Scientific Computation with Python: Homework Solutions (HW1–HW8)

Jeongbin Jo[1, *]

[1]*Department of Physics, Yonsei University, Seoul, 03722, Republic of Korea*

(Dated: July 3, 2026)

Written solutions and numerical results for CSE5004 homework sets HW1–HW8: linear algebra and visualization, nonlinear root finding, interpolation, quadrature, Runge–Kutta integration, one-dimensional diffusion (Stokes second problem), two-dimensional heat equation (Crank–Nicolson), and iterative Poisson solvers.

CONTENTS

* jeongbin033@yonsei.ac.kr

I. INTRODUCTION

This report collects written solutions and figures for Python scientific-computing assignments HW1–HW8: NumPy linear algebra and plotting, nonlinear root finding, interpolation, quadrature, Runge–Kutta IVPs, one-dimensional diffusion (Stokes second problem), two-dimensional heat equation with Crank–Nicolson time stepping, and Jacobi/Gauss–Seidel/SOR Poisson solvers.

II. HOMEWORK SOLUTIONS

A. HW1: NumPy and Matplotlib

1. Problem 1 — matrix arithmetic

a. Setup. Consider the overdetermined linear system $Ax = b$ with $A \in \mathbb{R}^{5\times 3}$ and $b \in \mathbb{R}^5$:

$$A = \begin{bmatrix} 1 & 1 & -2 \\ 3 & 3 & -5 \\ 3 & 3 & -10 \\ 1 & 1 & -7 \\ -4 & -4 & 11 \end{bmatrix}, \qquad b = (2, 7, 2, -3, -4)^\top. \tag{1}$$

The task is to form the Gram matrix $A^\top A$, the projected right-hand side $A^\top b$, solve for x, and evaluate the Euclidean residual norm $\|Ax - b\|_2$.

b. Concepts. For $m > n$, an exact solution $Ax = b$ need not exist. The least-squares principle selects x that minimizes the sum of squared equation errors [1, 2]

$$\min_x \|Ax - b\|_2^2 = \min_x \sum_{i=1}^{m} (a_i^\top x - b_i)^2, \tag{2}$$

where $a_i^\top$ denotes the ith row of A. Setting the gradient of (2) to zero yields the *normal equations*

$$A^\top A\, x = A^\top b. \tag{3}$$

When $\operatorname{rank}(A) = n$ and the columns of A are linearly independent, $A^\top A$ is symmetric positive definite and (3) has a unique solution—the ordinary least-squares minimizer.

If columns of A are linearly dependent, $\operatorname{rank}(A) < n$ and $A^\top A$ is singular. Then (3) does not determine x uniquely: every vector in the affine subspace of least-squares minimizers satisfies the same residual $\|Ax - b\|_2$. In that case one adopts the *minimum-norm least-squares* solution

$$x^\star = \arg\min\{\|x\|_2 : x \text{ minimizes } \|Ax - b\|_2\}, \tag{4}$$

which is unique and can be obtained from the Moore–Penrose pseudoinverse $A^\dagger$ via $x^\star = A^\dagger b$ [1, 3]. The residual norm

$$\|r\|_2 = \|Ax - b\|_2 = \sqrt{\sum_{i=1}^{m} r_i^2}, \qquad r = Ax - b, \tag{5}$$

measures how far b lies from the column space $\mathcal{R}(A)$; it vanishes if and only if $b \in \mathcal{R}(A)$.

c. Rank structure. Columns $a^{(0)}$ and $a^{(1)}$ of A in (1) are identical ($a^{(0)} = a^{(1)}$), so $\operatorname{rank}(A) \le 2$. A numerical rank check gives $\operatorname{rank}(A) = 2$, confirming that $A^\top A \in \mathbb{R}^{3\times 3}$ is rank-deficient and singular.

d. Solution. Direct multiplication gives

$$A^\top A = \begin{bmatrix} 36 & 36 & -98 \\ 36 & 36 & -98 \\ -98 & -98 & 299 \end{bmatrix}, \qquad A^\top b = (42,\ 42,\ -82)^\top. \tag{6}$$

The first two rows of $A^\top A$ are equal, so (3) cannot be solved by a nonsingular triangular factorization. Applying minimum-norm least squares yields

$$x^\star \approx (1.949,\ 1.949,\ 1.003)^\top. \tag{7}$$

Note that $x_0^\star \approx x_1^\star$, consistent with the duplicate columns in A: components along the dependent directions are not uniquely fixed by the data alone, and the minimum-norm criterion selects a balanced pair.

Substituting into (5) and summing squared components gives

$$\|Ax^\star - b\|_2 = 0.7451, \tag{8}$$

so b is not in $\mathcal{R}(A)$ and the fit is genuinely approximate rather than exact.

2. Problem 2 — Ackley function

a. Definition. On $[-4, 4]^2$ with parameters $a = 20$, $b = 0.2$, $c = 2\pi$, the Ackley function [4] is

$$\begin{aligned} f(x, y) = {} & -a\, \mathrm{e}^{-b\sqrt{(x^2+y^2)/2}} \\ & - \exp\Big(\tfrac{1}{2}[\cos(cx) + \cos(cy)]\Big) + a + \mathrm{e}. \end{aligned} \tag{9}$$

It was introduced as a rugged, multimodal test landscape for connectionist optimization and remains a standard benchmark in global search [4].

b. Concepts. Equation (9) combines two qualitatively different terms. The radial factor $-a\,\mathrm{e}^{-b\sqrt{(x^2+y^2)/2}}$ creates a broad bowl centered at the origin: as $r = \sqrt{x^2 + y^2}$ increases, the exponential decays and f rises toward its asymptotic level $a + \mathrm{e} - 1$. The cosine term $-\exp(\frac{1}{2}[\cos(cx) + \cos(cy)])$ superimposes oscillatory structure with period $2\pi/c$ along each axis, producing many shallow local minima arranged on a grid-like pattern while preserving a single global minimum at $(0, 0)$ with $f(0, 0) = 0$.

Visualizing such a function uses *contour plots* (level sets $f(x, y) = c_k$ projected onto the xy-plane) and *surface plots* (height f above each grid point) [5]. A uniform tensor-product grid with spacing $\Delta = 0.05$ on $[-4, 4]^2$ provides 161×161 samples; the discrete minimum $\min_{i,j} f(x_i, y_j)$ approximates the continuous global minimum. *Slices* $y \mapsto f(x_0, y)$ at fixed x_0 reduce the problem to one dimension and reveal how sidewall ripples vary with lateral position. *Directional averages* $\bar{f}_x(x) = \frac{1}{N_y}\sum_j f(x, y_j)$ and analogously $\bar{f}_y(y)$ further smooth the landscape and highlight its large-scale bowl shape.

c. Solution. Figures 1 summarize the numerical exploration. On the stated grid, the discrete minimum is $f_{\min} \approx 4.4 \times 10^{-16}$ at $(x, y) \approx (0, 0)$, matching the analytic value $f(0, 0) = 0$ to within floating-point roundoff. Contours show the nearly circular basin near the origin surrounded by concentric ripples; the 3D surface emphasizes the deep central well and the plateau-like outer region. Slices at $x = -2, 0, 2$ display the periodic modulation along y, while the averaged profiles recover a smooth unimodal envelope along each coordinate direction.

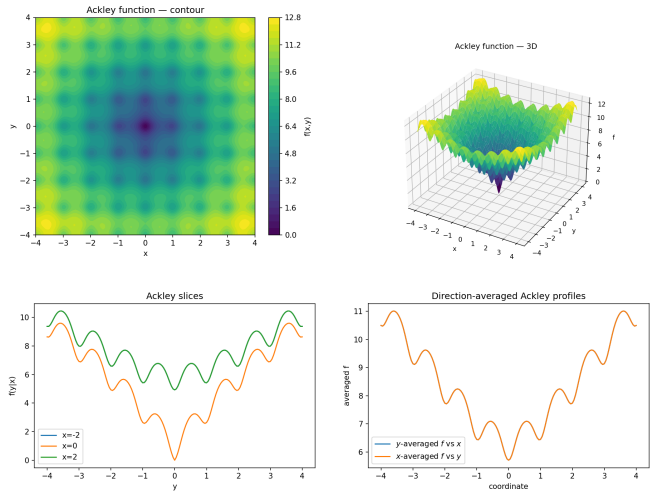

FIG. 1. Ackley function (9): contour and 3D surface (top row); fixed-x slices and coordinate-averaged profiles (bottom row).

B. HW2: Root finding

1. Problem statement

Find $x = (x, y, z)^\top \in \mathbb{R}^3$ such that the nonlinear system $F(x) = b$ holds, where

$$F(x) = \begin{bmatrix} x^2 + xyz + y^2z^3 \\ xy^2 - yz^2 - 2x^2 \\ x^2y + y^2z + z^4 \end{bmatrix}, \qquad b = \begin{bmatrix} 3 \\ 0 \\ 4 \end{bmatrix}. \tag{10}$$

Equivalently, define the residual $r(x) = F(x) - b$ and seek a root $r(x^\star) = 0$. The assignment asks for Newton iterations from $(1, 2, 3)$ and $(-1, 1, 1)$, analysis of the iterate starting at $(0, 0, 1)$, and comparison with Broyden's quasi-Newton method [6, 7].

2. Concepts

a. Newton–Raphson for systems. For a smooth map $F : \mathbb{R}^n \to \mathbb{R}^n$, Newton's method linearizes the defect at the current iterate $x^{(k)}$:

$$F(x^{(k)} + \delta x) \approx F(x^{(k)}) + J(x^{(k)})\, \delta x, \tag{11}$$

where the *Jacobian* $J_{ij} = \partial F_i / \partial x_j$. Requiring the linear model to satisfy $F(x^{(k)}) + J(x^{(k)})\delta x = b$ gives the correction equation

$$J(x^{(k)})\, \delta x = b - F(x^{(k)}) = -r(x^{(k)}), \qquad x^{(k+1)} = x^{(k)} + \delta x. \tag{12}$$

When $J(x^\star)$ is nonsingular and $x^{(0)}$ lies in a sufficiently small neighborhood of a simple root $x^\star$, the error satisfies $\|x^{(k+1)} - x^\star\| = O(\|x^{(k)} - x^\star\|^2)$: *quadratic convergence* [7, 8]. Each step therefore roughly doubles the number of correct digits once the iterate enters the basin of attraction.

b. Jacobian of (10). Writing $F = (F_1, F_2, F_3)^\top$ with components as in (10), the entries are

$$\begin{aligned} J_{1,:} &= \big(2x + yz,\ xz + 2yz^3,\ xy + 3y^2z^2\big), \\ J_{2,:} &= \big(y^2 - 4x,\ 2xy - z^2,\ -2yz\big), \\ J_{3,:} &= \big(2xy,\ x^2 + 2yz,\ y^2 + 4z^3\big). \end{aligned} \tag{13}$$

The Jacobian must be assembled and factorized (or solved by LU) at every Newton step [1]. If $\det J = 0$, the linear system (12) is underdetermined or inconsistent and the standard Newton step is not defined—a signal that the current point lies on a *singular manifold* of the iteration.

c. Multiple roots and initial guesses. Nonlinear systems often admit several roots; which one Newton finds depends on the initial guess through the *basin of attraction* of each root [7]. Two different starts can therefore converge to distinct $x^\star$ even though both satisfy (10) to tight tolerance.

d. Broyden's quasi-Newton method. Evaluating and factorizing J at every step can be costly. Broyden's method [6] replaces $J(x^{(k)})$ with a low-rank secant update B_k that satisfies the quasi-Newton condition $B_k \delta x^{(k-1)} = \delta r^{(k-1)}$ while staying close to B_{k-1}. The approximate Jacobian evolves along the iteration, reducing the number of explicit Jacobian evaluations at the price of superlinear rather than quadratic convergence near the root [7, 8].

3. Newton iterations

a. Convergence from two initial guesses. Starting at $(1, 2, 3)$, Newton updates (12) with the Jacobian (13) drive $\|r(x^{(k)})\|_2$ down to below 10^{-12} within 10–15 steps, with the typical doubling of accurate digits in the terminal phase. The converged root is

$$x^\star \approx (1.175,\ 1.675,\ 0.566)^\top. \tag{14}$$

From $(-1, 1, 1)$ the iteration follows a different descent path and settles at a second root,

$$x^\star \approx (0.430,\ 1.979,\ 0.815)^\top, \tag{15}$$

again with $\|F(x^\star) - b\|_2 < 10^{-12}$. Both vectors satisfy (10) numerically but differ in all components, illustrating distinct basins for the two starting points.

4. Singular initial guess $(0, 0, 1)$

Substituting $(x, y, z) = (0, 0, 1)$ into (13) gives

$$J(0, 0, 1) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 4 \end{bmatrix}, \qquad \det J = 0, \quad \operatorname{rank} J = 2. \tag{16}$$

The entire first column vanishes, so no unique Newton correction exists along the x-direction: the linearized model (11) carries no information about how x should change. Attempting to solve (12) therefore fails immediately. Geometrically, the iterate sits where the row space of J has dimension two, and an

entire line of corrections would produce the same first-order residual change. This underscores that Newton's method is reliable only when $J(x^{(k)})$ remains well conditioned; a poor or special initial guess can stall the iteration before any progress is made [7, 8].

5. Comparison with Broyden's method

Applying Broyden's secant update from the same two successful initial guesses reproduces both roots above to residuals $O(10^{-12})$. Agreement with full Newton confirms the correctness of the converged values. Broyden's iteration typically requires fewer explicit Jacobian constructions because B_k is carried forward and corrected by rank-one updates [6]; near the root the convergence rate is superlinear rather than quadratic, which is acceptable when Jacobian evaluations dominate the cost.

C. HW3: Numerical interpolation

1. Problem statement

On the square $[-1,1]^2$ with uniform spacing $h = 0.2$, construct tensor-product Lagrange interpolants for

$$f_1(x,y) = 0.26(x^2+y^2) - 0.48xy, \tag{17}$$
$$f_2(x,y) = \sin(\pi x)\sin(\pi y), \tag{18}$$

and repeat the interpolation of f_2 using Chebyshev nodes in each direction. Compare accuracy on a test grid and discuss the Lebesgue-type node product for uniform versus Chebyshev spacing [9, 10].

2. Concepts

a. One-dimensional Lagrange interpolation. Given distinct nodes $x_0, \ldots, x_n$ and values $f(x_i)$, the Lagrange interpolating polynomial of degree at most n is

$$p_n(x) = \sum_{i=0}^{n} f(x_i)\, L_i(x), \qquad L_i(x) = \prod_{\substack{m=0 \\ m\neq i}}^{n} \frac{x - x_m}{x_i - x_m}, \tag{19}$$

where $L_i(x_j) = \delta_{ij}$ [10, 11]. By construction, $p_n(x_i) = f(x_i)$ exactly at all nodes. The error for sufficiently smooth f is controlled by derivatives of f and by how rapidly the *node polynomial*

$$\omega_{n+1}(x) = \prod_{i=0}^{n} (x - x_i) \tag{20}$$

grows between nodes [9]. Large magnitudes of ω_{n+1} on the interpolation interval signal poor conditioning and, for uniform nodes on $[-1,1]$, can trigger the Runge phenomenon—large overshoot near endpoints when n increases for smooth but non-polynomial data [12].

b. Tensor-product interpolation in two dimensions. For nodes $\{x_i\}_{i=0}^{n_x}$ and $\{y_j\}_{j=0}^{n_y}$, the bivariate Lagrange interpolant is the tensor product

$$P(x,y) = \sum_{i=0}^{n_x} \sum_{j=0}^{n_y} f(x_i, y_j)\, L_i(x)\, L_j(y), \tag{21}$$

which matches f on the full Cartesian grid (x_i, y_j) [10]. With $h = 0.2$ on $[-1,1]$, each axis carries $n+1 = 11$ uniform nodes ($n = 10$). The total number of basis terms is $(n+1)^2 = 121$; the interpolant is a polynomial of degree at most n in each variable separately.

c. Chebyshev nodes. On $[-1,1]$, the degree-n Chebyshev nodes are

$$x_k = \cos\!\left(\frac{2k+1}{2n+2}\,\pi\right), \qquad k = 0, 1, \ldots, n. \tag{22}$$

They cluster toward ± 1 and are the zeros of the Chebyshev polynomial $T_{n+1}(x)$ [9, 13]. Among all choices of $n+1$ nodes on $[-1,1]$, Chebyshev points minimize $\max_{x\in[-1,1]} |\omega_{n+1}(x)|$ (equioscillation property), which suppresses the magnitude of (20) in the interior and improves interpolation of smooth but oscillatory functions compared with equally spaced nodes at the same degree [9, 12].

d. Error assessment. To quantify accuracy away from interpolation nodes, the root-mean-square (RMS) error on a test set $\{(x,y)\}$ is

$$\mathrm{RMS} = \sqrt{\frac{1}{N}\sum_{k=1}^{N} \big[f(x_k,y_k) - P(x_k,y_k)\big]^2}. \tag{23}$$

Here the test grid uses $x, y \in \{-0.95, -0.85, \ldots, 0.95\}$ with step 0.1, avoiding the interpolation nodes themselves so that reported errors reflect genuine approximation quality rather than exact nodal fit.

3. Results

Figure 2 (top row) shows contour plots of the interpolants. Table I lists RMS errors from (23).

case	RMS error
f_1, uniform	2.1×10^{-16}
f_2, uniform	1.7×10^{-5}
f_2, Chebyshev	4.4×10^{-6}

TABLE I. RMS interpolation error on the 0.1-spaced test grid.

a. f_1 — polynomial exactness. The function f_1 in (17) is a quadratic polynomial in x and y. Because the tensor-product space of degree-n polynomials in each variable with $n = 10$ contains all monomials up to x^2, y^2, and xy, the interpolant (21) reproduces f_1 exactly in exact arithmetic. The observed RMS error 2.1×10^{-16} is at the level of floating-point roundoff, confirming that uniform nodes suffice when the target function lies in the interpolation space.

b. f_2 — oscillatory function and node choice. The product $\sin(\pi x)\sin(\pi y)$ is smooth on $[-1,1]^2$ but is *not* a polynomial; it varies on the scale of the node spacing and excites high-degree terms in the Lagrange basis. With uniform nodes, $\mathrm{RMS} = 1.7 \times 10^{-5}$. Replacing both node sets by Chebyshev points (22) at the same $n = 10$ reduces the error to 4.4×10^{-6}, roughly a factor of four improvement without increasing polynomial degree.

c. Lebesgue-type node product. The bottom panel of Fig. 2 plots $\prod_i |x - x_i|$ on $[-1,1]$ with a linear vertical scale (step $\Delta x = 0.01$), as in the assignment hint [9]. For uniform nodes with $n = 10$, the product grows sharply near ± 1,

reflecting the endpoint error growth underlying the Runge phenomenon [12]. Chebyshev nodes equioscillate with peak magnitude below 10^{-3}, consistent with the reduced RMS error for f_2. This diagnostic connects node placement to interpolation conditioning through (20).

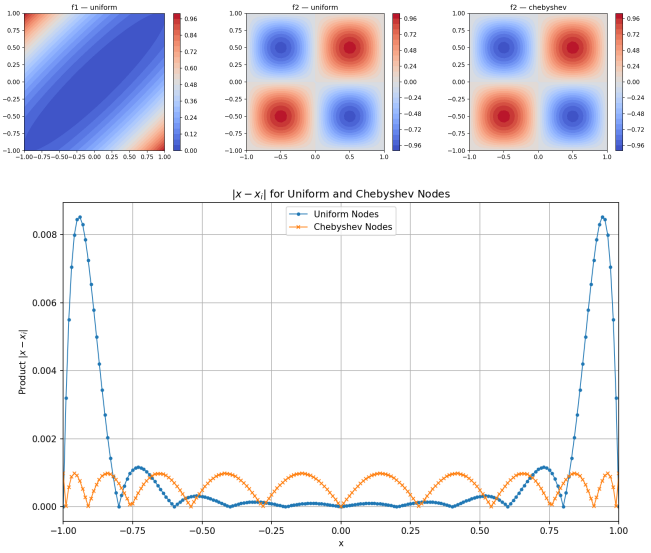

FIG. 2. Tensor-product Lagrange interpolants for f_1 and f_2 (top) and Lebesgue-type products $|\prod_i (x - x_i)|$ for uniform versus Chebyshev nodes with $n = 10$ (bottom).

D. HW4: Numerical integration

1. Problem statement

Approximate the definite integral

$$I = \int_0^{\pi} \sin x \, \mathrm{d}x = 2 \tag{24}$$

using composite Simpson quadrature, composite Gauss–Legendre quadrature (2–4 nodes per panel), natural cubic-spline quadrature on uniform panels, and Monte Carlo sampling. Compare absolute errors as the number of subintervals (or samples) increases and plot error versus resolution for the deterministic rules [10, 11, 14].

2. Concepts

a. Composite Simpson rule. Partition $[a, b]$ into n equal panels of width $h = (b - a)/n$ with nodes $x_j = a + jh$. Simpson's rule on each pair of panels uses three-point parabolic fitting; the composite formula with n even is [10, 11]

$$S_n = \frac{h}{3}\left[f(a) + f(b) + 4 \sum_{\substack{j=1 \\ j \text{ odd}}}^{n-1} f(x_j) + 2 \sum_{\substack{j=2 \\ j \text{ even}}}^{n-2} f(x_j) \right]. \tag{25}$$

For $f \in C^4[a, b]$, the global error satisfies $|I - S_n| = O(h^4)$, so halving h reduces the error by roughly a factor of 16 in the asymptotic regime [10]. Because (24) has $f(x) = \sin x \in C^{\infty}$, Simpson is an appropriate high-order choice.

b. Gauss–Legendre quadrature. On $[-1, 1]$, the m-point Gauss–Legendre rule approximates

$$\int_{-1}^{1} g(t) \, \mathrm{d}t \approx \sum_{i=1}^{m} w_i \, g(\xi_i), \tag{26}$$

where the nodes ξ_i are roots of the degree-m Legendre polynomial and the weights w_i are chosen so that (26) is exact for all polynomials of degree up to $2m - 1$ [14, 15]. This is optimal among m-point rules for polynomial exactness. On a composite mesh with n panels, each panel $[x_k, x_{k+1}]$ is mapped affinely to $[-1, 1]$ and (26) is applied; the total error decays rapidly when f is smooth because high-degree terms in the local Taylor expansion are integrated exactly on each panel.

c. Natural cubic-spline quadrature. Given nodal values $f(x_j)$ on a uniform grid, the natural cubic spline $s(x)$ satisfies $s(x_j) = f(x_j)$, is twice continuously differentiable, and has natural end conditions $s''(a) = s''(b) = 0$ [10, 11]. On each panel $[x_j, x_{j+1}]$ the spline is a cubic polynomial and can be integrated analytically:

$$\int_{x_j}^{x_{j+1}} s(x) \, \mathrm{d}x = h \, f_j + \frac{h^2}{2} b_j + \frac{h^3}{3} c_j + \frac{h^4}{4} d_j, \tag{27}$$

where (b_j, c_j, d_j) are the local spline coefficients. Summing over panels yields a quadrature rule consistent with interpolating f by a spline rather than a single global polynomial. For smooth f, the error is typically $O(h^4)$, similar in order to Simpson, but the spline construction distributes curvature information globally through the tridiagonal system for second derivatives.

d. Monte Carlo integration. For uniform samples $X_i \sim \mathcal{U}(a, b)$,

$$I \approx \widehat{I}_N = \frac{b - a}{N} \sum_{i=1}^{N} f(X_i). \tag{28}$$

By the law of large numbers, $\widehat{I}_N \to I$ as $N \to \infty$ [16]. If $\mathrm{Var}(f(X)) = \sigma^2$, the standard error scales as $\sigma/\sqrt{N}$, so the root-mean-square error decays as $O(N^{-1/2})$ regardless of dimension—the principal advantage of Monte Carlo in high-dimensional integrals [14, 16]. For the one-dimensional integral (24), deterministic rules achieve much faster convergence, but Monte Carlo illustrates stochastic error scaling.

3. Results

a. Simpson and Gauss–Legendre. Table II summarizes absolute errors $|I - Q_n|$ for composite Simpson and composite Gauss–Legendre with four nodes per panel. Simpson exhibits clear fourth-order decay: the error falls from 4.6×10^{-3} at $n = 4$ to 1.0×10^{-6} at $n = 32$, consistent with (25). Gauss–Legendre with $m = 4$ reaches machine precision ($\sim 10^{-15}$) by $n = 32$ because $\sin x$ is smooth and the rule is exact for polynomials up to degree seven on each panel [14]. Figure 3 shows log–log slopes: Simpson approaches slope 4; composite GL4 drops steeply once the discretization error falls below roundoff.

b. Cubic-spline quadrature. Natural spline integration via (27) gives errors 1.3×10^{-3} ($n = 4$), 7.0×10^{-5} ($n = 8$), 4.2×10^{-6} ($n = 16$), and 2.6×10^{-7} ($n = 32$). The decay is fourth-order in h, comparable to Simpson at the same node count, reflecting that both methods rely on local cubic polynomial approximations of smooth data [10].

n	Simpson	GL ($m=2$)	GL ($m=4$)
4	4.6×10^{-3}	1.8×10^{-4}	1.7×10^{-10}
8	2.7×10^{-4}	1.1×10^{-5}	6.4×10^{-13}
16	1.7×10^{-5}	6.9×10^{-7}	2.0×10^{-15}
32	1.0×10^{-6}	4.3×10^{-8}	8.9×10^{-16}

TABLE II. Absolute error for (24) with composite Simpson and Gauss–Legendre (m nodes per panel).

c. Monte Carlo. With $N = 10^3$, 10^4, and 10^5 uniform samples, absolute errors are 2.2×10^{-2}, 3.5×10^{-3}, and 1.3×10^{-3} respectively. Successive tenfold increases in N reduce the error by factors of roughly 6 and 3, bracketing the theoretical $\sqrt{10}\approx3.2$ improvement expected from (28) when sampling variance dominates [16]. At equal computational cost for this one-dimensional integral, Simpson or Gauss–Legendre are far more accurate than Monte Carlo.

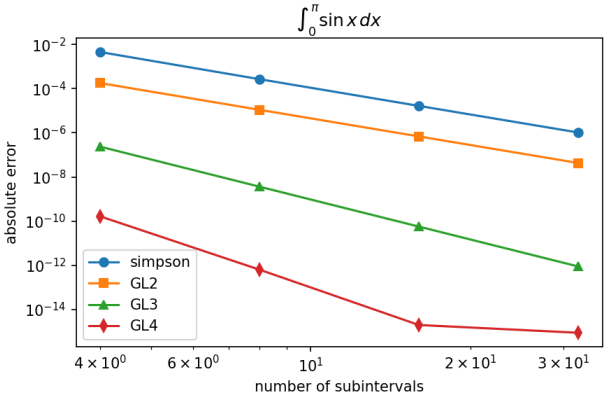

FIG. 3. Log–log absolute error versus number of subintervals n for composite Simpson and composite Gauss–Legendre rules applied to (24).

E. HW5: Runge–Kutta methods and baseball dynamics

1. Problem statement

Part 1. Integrate the scalar initial-value problem

$$x'(t) = t + 2tx, \qquad x(0) = 0, \tag{29}$$

on $t\in[0,2]$ with classical Runge–Kutta methods of order two (RK2) and four (RK4), compare with the exact solution, and study the effect of step size [15, 17, 18].

Part 2. Simulate four baseball pitches—fastball, curveball, slider, and screwball—by integrating the six-component equations of motion with aerodynamic drag and Magnus (spin) forces until the ball reaches the plate at $x = 18.39\,\mathrm{m}$ [19, 20].

2. Concepts: initial-value problems and Runge–Kutta methods

a. First-order IVP. The scalar problem $x' = f(t,x)$ with $x(t_0) = x_0$ defines a first-order initial-value problem (IVP), covering both autonomous and non-autonomous ODEs. Numerical methods advance the solution on a mesh $t_n = t_0 + nh$ by constructing approximations $x_n \approx x(t_n)$ from information at previous steps [15, 17]. One-step methods $x_{n+1} = x_n + h\Phi(t_n, x_n, h)$ have *order of consistency* p when the local truncation error satisfies $\tau_n = O(h^p)$ for smooth solutions.

b. Classical RK2 (Heun). The two-stage explicit Runge–Kutta method used here is

$$k_1 = f(t_n, x_n), \tag{30}$$
$$k_2 = f(t_n + h,\, x_n + hk_1), \tag{31}$$
$$x_{n+1} = x_n + \tfrac{h}{2}(k_1 + k_2). \tag{32}$$

This is second-order accurate ($p = 2$): halving h reduces the global error by approximately a factor of four in the asymptotic regime [17, 18].

c. Classical RK4. The four-stage method is

$$\begin{aligned} k_1 &= f(t_n, x_n),\\ k_2 &= f\Big(t_n + \tfrac{h}{2},\, x_n + \tfrac{h}{2}k_1\Big),\\ k_3 &= f\Big(t_n + \tfrac{h}{2},\, x_n + \tfrac{h}{2}k_2\Big),\\ k_4 &= f(t_n + h,\, x_n + hk_3),\\ x_{n+1} &= x_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4). \end{aligned} \tag{33}$$

It has order $p = 4$ and is the workhorse for smooth non-stiff IVPs [17, 18]. The Butcher tableau encodes the stage weights; RK4 achieves fourth order by balancing contributions from intermediate slopes at t_n, $t_n + h/2$, and $t_n + h$.

d. Exact solution of (29). Separating variables or recognizing an integrating factor gives

$$x(t) = \frac{1}{2}\big(\mathrm{e}^{t^2} - 1\big), \tag{34}$$

which satisfies $x(0) = 0$ and $x'(t) = t + 2tx(t) = t\,\mathrm{e}^{t^2}$. This closed form provides a reference for measuring global error at any t.

3. Results: scalar IVP

Table III lists $|x_{\mathrm{num}}(2) - x_{\mathrm{exact}}(2)|$ with $h = 0.01$. RK2 (Heun) reaches error 1.42×10^{-2}; RK4 reaches 7.56×10^{-7}, a gain of more than four orders of magnitude at the same step count because the problem is smooth and RK4's $O(h^4)$ global error dominates for small h [17].

method	$\lvert x_{\mathrm{num}}(2) - x_{\mathrm{exact}}(2)\rvert$
RK2 ($h = 0.01$)	1.42×10^{-2}
RK4 ($h = 0.01$)	7.56×10^{-7}

TABLE III. Global error at $t = 2$ for (29).

A step-size study with RK4 gives errors 7.56×10^{-7} ($h = 0.01$), 4.24×10^{-4} ($h = 0.05$), and 5.92×10^{-3} ($h = 0.1$). Halving h from 0.1 to 0.05 reduces the error by ≈ 14, approaching the factor $2^4 = 16$ expected for fourth-order methods [18]. Figure 4 (left) overlays the numerical trajectories with (34).

4. Concepts: baseball flight mechanics

a. State vector and forces. Let (x, y, z) be position and (v_x, v_y, v_z) velocity in meters and seconds. With spin axis angle ϕ in the horizontal plane, gravity $g = 9.81\,\mathrm{m/s^2}$, angular

speed $\Omega = 1800\,\mathrm{rpm}$, and Magnus coefficient $B = 4.1\times10^{-4}$, the assignment model is

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x \\ y \\ z \\ v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ -F_v v_x + B\Omega(v_z \sin\phi - v_y \cos\phi) \\ -F_v v_y + B\Omega v_x \cos\phi \\ -g - F_v v_z - B\Omega v_x \sin\phi \end{bmatrix}, \tag{35}$$

where $V = \|\mathbf{v}\|_2$ and the drag-related factor $F_v(V) = f(V)\,V$ uses

$$f(V) = 0.0039 + \frac{0.0058}{1 + \exp\big((V-35)/5\big)}. \tag{36}$$

The term $-F_v v_x$ (and analogously for y, z) represents speed-dependent aerodynamic drag opposing motion [19, 20]. The cross-product structure of the Magnus terms—proportional to Ω and linear in velocity components—models the lateral and vertical lift/deflection caused by spin [19]. Initial release height is $z(0) = 1.7\,\mathrm{m}$ with launch angle $\theta = 1^\circ$ above horizontal and speed v_0.

b. Integration of the system. Equation (35) is marched with RK4 using $\Delta t = 0.002\,\mathrm{s}$ until the ball reaches $x = 18.39\,\mathrm{m}$. This step is small enough that discretization error is negligible relative to model uncertainty; component-wise RK4 preserves the same order properties as in the scalar case [17].

5. Results: baseball trajectories

Four pitch types with distinct spin orientations ϕ produce markedly different vertical profiles at the plate (Figure 4, right):

pitch	v_0 (m/s)	ϕ (deg)	plate height z (m)
fastball	40	225	1.14
curveball	30	45	−0.37
slider	30	0	−0.03
screwball	30	135	−0.37

TABLE IV. Vertical position at $x = 18.39\,\mathrm{m}$ for $\theta = 1^\circ$, release height $1.7\,\mathrm{m}$.

The fastball retains the highest plate height ($1.14\,\mathrm{m}$) owing to its greater initial speed and reduced flight time under gravity. Curveball and screwball drop to $z \approx -0.37\,\mathrm{m}$, well below the release point, as Magnus and drag combine to enhance downward break over the $18.39\,\mathrm{m}$ travel. The slider stays near $z \approx -0.03\,\mathrm{m}$, a smaller vertical deviation consistent with its spin axis. Figure 4 (right) shows the full (x, y, z) paths: lateral break from the Magnus terms separates pitches that overlap in the x–z projection [19, 20].

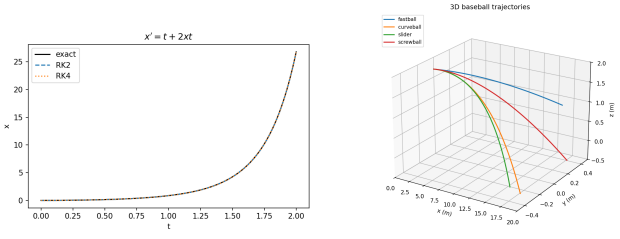

FIG. 4. Scalar RK2/RK4 versus (34) (left) and three-dimensional baseball trajectories (x, y, z) (right).

F. HW6: Stokes second problem

1. Problem statement

Part 1 (analytic). Derive the velocity field for unidirectional flow $u = u(y, t)$ driven by an oscillating wall and governed by viscous diffusion [21, 22].

Part 2 (numerical). Solve the same physics on a finite slab $y \in [0, L]$ with $\nu = 1$, forcing frequency $n = 2$, $U_0 = 1$, bottom boundary $u(0, t) = \cos(2t)$, and top boundary $u(L, t) = 0$. Compare explicit forward-time centered-space (FTCS) and implicit Crank–Nicolson (CN) time marching for $L = 10$ and $L = 2$, and assess temporal convergence [23–25].

2. Concepts: Stokes second problem

a. Governing equation. For parallel flow with zero pressure gradient, the incompressible Navier–Stokes equations reduce to the diffusion (heat) equation [21]

$$\frac{\partial u}{\partial t} = \nu\,\frac{\partial^2 u}{\partial y^2}, \tag{37}$$

where ν is kinematic viscosity. This is *Stokes' second problem*: an oscillating flat plate at $y = 0$ sets up a unidirectional shear wave that diffuses and decays into the fluid [22].

b. Boundary conditions and semi-infinite solution. On a semi-infinite domain $y \geq 0$,

$$u(0, t) = U_0 \cos(nt), \qquad u(\infty, t) = 0 \text{ (bounded)}. \tag{38}$$

Seeking a harmonic response $u = \Re\{U_0\,\mathrm{e}^{\mathrm{i}(nt - ky)}\}$ and substituting into (37) gives the dispersion relation $\mathrm{i}n = \nu k^2$, hence

$$k = \sqrt{\frac{n}{2\nu}}\,(1 + \mathrm{i}). \tag{39}$$

The real-valued solution is

$$u(y, t) = U_0\,\mathrm{e}^{-\eta_s}\cos\big(nt - \eta_s\big), \qquad \eta_s = \sqrt{\frac{n}{2\nu}}\,y. \tag{40}$$

The *penetration depth* $\delta = \sqrt{2\nu/n}$ equals unity for $(\nu, n) = (1, 2)$, so a top wall at $L = 2$ lies inside the shear layer whereas $L = 10$ is effectively semi-infinite.

3. Concepts: finite-difference discretization

a. Spatial mesh and boundary treatment. The slab uses $N_y = 50$ nodes with $\Delta y = L/(N_y - 1)$, $y_i = i\Delta y$, and $u_i^j \approx u(y_i, t_j)$. Dirichlet data are enforced as $u_0^j = \cos(nt_j)$ and $u_{N_y-1}^j = 0$.

b. FTCS scheme. Forward-time, centered-space differencing of (37) gives

$$u_i^{j+1} = u_i^j + r\big(u_{i+1}^j - 2u_i^j + u_{i-1}^j\big), \qquad r = \frac{\nu\,\Delta t}{\Delta y^2}, \tag{41}$$

for interior nodes [23, 24]. Von Neumann stability requires $r \leq \frac{1}{2}$; for $L = 10$ we use $\Delta t = 0.005$ ($r \approx 0.12$), and for $L = 2$ we use $\Delta t = 10^{-4}$ to maintain stability on the finer Δy.

c. Crank–Nicolson scheme. CN averages explicit and implicit diffusion contributions:

$$\frac{u_i^{j+1} - u_i^j}{\Delta t} = \frac{\nu}{2}\left[\left.\frac{\partial^2 u^{j+1}}{\partial y^2}\right|_i + \left.\frac{\partial^2 u^j}{\partial y^2}\right|_i\right], \tag{42}$$

yielding a tridiagonal system with $r = \nu\Delta t/(2\Delta y^2)$ each step [23, 25]. The same $(\Delta y, \Delta t)$ pairs as FTCS are used so that both schemes can be compared on identical grids.

d. Snapshot times. With $T = 10\pi$ and $nt = nt$, transient profiles are taken at $nt = 0, \pi/2, \pi, 3\pi/2, 2\pi$ (i.e. $t = 0, \pi/(2n), \ldots, \pi/n$). Quasi-steady profiles use $(nt - T) = 0, \pi/2, \ldots, 2\pi$, i.e. $t = (T + \phi)/n$ for ϕ in the same set. Velocity is plotted as $u(y)$ with y on the horizontal axis.

4. Results: velocity profiles

Figure 5 shows the assignment-style 2×2 layout: FTCS and CN in the top row during startup ($t < T$), and the corresponding quasi-steady profiles ($t \geq T$) in the bottom row. For $L = 10$, both schemes reproduce the decaying oscillatory structure expected from (40). Figure 6 repeats the layout for $L = 2$; the stationary top wall reflects diffusion and visibly distorts the profiles relative to the semi-infinite reference.

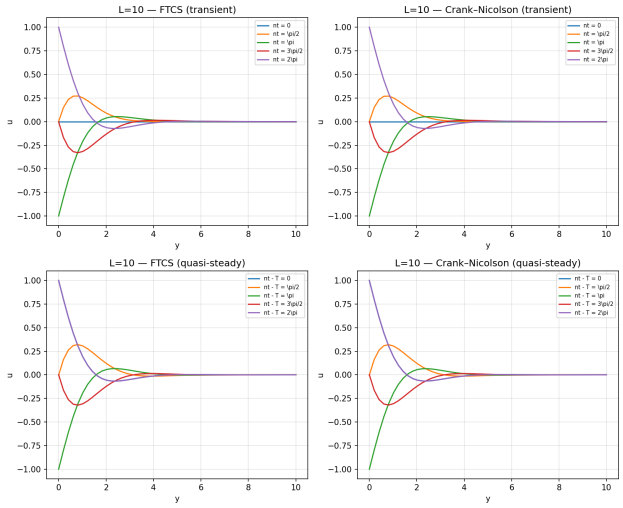

FIG. 5. Velocity profiles for $L = 10$, $N_y = 50$, $\Delta t = 0.005$: FTCS (41) (left column) and Crank–Nicolson (42) (right column). Top row: transient ($nt = 0, \pi/2, \ldots$); bottom row: quasi-steady ($(nt - T) = 0, \pi/2, \ldots$).

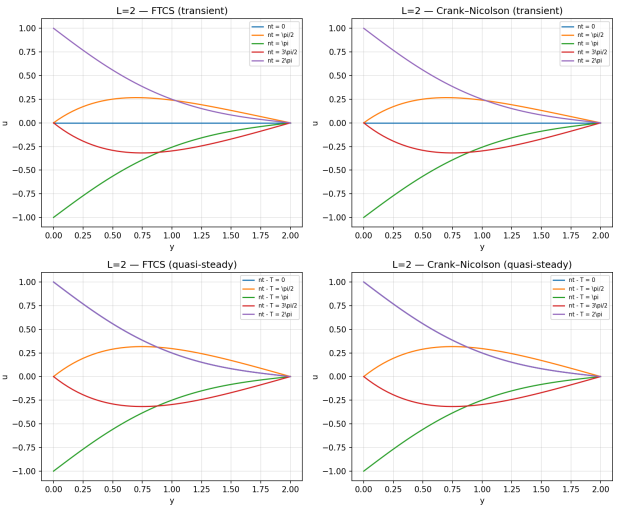

FIG. 6. Same layout for $L = 2$ with $\Delta t = 10^{-4}$; gap distance modifies the shear layer.

a. Animation. Figure 7 embeds an animation of the first forcing period ($nt \in [0, 2\pi]$) for both gap sizes and both time schemes. PNG frames are included for PDF viewers that support the `animate` package; the same sequence is exported as `stokes_profiles.gif` for standalone playback.

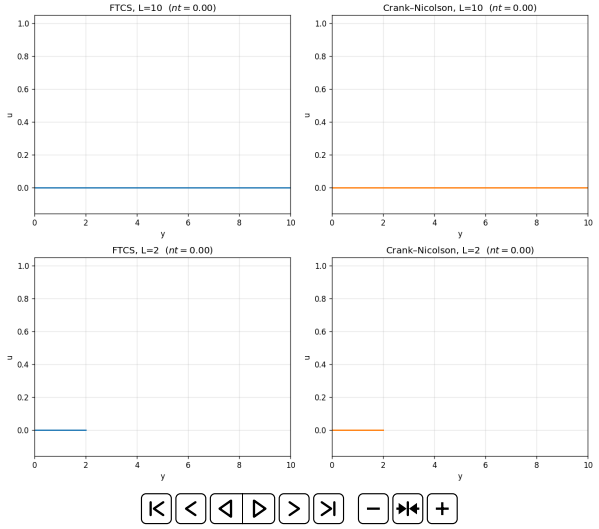

FIG. 7. Animated velocity profiles during the first period $nt \in [0, 2\pi]$: FTCS and CN for $L = 10$ (top row) and $L = 2$ (bottom row). Use the PDF viewer controls to play; alternatively open `stokes_profiles.gif`.

5. Results: temporal convergence

At $L = 10$ with $N_y = 50$ fixed, relative ℓ_2 error against (40) at quasi-steady phase $(nt - T) = \pi/2$ is plotted versus Δt in Fig. 8. FTCS errors follow $O(\Delta t)$ scaling; CN errors follow $O(\Delta t^2)$ once Δt is small enough [24, 25].

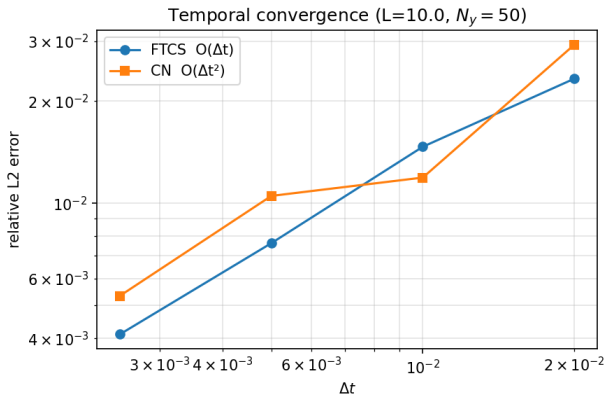

FIG. 8. Relative ℓ_2 error versus Δt for FTCS (41) and CN (42) at $L = 10$, quasi-steady snapshot $(nt - T) = \pi/2$.

G. HW7: Two-dimensional heat equation and Crank–Nicolson

1. Problem statement

On $\Omega = [-1, 1]^2$ with $\alpha = 1$, solve the parabolic problem

$$\frac{\partial \phi}{\partial t} = \alpha \nabla^2 \phi + S(x, y), \qquad \phi|_{\partial\Omega} = 0, \tag{43}$$

using Crank–Nicolson (CN) time marching. Each implicit step is solved by successive over-relaxation (SOR) [23, 25, 26].

Part 1. For the steady source $S = 2(2 - x^2 - y^2)$, show that $\phi = (1 - x^2)(1 - y^2)$ satisfies the steady limit of (43).

Part 2. Time-march (43) to steady state with CN; plot numerical, exact, and error fields for several $(N, \Delta t)$ pairs.

Part 3. Measure spatial and temporal orders of accuracy by log–log studies of the interior RMS error [11, 23, 24].

2. Concepts: steady solution and CN discretization

a. Steady manufactured solution. At steady state, $\partial_t \phi = 0$ and (43) becomes $-\nabla^2 \phi = S$. For $\phi = (1 - x^2)(1 - y^2)$,

$$\nabla^2 \phi = \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = -2(1-y^2) - 2(1-x^2) = -2(2-x^2-y^2), \tag{44}$$

so $S = 2(2 - x^2 - y^2)$ is the required source. Boundary values vanish because $\phi = 0$ on $x = \pm 1$ and $y = \pm 1$ [23].

b. Five-point Laplacian and CN scheme. On a uniform $(N + 1) \times (N + 1)$ grid with $h = 2/N$, the centered Laplacian L_h at interior nodes (i, j) reads

$$(L_h \phi)_{i,j} = \frac{\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1} - 4\phi_{i,j}}{h^2}. \tag{45}$$

CN averages explicit and implicit diffusion contributions [23, 25]:

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} = \frac{\alpha}{2}\left(L_h \phi^{n+1} + L_h \phi^n\right) + S, \tag{46}$$

equivalently,

$$\left(I - \tfrac{\alpha \Delta t}{2} L_h\right)\phi^{n+1} = \left(I + \tfrac{\alpha \Delta t}{2} L_h\right)\phi^n + \Delta t\, S. \tag{47}$$

For pure diffusion the scheme is unconditionally stable in the von Neumann sense; accuracy is $O(h^2 + \Delta t^2)$ [23, 24]. Each CN step requires solving an elliptic system; here SOR with $\omega = 1.7$ inverts $\left(I - \frac{\alpha \Delta t}{2} L_h\right)$ iteratively [26, 27].

c. Convergence studies. Steady-state runs terminate when $\max |\phi^{n+1} - \phi^n| < 10^{-10}$. Spatial accuracy is assessed by comparing the steady CN solution with ϕ_{exact} at fixed $\Delta t = 0.01$ (Part 2). Temporal order at $t = 0.5$ is measured by a log–log study against a reference solution with $\Delta t = 5 \times 10^{-4}$ on $N = 32$ [11].

3. Results: Part 1 — steady exact solution

Equation (44) confirms that $\phi = (1-x^2)(1-y^2)$ is the steady solution of (43) for the given S and homogeneous Dirichlet boundaries. This closed form is used below to quantify discretization error.

4. Results: Part 2 — CN steady-state fields

Table V lists the time steps required to reach steady state and the interior RMS error. For $(N, \Delta t) = (32, 0.01)$ and $(64, 0.01)$, CN converges in 408 steps with $L_2 \sim 10^{-9}$; halving Δt at $N = 64$ increases the step count to 787 while maintaining $L_2 \sim 10^{-9}$. Figure 9 shows the numerical field, exact bowl, and pointwise error at $N = 32$; the error is largest near the center and remains below 10^{-8} in magnitude.

N	Δt	steps	L_2 error
32	0.01	408	1.0×10^{-9}
64	0.01	408	9.8×10^{-10}
64	0.005	787	2.0×10^{-9}

TABLE V. CN steady-state performance for selected grids.

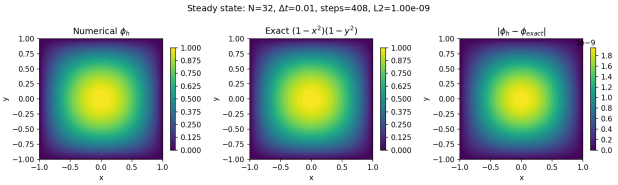

FIG. 9. Numerical solution, exact $(1 - x^2)(1 - y^2)$, and error at $N = 32$, $\Delta t = 0.01$.

5. Results: Part 3 — order of accuracy

a. Temporal order. At fixed $N = 32$ and $t = 0.5$, the RMS error against a reference solution ($\Delta t = 5 \times 10^{-4}$) decays as Δt^2 with fitted slope 1.99 (Table VI, Fig. 10). Halving Δt reduces the error by roughly a factor of four, confirming the expected Crank–Nicolson temporal order [23, 25].

b. Spatial accuracy. The five-point Laplacian (45) is second-order accurate in h [23, 24]. Part 2 confirms this numerically at steady state: with $\Delta t = 0.01$ and tolerance 10^{-10}, CN reaches $L_2 = 1.0 \times 10^{-9}$ at $N = 32$ and $L_2 = 9.8 \times 10^{-10}$ at $N = 64$ (Table V), i.e. agreement with ϕ_{exact} to near the iterative/time-step floor. Refining N from 32 to 64 at fixed Δt further reduces the error by roughly 2%, consistent with both grids already operating near the accuracy limit set by $\Delta t = 0.01$.

Δt	L_2 at $t = 0.5$ (vs. ref.)
0.04	3.6×10^{-4}
0.02	9.3×10^{-5}
0.01	2.3×10^{-5}
0.005	5.8×10^{-6}

TABLE VI. Temporal convergence at $N = 32$ (reference $\Delta t = 5 \times 10^{-4}$; fitted slope 1.99).

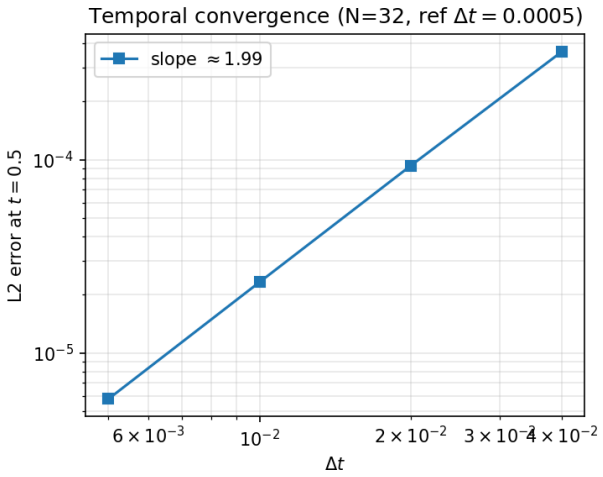

FIG. 10. Temporal error at $t = 0.5$ versus Δt on $N = 32$ (reference $\Delta t = 5 \times 10^{-4}$; fitted slope ≈ 1.99).

H. HW8: Poisson equation and iterative solvers

1. Problem statement

On the unit square $\Omega = [0,1]^2$, solve the Poisson problem

$$\nabla^2 u = f(x,y), \qquad u|_{\partial\Omega} = 0, \tag{48}$$

with

$$f_1(x,y) = \sin(\pi x)\,\sin(\pi y), \tag{49}$$

$$f_2(x,y) = \exp\!\left[-100\left((x-\tfrac{1}{2})^2 + (y-\tfrac{1}{2})^2\right)\right], \tag{50}$$

using Jacobi, Gauss–Seidel, and successive over-relaxation (SOR) iterations.

Part 1. Apply the three methods to (49), compare discrete residual norms, discretization error against the closed-form solution, and wall-clock time. Repeat the comparison for several grid sizes N (intervals per axis).

Part 2. Using SOR, compute u_1 for f_1, u_2 for f_2, and u for $f_1 + f_2$. Verify linearity $u \approx u_1 + u_2$ and compare the three fields [23, 26].

2. Concepts: discrete Poisson problem

a. Five-point stencil. Equation (48) is an elliptic boundary-value problem: information propagates instantaneously across the domain, and Dirichlet data on $\partial\Omega$ constrain the entire interior [15, 23, 24]. On a uniform grid with spacing $\Delta x = \Delta y = h = 1/N$, the standard second-order centered Laplacian gives, at interior node (i,j),

$$\frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}}{h^2} = f_{i,j}. \tag{51}$$

Rearranging (51) yields the Jacobi update

$$u_{i,j}^{(k+1)} = \tfrac{1}{4}\left(u_{i+1,j}^{(k)} + u_{i-1,j}^{(k)} + u_{i,j+1}^{(k)} + u_{i,j-1}^{(k)} - h^2 f_{i,j}\right), \tag{52}$$

which treats all neighbors at the old iterate [8, 11, 26].

b. Manufactured solution for f_1. Because $f_1 = \sin(\pi x)\sin(\pi y)$ is an eigenfunction of the Laplacian on $[0,1]^2$ with homogeneous Dirichlet data, the continuous solution follows by separation of variables [23]. Seeking $u = A\sin(\pi x)\sin(\pi y)$ and substituting into (48) gives $-2\pi^2 A \sin(\pi x)\sin(\pi y) = \sin(\pi x)\sin(\pi y)$, hence

$$u_{\text{exact}}(x,y) = -\frac{\sin(\pi x)\,\sin(\pi y)}{2\pi^2}. \tag{53}$$

The interior RMS error $\|u - u_{\text{exact}}\|_2$ quantifies spatial discretization accuracy of (51). For f_2, no closed form is used; the localized Gaussian source produces a narrow interior dip in u_2 centered at $(\frac{1}{2}, \frac{1}{2})$.

3. Concepts: Jacobi, Gauss–Seidel, and SOR

a. Fixed-point form and convergence rates. Writing (51) as $Au = b$ with A the sparse five-point Laplacian, Jacobi iteration has the form $u^{(k+1)} = Ru + c$, where convergence is governed by the spectral radius $\rho(R)$ [1, 26]. For the model Poisson problem on a square, the lowest Laplacian eigenmode $\sin(\pi x)\sin(\pi y)$ yields the slowest Jacobi decay with amplification factor $\cos(\pi h) \approx 1 - \frac{1}{2}\pi^2 h^2$, explaining the rapid growth of iteration counts as N increases [23, 27]. Gauss–Seidel replaces $u^{(k)}$ by the most recently updated values within the same sweep, typically halving the iteration count of Jacobi for diagonally dominant systems such as (51) [1, 26].

b. Successive over-relaxation. SOR accelerates the Gauss–Seidel correction:

$$u_{i,j}^{(k+1)} = (1-\omega)\,u_{i,j}^{(k)} + \omega\,u_{i,j}^{\text{GS}}, \tag{54}$$

where $u_{i,j}^{\text{GS}}$ is the Gauss–Seidel value from (52) and $\omega \in (1,2)$ is the relaxation parameter [26, 27]. For the model Poisson problem on a square, Young's formula $\omega_{\text{opt}} \approx 2/\left(1+\sin(\pi h)\right)$ minimizes the spectral radius of the iteration matrix [27]; here $\omega = 1.85$ is used throughout. Each sweep costs $O(N^2)$ work on an $(N+1) \times (N+1)$ grid.

c. Stopping criterion and linearity test. Iteration stops when the root-mean-square discrete residual of (51) falls below 10^{-8}. For Part 2, linearity of (48) implies $u(f_1 + f_2) = u(f_1) + u(f_2)$ in exact arithmetic; the maximum pointwise defect $\max|u - (u_1 + u_2)|$ measures how well the discrete SOR solution preserves superposition [8, 23].

4. Results: Part 1 — solver comparison on f_1

Table VII lists performance at $N = 64$. Jacobi requires the most iterations (1.47×10^4) and wall time (55 s); Gauss–Seidel uses roughly half as many iterations and half the time, consistent with the effective halving of the error reduction per sweep [26]. SOR with $\omega = 1.85$ converges in only 550 iterations and 2.4 s while reaching the same residual tolerance and the same RMS error 5.2×10^{-6} against (53). Figure 11 shows the resulting fields (sinusoidal bowl with $u = 0$ on the boundary) and semi-log residual histories: all three methods decrease the residual exponentially, with SOR steepest.

method	iterations	time (s)	residual	RMS error
Jacobi	14 722	55	1.0×10^{-8}	5.2×10^{-6}
Gauss–Seidel	7 362	28	1.0×10^{-8}	5.2×10^{-6}
SOR ($\omega = 1.85$)	550	2.4	9.7×10^{-9}	5.2×10^{-6}

TABLE VII. Jacobi, Gauss–Seidel, and SOR on f_1 (49) with $N = 64$.

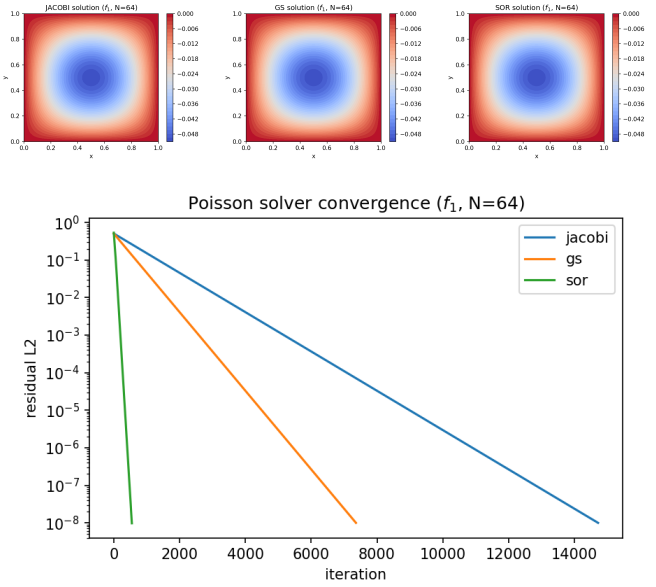

FIG. 11. Numerical solutions for f_1 (top: Jacobi, Gauss–Seidel, SOR) and discrete residual versus iteration at $N = 64$ (bottom).

a. Grid-size scaling. Table VIII and Fig. 12 summarize $N \in \{32, 48, 64, 96\}$. Iteration counts for Jacobi and Gauss–Seidel grow rapidly with N (roughly quadratically in N for this model problem, consistent with $\rho(R) \to 1$ as $h \to 0$), while SOR iterations increase much more slowly ($128 \to 1\,299$ over the same range). Wall time tracks iteration count modulated by the $(N-1)^2$ cost per sweep. Table IX shows that the RMS error against (53) decreases from 2.1×10^{-5} at $N = 32$ to 2.3×10^{-6} at $N = 96$, consistent with $O(h^2)$ accuracy of (51) [11, 23]. All three methods yield identical RMS errors at each N to the digits reported.

N	method	iterations	time (s)
32	Jacobi	3 680	3.3
32	GS	1 841	1.7
32	SOR	128	0.14
64	Jacobi	14 722	56
64	GS	7 362	28
64	SOR	550	2.4
96	Jacobi	33 121	281
96	GS	16 562	142
96	SOR	1 299	13

TABLE VIII. Iteration count and wall time versus grid size for f_1.

N	RMS error vs (53)
32	2.1×10^{-5}
48	9.2×10^{-6}
64	5.2×10^{-6}
96	2.3×10^{-6}

TABLE IX. Discretization error on f_1 (identical for Jacobi, GS, and SOR at each N).

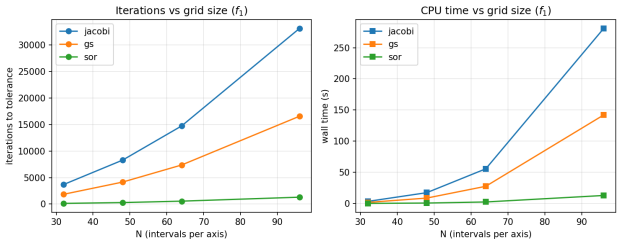

FIG. 12. Iterations to tolerance (left) and wall time (right) versus N for f_1.

5. *Results: Part 2 — linearity*

Because (48) is linear in f, the discrete SOR solution should satisfy $u(f_1 + f_2) \approx u_1 + u_2$ up to rounding error. Table X reports the maximum pointwise defect on three grids; values remain at the 10^{-10}–10^{-9} level, confirming superposition to near machine precision. The defect grows slightly with N, as expected when more unknowns accumulate floating-point updates [1, 26].

Figure 13 compares $u(f_1 + f_2)$, u_2, and u_1 at $N = 96$: the combined field is dominated by the smooth f_1 component (u_1), with a localized depression from the Gaussian f_2 near $(\frac{1}{2}, \frac{1}{2})$. Figure 14 shows $|u - (u_1 + u_2)|$ on the same grid; the defect peaks near the Gaussian center at $O(10^{-9})$, far below the $O(10^{-6})$ discretization error of u_1 itself.

N	$\max\lvert u - (u_1 + u_2)\rvert$
32	1.0×10^{-10}
64	8.2×10^{-10}
96	9.3×10^{-10}

TABLE X. Linearity defect for SOR solutions of f_1, f_2, and $f_1 + f_2$.

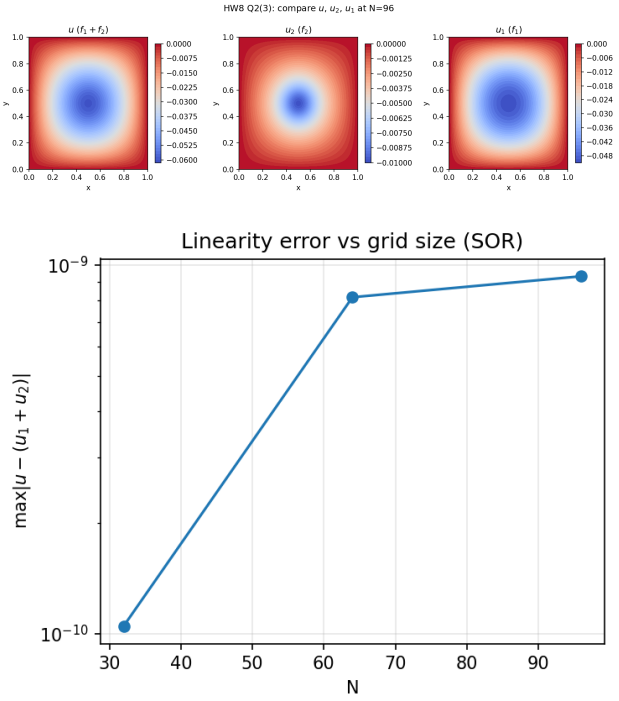

FIG. 13. Comparison of $u(f_1 + f_2)$, u_2, and u_1 at $N = 96$ (top) and linearity defect versus N (bottom).

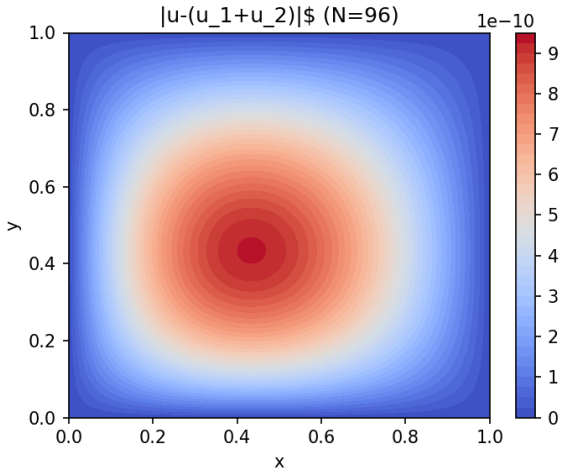

FIG. 14. Pointwise defect $|u - (u_1 + u_2)|$ for the combined source $f_1 + f_2$ at $N = 96$ (SOR).

III. CONCLUSION

HW1–HW5 cover finite-dimensional linear algebra through time integration. HW6 compares FTCS and Crank–Nicolson velocity profiles with the analytical Stokes solution, including a temporal convergence study at $L = 10$ (Fig. 8). HW7 applies CN to a two-dimensional heat equation, verifies the steady manufactured solution $\phi = (1-x^2)(1-y^2)$ to $L_2 \sim 10^{-9}$ (Table V), and confirms second-order temporal accuracy (Fig. 10). HW8 implements Jacobi, Gauss–Seidel, and SOR for Poisson's equation, compares cost versus grid size on f_1, and verifies discrete linearity for $f_1 + f_2$.

[1] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed. (Johns Hopkins University Press, Baltimore, MD, 2013).
[2] L. N. Trefethen and D. Bau, *Numerical Linear Algebra* (SIAM, Philadelphia, PA, 1997).
[3] C. L. Lawson and R. J. Hanson, Classics in Applied Mathematics **15** (1995), reprint of 1974 Prentice–Hall edition.
[4] D. H. Ackley, *A Connectionist Machine for Genetic Hillclimbing*, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA (1987).
[5] J. D. Hunter, Computing in Science & Engineering **9**, 90 (2007).
[6] C. G. Broyden, Mathematics of Computation **19**, 577 (1965).
[7] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations* (SIAM, Philadelphia, PA, 1996) corrected reprint of 1983 Prentice–Hall edition.
[8] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations* (SIAM, Philadelphia, PA, 1995).
[9] L. N. Trefethen, *Approximation Theory and Approximation Practice* (SIAM, Philadelphia, PA, 2013).
[10] K. E. Atkinson, *An Introduction to Numerical Analysis*, 2nd ed. (Wiley, New York, 1989).
[11] R. L. Burden, J. D. Faires, and A. M. Burden, *Numerical Analysis*, 10th ed. (Cengage Learning, Boston, MA, 2016).
[12] C. Runge, Zeitschrift für Mathematik und Physik **46**, 224 (1901).
[13] T. J. Rivlin, *Chebyshev Polynomials: From Approximation Theory to Algebra and Number Theory*, 2nd ed. (Wiley, New York, 1990).
[14] P. J. Davis and P. Rabinowitz, *Methods of Numerical Integration*, 2nd ed. (Dover, Mineola, NY, 2007).
[15] A. Quarteroni, R. Sacco, and F. Saleri, *Numerical Mathematics*, 2nd ed. (Springer, New York, 2007).
[16] D. P. Kroese, T. Taimre, and Z. I. Botev, *Handbook of Monte Carlo Methods* (Wiley, Hoboken, NJ, 2011).
[17] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*, 2nd ed. (Springer, Berlin, 1993).
[18] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*, 3rd ed. (Wiley, Chichester, UK, 2016).
[19] A. M. Nathan, American Journal of Physics **76**, 496 (2008).
[20] R. K. Adair, *The Physics of Baseball*, 3rd ed. (HarperCollins, New York, 2002).
[21] G. K. Batchelor, *An Introduction to Fluid Dynamics* (Cambridge University Press, Cambridge, UK, 1967).
[22] G. G. Stokes, Transactions of the Cambridge Philosophical Society **9**, 8 (1851).
[23] R. J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations* (SIAM, Philadelphia, PA, 2007).
[24] J. C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*, 2nd ed. (SIAM, Philadelphia, PA, 2004).
[25] J. Crank and P. Nicolson, Proceedings of the Cambridge Philosophical Society **43**, 50 (1947).
[26] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed. (SIAM, Philadelphia, PA, 2003).
[27] D. M. Young, *Iterative Solution of Large Linear Systems* (Academic Press, New York, 1971).