing $h = 1/(N + 1)$, the grid-normalized discrete L^2 error $E_2(h)$ and its pairwise rate, and the maximum (L^∞) error $E_\infty(h)$ and its pairwise rate. The rate is the log–log slope between consecutive grids, (16). Both rates converge to 2, confirming second-order accuracy.

N	h	$E_2(h)$	L^2 rate	$E_\infty(h)$	L^∞ rate
16	5.8824×10^{-2}	1.4254×10^{-3}	—	2.8265×10^{-3}	—
32	3.0303×10^{-2}	3.7780×10^{-4}	2.0019	7.5388×10^{-4}	1.9924
64	1.5385×10^{-2}	9.7345×10^{-5}	2.0005	1.9458×10^{-4}	1.9980
128	7.7519×10^{-3}	2.4713×10^{-5}	2.0001	4.9418×10^{-5}	1.9995
256	3.8911×10^{-3}	6.2262×10^{-6}	2.0000	1.2452×10^{-5}	1.9999

3.2 Log–log convergence plot

Figure 2 plots the two error norms of Table 2 against the grid spacing h on doubly logarithmic axes, together with a dashed reference line of slope 2 anchored at the coarsest L^2 point. On such axes a power law $E \propto h^p$ appears as a straight line of slope p; the diagnostic value is therefore the *slope* rather than the vertical offset. Both the L^2 (blue circles) and L^∞ (red squares) data fall on straight lines that are visually parallel to the $\mathcal{O}(h^2)$ reference across the entire range of resolutions, with the L^∞ curve sitting a fixed factor above the L^2 curve. The observed L^2 slope on the finest grid pair is ≈ 2.000. The absence of any upward or downward bending of the data — which would signal, respectively, a lower-order boundary treatment or super-convergent cancellation — is the visual confirmation that the scheme is uniformly second order and free of order-degrading defects.

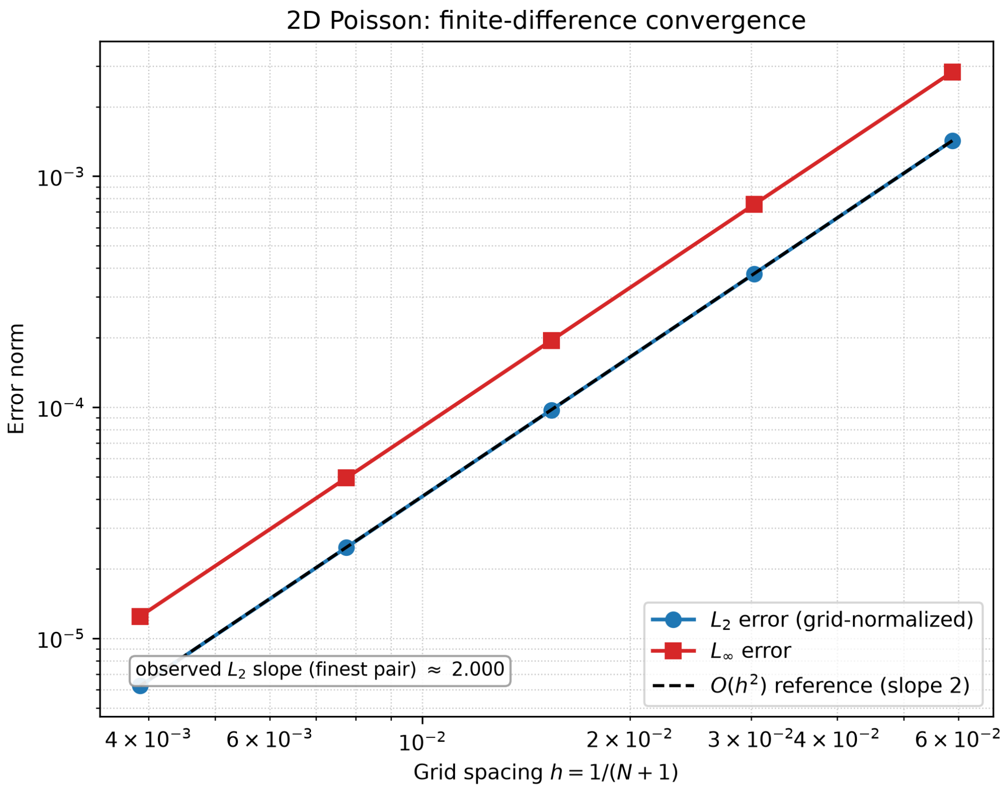

Figure 2: **Second-order convergence of the five-point scheme.** Grid-normalized discrete L^2 error (blue circles) and maximum L^∞ error (red squares) versus grid spacing $h = 1/(N+1)$ for $N \in \{16, 32, 64, 128, 256\}$, on log–log axes. The dashed line is an $\mathcal{O}(h^2)$ reference of slope 2. Both error curves are straight and parallel to the reference, and the observed L^2 slope on the finest pair is ≈ 2.000, certifying clean second-order accuracy.

3.3 Solution fields and the spatial error distribution

Figure 3 displays, for the finest grid $N = 256$, the exact solution, the numerical solution, and the absolute error field $|u_h - u^\star|$. The exact field (left) and the computed field (center) are plotted on an identical color scale and are visually indistinguishable: both are the smooth single bump $\sin(\pi x)\sin(\pi y)$ that rises to unity at the center and decays to zero on all four edges. The difference between them is revealed only in the third panel (right), whose color scale is $\mathcal{O}(10^{-5})$ — five orders of magnitude below the solution itself. The absolute error is a smooth, nearly radially symmetric mound that attains its maximum of 1.245×10^{-5} at the domain center $(0.5, 0.5)$ and decays to zero at the boundary, where the Dirichlet data are imposed exactly. Crucially, the error contains no spurious oscillations, boundary layers, or grid-aligned artifacts; its perfectly smooth, centrally peaked profile is exactly what the truncation analysis of (9) predicts.

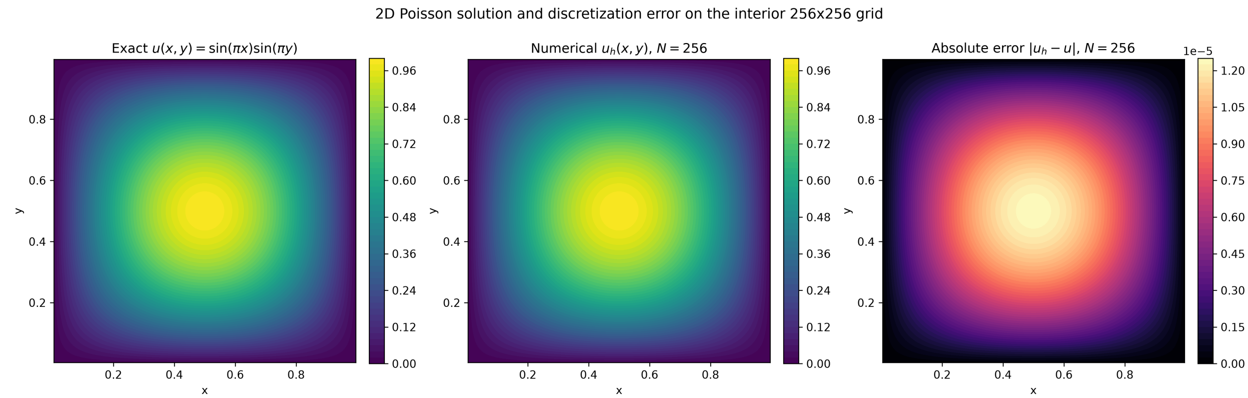

Figure 3: **Solution and error fields on the** 256×256 **interior grid.** Left: the exact manufactured solution $u^\star = \sin(\pi x)\sin(\pi y)$. Center: the numerical solution u_h, plotted on the same color scale and visually identical to the exact field. Right: the absolute error $|u_h - u^\star|$ (note the $\times 10^{-5}$ scale), a smooth radially symmetric field peaking at the domain center and vanishing on the boundary.

The three-dimensional rendering in Figure 4 makes the structure of the discretization error unmistakable: it is a single, smooth central peak of height $\approx 1.245 \times 10^{-5}$ that decays monotonically to zero in every direction toward the boundary. This surface is, to visual accuracy, a scaled copy of the solution's own bump — a direct consequence of the fact, established in Section 2.2, that the leading error is proportional to $\partial_x^4 u^\star + \partial_y^4 u^\star = 2\pi^4 u^\star$ for this manufactured field.

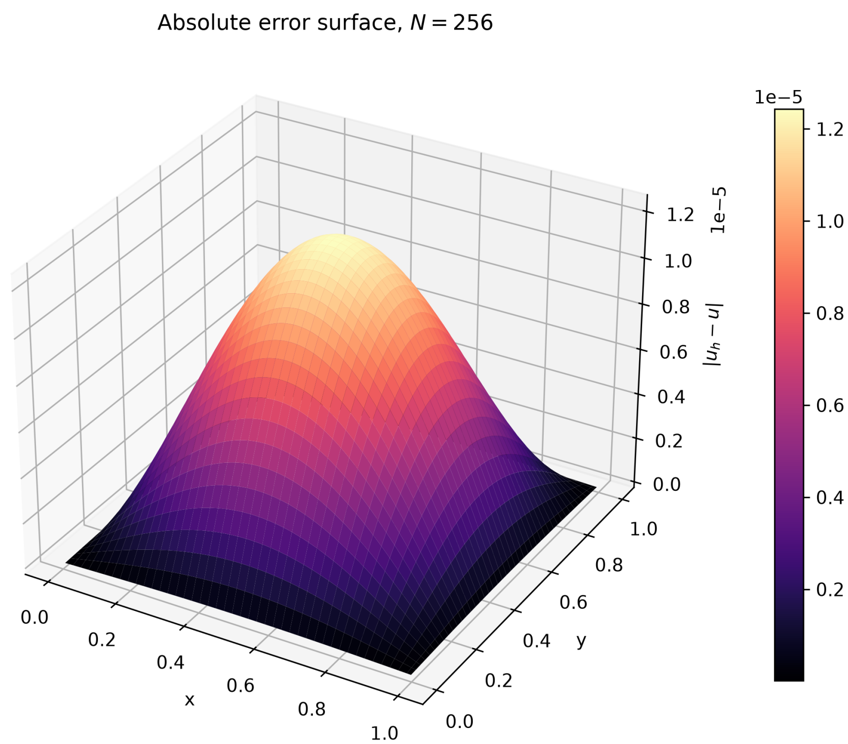

Figure 4: **Three-dimensional error surface for $N = 256$.** The absolute error $|u_h - u^\star|$ over the unit square forms a single smooth peak of height $\approx 1.245 \times 10^{-5}$ at the center $(0.5, 0.5)$, decaying to zero on the boundary. Its shape mirrors the solution's central antinode, where the fourth derivatives that drive the truncation error are largest.

4 Discussion

4.1 Confirmation of second-order accuracy

The observed convergence rates in Table 2 constitute a clean pass of the MMS verification test. The L^2 order is essentially exactly two across every grid pair, and any residual departure of the L^∞ order from two shrinks monotonically as $h \to 0$: $1.9924 \to 1.9980 \to 1.9995 \to 1.9999$. In Richardson-extrapolation terms, each halving of h divides the error by $2^r \approx 4$, and the small overshoot of the L^2 rate above two on the coarse grids (e.g. 2.0019 at N=32) is the expected imprint of the higher-order $\mathcal{O}(h^4)$ terms in the truncation expansion (9), which are relatively more visible when h is large and which vanish under refinement. This behavior — rates that approach the nominal order smoothly and from a consistent direction — is exactly the signature that verification practice regards as evidence of a correct implementation [15, 17, 19]. Had the code contained a defect that degrades the formal order — a boundary contribution handled at first order, a mis-scaled $1/h^2$ factor, or an off-by-one indexing error at the domain edge — the observed rates would have stalled near one, or drifted, rather than converging cleanly to two. The fact that they do not is a strong, quantitative statement of correctness for the discretization, the Kronecker-sum assembly, and the direct solve alike.

The agreement is reinforced by an independent consistency check across the two error norms. For a smooth, non-oscillatory error field the L^∞ and grid-normalized L^2 norms differ only by a fixed geometric factor determined by the error's spatial profile, and indeed the ratio E_∞/E_2 in Table 2 is essentially constant (approximately 1.98–2.00) across all five grids. A drifting ratio would have indicated a change in the error's shape with resolution — for example the emergence of a boundary layer — whereas the observed constancy confirms that the error retains a single self-similar profile as the mesh is refined, precisely as the truncation analysis requires.

4.2 Spatial structure of the error

Figures 3 and 4 explain *where* the discretization error lives, and the explanation is entirely analytic. The leading local truncation error (9) is proportional to $\partial_x^4 u^\star + \partial_y^4 u^\star$. For the manufactured solution (2) one computes

$$\partial_x^4 u^\star + \partial_y^4 u^\star = \pi^4 \sin(\pi x)\sin(\pi y) + \pi^4 \sin(\pi x)\sin(\pi y) = 2\pi^4 \, \sin(\pi x)\sin(\pi y) = 2\pi^4 \, u^\star(x,y). \quad (17)$$

The source that drives the pointwise error is thus proportional to the solution itself, and it is therefore largest exactly where $u^\star$ is largest — at the central antinode $(0.5, 0.5)$ — and it vanishes on the boundary. Because the error field is the response of the (smoothing, globally coupled) inverse discrete Laplacian to this centrally concentrated forcing, the accumulated global error is likewise a smooth, radially symmetric mound centered on the domain, exactly as observed. The magnitude is also consistent with theory: from (9) and (17) the leading truncation error at the center is approximately $\frac{h^2}{12} \cdot 2\pi^4$, which for $N = 256$ ($h \approx 3.891 \times 10^{-3}$) evaluates to $\approx 2.460 \times 10^{-5}$; the actual maximum global error, 1.245×10^{-5}, is of the same order and slightly smaller, reflecting the averaging effect of the discrete Green's function that maps local truncation error into global solution error. This quantitative accord between the predicted and measured error — in *location*, in *shape*, and in *order of magnitude* — closes the verification loop.

4.3 The role of the direct solver and the choice of norms

Two methodological choices deserve emphasis. First, solving each system with a sparse *direct* LU factorization [5, 6] rather than an iterative method means that the algebraic error is at the level of machine round-off, $\mathcal{O}(10^{-16})$ relative, which is many orders of magnitude below the discretization error being measured (as small as 6.000×10^{-6}). The reported errors are therefore pure discretization errors, uncontaminated by solver tolerance. Had we instead used an iterative solver, the stopping tolerance would have had to be pushed well below the finest-grid discretization error to avoid biasing the observed rate — a subtle but common pitfall in convergence studies that the direct solve sidesteps entirely. For the two-dimensional problems here, the sparse direct approach is comfortably efficient even at 65,536 unknowns; for substantially larger or three-dimensional problems, optimal-complexity multigrid [3, 10, 24] or FFT-based fast Poisson solvers would be the natural successors, and the same Kronecker structure (12) that we used for assembly is precisely what those fast solvers exploit. Second, reporting both a grid-normalized L^2 norm and an L^∞ norm provides complementary guarantees: the former certifies convergence in an averaged, energy-like sense that mirrors the continuous $L^2(\Omega)$ error, while the latter certifies that *no* individual node — including the worst-case central node — escapes the second-order bound. Their simultaneous convergence at order two is a stronger statement than either alone.

4.4 Scope and limitations

The conclusions of this study are deliberately scoped to a smooth solution on a smooth (rectangular) domain with homogeneous Dirichlet data, which is the regime in which the clean $\mathcal{O}(h^2)$ theory applies. Several practically important departures would change the picture and are natural extensions of this verification baseline. Solutions with reduced regularity — corner singularities on non-convex domains, or discontinuous coefficients — generally degrade the observed order below two unless local mesh refinement or specialized stencils are employed. Curved boundaries require either body-fitted grids, cut-cell/ghost-point boundary treatments whose consistency must itself be verified to preserve the interior order [1], or a switch to finite-element discretizations on unstructured meshes, whose approximation theory provides a complementary route to the same convergence guarantees [2]. Higher accuracy at fixed cost can be pursued with fourth-order compact stencils [1], which retain a nearest-neighbor footprint while raising the order to four; verifying such schemes with the same MMS procedure would show the log–log slope steepening from two to four. Finally, while the direct solver is ideal for this two-dimensional benchmark, extending the study to three dimensions or to time-dependent diffusion would motivate the iterative and multiscale solvers referenced above. In every one of these extensions, the MMS methodology employed here — manufacture a smooth exact solution, derive the matching source, and confirm the observed order against the theoretical one — remains the appropriate instrument of verification.

5 Conclusion

We have implemented, verified, and analyzed the standard second-order five-point finite-difference solver for the two-dimensional Poisson equation on the unit square with homogeneous Dirichlet boundary conditions. Using the Method of Manufactured Solutions with the smooth exact field $u^\star = \sin(\pi x)\sin(\pi y)$ and its analytic source $f = 2\pi^2 \sin(\pi x)\sin(\pi y)$, and assembling the discrete negative Laplacian as a sparse Kronecker sum solved by a direct LU factorization, we measured the discrete L^2 and L^∞ errors on five uniform grids with $N \in \{16, 32, 64, 128, 256\}$ interior nodes per dimension. The errors decrease by a factor of four under each mesh halving, and the pairwise observed orders of convergence rise monotonically to 2.0000 (L^2) and 1.9999 (L^∞) on the finest pair — an unambiguous confirmation of second-order accuracy and, by extension, of the correctness of the discretization, assembly, and solver. The spatial error field is a smooth, centrally peaked mound whose location, shape, and magnitude are quantitatively explained by the leading truncation term, which for this solution is proportional to the solution itself and hence maximal at the domain center. The complete error table (Table 2), the diagnostic figures, and the solver source are delivered with this report, and the $N = 256$ grid with 65,536 unknowns is included as required. Together they constitute a compact, fully reproducible verification of a canonical elliptic solver and a template for the MMS-based verification of its higher-order, higher-dimensional, and iteratively solved successors.

References

[1] Stéphane Abide. Finite difference preconditioning for compact scheme discretizations of the poisson equation with variable coefficients. *Journal of Computational and Applied Mathematics*, 379:112872, 2020. doi: 10.1016/j.cam.2020.112872.

[2] Dietrich Braess. *Finite Elements: Theory, Fast Solvers, and Applications in Solid Mechanics*. Cambridge University Press, Cambridge, UK, 3rd edition, 2007. ISBN 978-0-521-70518-9.

[3] William L. Briggs, Van Emden Henson, and Steve F. McCormick. *A Multigrid Tutorial*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2nd edition, 2000. ISBN 978-0-89871-462-3. doi: 10.1137/1.9780898719505.

[4] R. Courant, K. Friedrichs, and H. Lewy. Über die partiellen differenzengleichungen der mathematischen physik. *Mathematische Annalen*, 100(1):32–74, 1928. doi: 10.1007/BF01448839.

[5] Timothy A. Davis. Algorithm 832: UMFPACK v4.3—an unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software*, 30(2):196–199, 2004. doi: 10.1145/992200.992206.

[6] Timothy A. Davis. A column pre-ordering strategy for the unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software*, 30(2):165–195, 2004. doi: 10.1145/992200.992205.

[7] James W. Demmel. *Applied Numerical Linear Algebra*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997. ISBN 978-0-89871-389-3. doi: 10.1137/1.9781611971446.

[8] Lawrence C. Evans. *Partial Differential Equations*, volume 19 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2nd edition, 2010. ISBN 978-0-8218-4974-3.

[9] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, 4th edition, 2013. ISBN 978-1-4214-0794-4.

[10] Wolfgang Hackbusch. *Multi-Grid Methods and Applications*, volume 4 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, Heidelberg, 1985. ISBN 978-3-540-12761-1. doi: 10.1007/978-3-662-02427-0.

[11] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020. doi: 10.1038/s41586-020-2649-2.

[12] Arieh Iserles. *A First Course in the Numerical Analysis of Differential Equations*. Cambridge Texts in Applied Mathematics. Cambridge University Press, Cambridge, UK, 2nd edition, 2008. ISBN 978-0-521-73490-5.

[13] Randall J. LeVeque. *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2007. ISBN 978-0-89871-629-0. doi: 10.1137/1.9780898717839.

[14] K. W. Morton and D. F. Mayers. *Numerical Solution of Partial Differential Equations: An Introduction*. Cambridge University Press, Cambridge, UK, 2nd edition, 2005. ISBN 978-0-521-60793-3.

[15] William L. Oberkampf and Christopher J. Roy. *Verification and Validation in Scientific Computing*. Cambridge University Press, Cambridge, UK, 2010. ISBN 978-0-521-11360-1. doi: 10.1017/CBO9780511760396.

[16] Patrick J. Roache. *Verification and Validation in Computational Science and Engineering*. Hermosa Publishers, Albuquerque, NM, 1998. ISBN 978-0-913478-08-0.

[17] Patrick J. Roache. Code verification by the method of manufactured solutions. *Journal of Fluids Engineering*, 124(1):4–10, 2002. doi: 10.1115/1.1436082.

[18] Yousef Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2nd edition, 2003. ISBN 978-0-89871-534-7. doi: 10.1137/1.9780898718003.

[19] Kambiz Salari and Patrick Knupp. Code verification by the method of manufactured solutions. Technical Report SAND2000-1444, Sandia National Laboratories, Albuquerque, NM, 2000.

[20] John C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2nd edition, 2004. ISBN 978-0-89871-567-5. doi: 10.1137/1.9780898717938.

[21] Endre Süli and David F. Mayers. *An Introduction to Numerical Analysis*. Cambridge University Press, Cambridge, UK, 2003. ISBN 978-0-521-00794-8.

[22] Vidar Thomée. From finite differences to finite elements: A short history of numerical analysis of partial differential equations. *Journal of Computational and Applied Mathematics*, 128(1–2): 1–54, 2001. doi: 10.1016/S0377-0427(00)00507-0.

[23] Lloyd N. Trefethen and David Bau, III. *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997. ISBN 978-0-89871-361-9.

[24] Ulrich Trottenberg, Cornelis W. Oosterlee, and Anton Schüller. *Multigrid*. Academic Press, San Diego, CA, 2001. ISBN 978-0-12-701070-0.

[25] Charles F. Van Loan. The ubiquitous kronecker product. *Journal of Computational and Applied Mathematics*, 123(1–2):85–100, 2000. doi: 10.1016/S0377-0427(00)00393-9.

[26] Richard S. Varga. *Matrix Iterative Analysis*, volume 27 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, Heidelberg, 2nd edition, 2000. ISBN 978-3-540-66321-8. doi: 10.1007/978-3-642-05156-2.

[27] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, and Paul van Mulbregt. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272, 2020. doi: 10.1038/s41592-019-0686-2.