CMSE 821: Homework 2

Fall 2025

Assigned: Sep. 22, 2025 **Due:** Oct. 20, 2025

PAGE LIMIT: 25 pages (single-sided). Include a cover page (not counted toward the 25-page limit). Code listings *do* count toward the limit.

Submission. Submit one PDF (derivations, figures, and an AI Appendix) plus a repo/zip with runnable `.py` or `.ipynb`. Use `sympy`, `numpy`, `matplotlib`, `scipy`. **No Matlab.**

AI Collaboration Policy (Read First). You are encouraged to use AI tools to brainstorm and draft. However:

- **Verification is mandatory.** Every AI-produced formula or code snippet must be validated with *symbolic checks*, *unit tests*, and/or *numerical experiments*.
- **Provenance is required.** Include an *AI Appendix* with (i) your exact prompts; (ii) model name; (iii) date/time; (iv) a brief note on what you accepted or rejected and *why*.
- **Dual sourcing.** For core derivations, obtain two distinct approaches (e.g., Taylor vs. moment/Vandermonde; direct vs. variational) and reconcile them, or explain any discrepancy.
- **Authorship.** The final math, code comments, and explanations must be *in your own words*. Cite AI assistance where used.

Code Documentation (applies to all code).

- Each function/method begins with a header (purpose, author, dates, inputs/outputs with shapes/units, dependencies).
- Keep subroutines focused (about ≤ 1 page each). Factor long logic into helpers.
- Comment *why* each nontrivial step is done (not only *what*).
- Provide a minimal usage example or unit test for each public-facing routine.

Part 0 — AI-Assisted Problems (n = 32 experiments)

These four problems come first. Each requires: (i) an AI-produced draft/derivation; (ii) your independent *derivation; (iii) a symbolic/numerical verification; (iv) a short discussion. Use $n = 32$ interior points unless noted.*

Problem 1. Jacobi vs. Gauss–Seidel on 1D Poisson.

Consider Dirichlet Poisson $-u'' = f$ on $[0, 1]$, 3-point stencil, $n = 32$ interior points ($h = 1/(n+1)$).

(a) **AI:** Derive Jacobi and Gauss–Seidel iteration matrices and state $\rho(T)$ in terms of eigenmodes.

(b) **You:** Derive T_J and T_{GS} directly; identify eigenvectors $\phi_k(i) = \sin(\pi k i/(n + 1))$.

(c) **Verify:** Compute $\rho(T)$, run both methods on a manufactured solution, and plot $\|u^{(k)} - u^\star\|_2$ vs. k.

(d) **Discuss:** Why GS converges faster (high-frequency smoothing).

Problem 2. SOR and the role of ω.

(a) **AI:** Derive $T_{\text{SOR}}(\omega)$; predict how $\rho(T_{\text{SOR}})$ varies with $\omega \in (0,2)$.
(b) **You:** Implement SOR; sweep $\omega \in \{0.5, 0.8, 1.0, 1.2, 1.5, 1.8\}$; fit asymptotic rate from $\log E_k$.
(c) **Verify:** Compare measured rate vs. spectral radius of $T_{\text{SOR}}(\omega)$.
(d) **Discuss:** Identify near-optimal ω for this model problem.

Problem 3. When Jacobi fails.

(a) **AI:** Provide a 3×3 linear system where Jacobi diverges (e.g., not strictly diagonally dominant).
(b) **You:** Prove $\rho(T_J) \geq 1$ for the example; fix it if AI's example is incorrect.
(c) **Verify:** Run Jacobi and GS; plot error vs. iteration; relate to $\rho(T)$.

Problem 4. Smoothing property on sine modes.

Let A be the 1D Poisson matrix; use weighted Jacobi with weight ω.
(a) **AI:** Derive the amplification factor $\mu_k(\omega)$ for error mode ϕ_k.
(b) **You:** Derive $\mu_k(\omega)$ and evaluate $|\mu_k|$ for $n = 32$, $\omega = 2/3$; identify the most damped (high-k) range.
(c) **Verify:** Initialize error as ϕ_k, run 15 steps, and compare measured decay with $|\mu_k|^m$.
(d) **Discuss:** How this underpins multigrid (smooth high-k, coarse-grid fix low-k).

Part 1 — Two-Point Boundary Value Problems

Problem 5. Two-point BVP with integral constraint.

Consider

$$u''(x) = 1, \quad 0 < x < 1, \qquad u'(0) = 0,\ u'(1) = 1,$$

with the constraint $\int_0^1 u(x)\,dx = 1$.
(a) Solve analytically.
(b) Devise and implement a second-order accurate numerical method (you may adapt HW1 code). Verify the observed order.

Part 2 — Simple Iteration Methods

Problem 6. mSOR vs. SOR on model blocks.

Define the mSOR variant via

$$(D - L)\,u^{\text{GS}} = U\,u^{[k]} + f, \qquad u^{[k+1]} = u^{[k]} + \omega\big(u^{\text{GS}} - u^{[k]}\big),$$

with $A = D - L - U$. (Traditional SOR: $D\,u^{\text{GS}} = L\,u^{[k+1]} + U\,u^{[k]} + f$, same update.)
(a) For $A = \begin{bmatrix} 1 & \rho \\ -\rho & 1 \end{bmatrix}$: when does Gauss–Seidel converge?
(b) For the same A: for which ω does mSOR converge? What is the optimal ω?
(c) Repeat (a)–(b) for $A = \begin{bmatrix} I_n & S_n \\ -S_n^T & I_n \end{bmatrix}$, where S_n is arbitrary. (Hint: use the SVD of S_n.)

Problem 7. Richardson-type relaxation on a tridiagonal system.

Let $A \in \mathbb{R}^{n\times n}$ be tridiagonal with $a_{ii} = 3$ and $a_{i,i\pm1} = -1$. Consider

$$x^{k+1} = x^k + \omega\bigl(b - Ax^k\bigr), \quad k = 0, 1, 2, \ldots$$

For which real ω does this converge to the solution of $Ax = b$ for every $x^0 \in \mathbb{R}^n$?

Part 3 — Elliptic PDEs

Problem 8. Nine-point discretization with variable coefficient.

Discretize

$$-\partial_x\bigl(a(x,y)\,u_x\bigr) - \partial_y\bigl(a(x,y)\,u_y\bigr) = f(x,y), \quad u = 0 \text{ on } \partial\Omega, \quad \Omega = [-1,1]^2,$$

using the nine-point Laplacian approach (adapt to handle $a(x,y)$). Use natural rowwise ordering and derive the linear system. In the case $a \equiv \text{const}$, your method should be $\mathcal{O}(h^4)$. What is the order in the general case?

Problem 9. SOR implementation and tuning.

Implement SOR in **Python** for the linear system above without explicitly forming the full matrix. Experimentally determine the near-optimal relaxation ω_{opt}.

Problem 10. Order verification for constant coefficient.

Verify $\mathcal{O}(h^4)$ accuracy for

$$a(x,y) = 1, \quad f(x,y) = 5\pi^2 \sin\bigl(\pi(x-1)\bigr)\sin\bigl(2\pi(y-1)\bigr), \quad u(x,y) = \sin\bigl(\pi(x-1)\bigr)\sin\bigl(2\pi(y-1)\bigr).$$

Use $U \equiv 0$ as the initial guess.

Problem 11. Variable coefficient experiment.

Apply your method to

$$a(x,y) = 1 + 3\exp\bigl(-3(x+y)^2 - (x-y)^2\bigr), \qquad f(x,y) = 1.$$

Use $U \equiv 0$ as the initial guess. Do all of the following:

(a) Experimentally determine the order of accuracy. Use a very fine-grid solution as "exact."

(b) For several h, produce a `semilogy` plot of the L_2-norm of the residual vs. iteration number.

(c) From numerical experiments, how does the iteration count depend on h?

(d) From numerical experiments, how does wall time depend on h? (In Python, time with `time.perf_counter()` or similar.)

Part 4 — Geometric Multigrid for 1D Helmholtz

Problem 12. Step-by-step geometric multigrid for 1D Helmholtz.

Consider the 1D Helmholtz equation on $[0, 1]$:

$$-u''(x) + \kappa^2 u(x) = f(x), \qquad u(0) = u(1) = 0,$$

with $\kappa \geq 0$. Use uniform grids with $n = 2^L - 1$ interior points.

(a) **Discretization.** Derive the tridiagonal system $A_h u_h = f_h$ with stencil $\frac{1}{h^2}[-1,\ 2+\kappa^2 h^2,\ -1]$. State consistency and second-order accuracy.

(b) **Smoother.** Implement weighted Jacobi or Gauss–Seidel as a smoother: $u^{(m+1)} = u^{(m)} + \omega D^{-1}(f - Au^{(m)})$. Analyze the smoothing factor on sine modes and choose ω.

(c) **Intergrid transfer.** Define full-weighting restriction I_h^{2h} and linear interpolation I_{2h}^h. Give their stencils and prove that I_{2h}^h reproduces linear functions exactly.

(d) **Two-grid V-cycle.** Build a symmetric two-grid cycle with ν_1 pre-smooths and ν_2 post-smooths. Measure the convergence factor per V-cycle for several κ and h.

(e) **(Bonus)** Discuss how large κh impacts convergence and which modifications (e.g., Krylov acceleration, alternative smoothers) help.