Department of Applied Mathematics
Preliminary Examination in Numerical Analysis
Solution guide May 2025

Instructions

You have three hours to complete this exam. Submit solutions to four (and no more) of the following six problems. Please start each problem on a new page. You MUST prove your conclusions or show a counter-example for all problems unless otherwise noted. Write your student ID number (not your name!) on your exam.

Problem 1: Rootfinding methods
Suppose we want to compute the square root $x = \sqrt{a}$ for some $a > 0$. We can do this by using Newton's method to solve $x^2 = a$.

(a) Set up the problem outlined above as a rootfinding problem for a function $f(x)$. Suppose that α is a root of $f(x)$ and that the initial guess x_0 is sufficiently close to α. What are sufficient conditions required for local convergence of Newton's method? Check that this problem satisfies those requirements.

(b) Beginning with some $x_0 > 0$, the iteration formula for this is

$$x_{k+1} = \frac{1}{2}\left(x_k + \frac{a}{x_k}\right).$$

Derive this iteration formula.

(c) Suppose that the conditions in part (a) are satisfied. Show that Newton's method converges locally for x_0 sufficiently close to α.

(d) Show that the sequence of iterations is locally quadratically convergent. (Solving this part (d) correctly also gives full credit for part (c).)

Proposed solution:

(a) We have $f(x) = x^2 - a$ with root $\sqrt{a}$. We check that f has continuous first and second derivatives and that $f'(\sqrt{a}) \neq 0$. We have $f'(x) = 2x$ and $f''(x) = 2$ so these are continuous on $\mathbb{R}$. Also, we have $f'(\sqrt{a}) = 2\sqrt{a} \neq 0$ since $a > 0$.

(b) We have $f(x) = x^2 - a$ and $f'(x) = 2x$. Therefore, we have

$$x_{k+1} \quad = \quad x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{x_k^2 - a}{2x_k} = \frac{1}{2}\left(2x_k - x_k + \frac{a}{x_k}\right) = \frac{1}{2}\left(x_k + \frac{a}{x_k}\right).$$

(c) Let $\phi(x) = x - \frac{f(x)}{f'(x)}$. Then $\phi'(x) = \frac{f(x)f''(x)}{[f'(x)]^2}$ so that $\phi'(x)$ is continuous. Also, notice that $\phi'(\alpha) = 0$ since α is a root of f. Since $\phi'(\alpha)$ is continuous at α, $\exists \delta > 0$ such that for all $x \in \mathcal{B}(\alpha, \delta) = \{x : |x - \alpha| < \delta\}$,

$$|\phi'(x) - \phi'(\alpha)| = |\phi'(x)| < C,$$

where $0 < C < 1$. Let $x_k \in \mathcal{B}(\alpha, \delta)$ and let $e_{k+1} = x_{k+1} - \alpha$ be the error at the $(k+1)^{th}$ iteration. Then

$$e_{k+1} = \phi(x_k) - \phi(\alpha).$$

Using the first order Taylor expansion of ϕ about α, we have

$$e_{k+1} = \phi'(\xi_k)e_k$$

for $\xi_k \in \mathcal{B}(\alpha, x_k)$. Since $x_k \in \mathcal{B}(\alpha, \delta)$ and $\xi_k \in \mathcal{B}(\alpha, x_k)$, $\xi_k \in \mathcal{B}(\alpha, \delta)$. So $|\phi'(\xi_k)| < C < 1$. Also, $e_k = x_k - \alpha$ gives us $|e_k| < \delta$ and therefore,

$$|e_{k+1}| \leq |\phi'(\xi_k)||e_k| < C\,|e_k| < \delta.$$

Therefore, if $x_0 \in \mathcal{B}(\alpha, \delta)$, then $x_k \in \mathcal{B}(\alpha, \delta)$ for $k = 1, 2, ...$ Beginning with $k = 0$ and applying this recursively over k iterations, we have

$$|e_k| \leq C\,|e_{k-1}| < C^k\,|e_0|.$$

Since $C^k \to 0$, this means that $e_k \to 0$. So the iterate sequence converges to the root α.

(d) To get the rate of convergence, we again examine $e_{k+1} = \phi(x_k) - \phi(\alpha)$. After applying the second order Taylor expansion of ϕ around α, we have

$$e_{k+1} = \frac{1}{2}\phi''(\eta_k)e_k^2$$

for $\eta_k \in \mathcal{B}(\alpha, x_k)$. Notice that the second derivative of $\phi(x)$ is $\phi''(x) = \frac{f''(x)}{2f'(x)}$. Since f has continuous first and second derivatives, $\phi''(x)$ is also continuous.

Dividing both sides of the above equation by e_k^2 and taking the limit as $k \to \infty$ gives us

$$\lim_{k\to\infty} \frac{e_{k+1}}{e_k^2} = \lim_{k\to\infty} \frac{1}{2}\phi''(\eta_k).$$

Since $x_k \to \alpha$, and $\eta_k \in \mathcal{B}(\alpha, x_k)$, we have $\eta_k \to \alpha$. Combining this with the fact that $\phi''(x)$ is continuous gives us

$$\lim_{k\to\infty} \frac{e_{k+1}}{e_k^2} = \frac{1}{2}\phi''(\alpha).$$

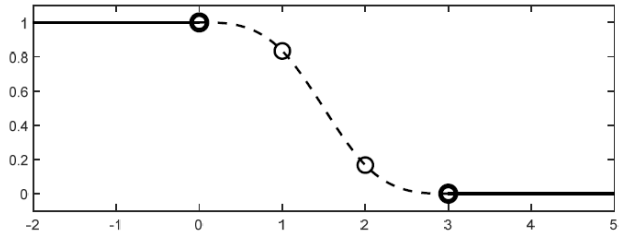
Therefore, the iterate sequence is quadratically locally convergent.

Problem: Interpolation and Approximation
We consider in this problem cubic splines.

(a) Determine how many extra end conditions are needed to determine a cubic spline uniquely. Also, describe two common ways to ensure this uniqueness.

(b) Define what is meant by a B-spline (also known as a basis spline).

(c) The cubic spline that transitions the fastest from identically one to identically zero on a unit-spaced grid looks graphically as follows:



Determine the exact function values for this spline at the two internal node points (located at $x = 1$ and $x = 2$).

Proposed Solution:

(a) With n nodes, continuity of function and of its first and second derivative at $(n-2)$ interior nodes gives $3(n-2)$ relations. We also need to provide n function values, for a total of $4n-6$ relations. The $n-1$ cubics have in all $4n-4$ parameters. For uniqueness, one typically supplies two more conditions (or arranges to remove two freedoms).

Common ways to handle this end condition issue include: (i) Enforce zero second derivative at both ends (giving what is called a natural spline), (ii) Disallow a discontinuity in the third derivative one step in from each end (not-a-knot condition), and (iii) Make the derivative at each end point match that of a cubic that interpolates at the four nodes nearest each end.

(b) The B-spline is the spline that non-trivially transitions from identically zero back to identically zero over the fewest possible number of nodes. It is typically also normalized so that its integral evaluates to one.

(c) Each time we pass a node, a cubic spline can feature a jump in its third derivative (but not in any lower order derivatives). In the present example, the functional form in the successive subintervals thus becomes:

$$\begin{aligned}
(-\infty, 0] &\quad 1 \\
[0, 1] &\quad 1 + \alpha x^3 \\
[1, 2] &\quad 1 + \alpha x^3 + \beta(x-1)^3 \\
[2, 3] &\quad 1 + \alpha x^3 + \beta(x-1)^3 + \gamma(x-2)^3 \\
[3, \infty) &\quad 1 + \alpha x^3 + \beta(x-1)^3 + \gamma(x-2)^3 + \delta(x-3)^3 = 0
\end{aligned}$$

The last expression should evaluate to identically zero. It can be expanded as

$$x^3(\alpha+\beta+\gamma+\delta) + x^2(-3\beta-6\gamma-9\delta) + x(3\beta+12\gamma+27\delta) + (1-\beta-8\gamma-27\delta) = 0$$

All the coefficients being zero gives the linear system:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -3 & -6 & -9 \\ 0 & 3 & 12 & 27 \\ 0 & -1 & -8 & -27 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1 \end{bmatrix}$$

with solution $[\alpha, \beta, \gamma, \delta] = [-\frac{1}{6}, \frac{1}{2}, -\frac{1}{2}, \frac{1}{6}]$. Thus, we obtain: Spline value at $x = 1$ is $1 - \frac{1}{6} = \frac{5}{6}$. . Spline value at $x = 2$ is $1 - \frac{1}{6}8 + \frac{1}{2} = \frac{1}{6}$. . The solution of the linear system can be simplified by noting that, by (anti-) symmetry of the spline around $(x, y) = (3/2, 1/2)$, it will hold that $\delta = -\alpha$ and $\gamma = -\beta$.

Problem 3: Numerical integration
Gaussian quadrature is commonly used for numerical approximation of integrals. One generalization is to instead apply it to approximate infinite sums. Determine the nodes x_1, x_2 and weights w_1, w_2 so that the formula

$$\sum_{n=0}^{\infty} \frac{f(n)}{n!} = w_1 f(x_1) + w_2 f(x_2)$$

becomes exact for polynomials $f(x)$ of as high degree as possible.

Hint: The inner product to use becomes

$$< f, g >= \sum_{n=0}^{\infty} \frac{1}{n!} f(n) g(n)$$

Sums of the form $\sum_{n=0}^{\infty} \frac{n^p}{n!}$ can be found in closed form by considering the derivative of $e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$ at $x = 1$, multiplying by x and differentiating, etc.

Proposed Solution:

- Determine the nodes: The first few orthogonal polynomials are of the form

$$\begin{aligned} p_0(n) &= 1 \\ p_1(n) &= n + a \\ p_2(n) &= n^2 + bn + c \end{aligned}$$

  Orthogonality (with $\frac{1}{n!}$ as weight function) implies:

$$< p_0, p_1 > = \sum \frac{1}{n!}(n + a) = 0 \tag{1}$$

$$< p_0, p_2 > = \sum \frac{1}{n!}(n^2 + bn + c) = 0 \tag{2}$$

$$< p_1, p_2 > = \sum \frac{1}{n!}(n + a)(n^2 + bn + c) = 0 \tag{3}$$

  Using the hint, we find that $\sum \frac{1}{n!} = e$, $\sum \frac{n}{n!} = e$, $\sum \frac{n^2}{n!} = 2e$ and $\sum \frac{n^3}{n!} = 5e$. We then get from (1) that $a = -1$, and then from (2) and (3) we get a 2×2 system of equations: $2 + b + c = 0$ and $3 + b = 0$, which means $b = -3, c = 1$.

  Hence, $p_2(n) = n^2 - 3n + 1$, with roots $x_1 = \frac{3-\sqrt{5}}{2}$ and $x_2 = \frac{3+\sqrt{5}}{2}$. There are therefore the node locations for the quadrature formula.

- Determine the weights: It suffices to impose that we get the exact result in the cases of $f(n) = 1$ and $f(n) = n$, giving the system:

$$w_1 + w_2 = e \left(\frac{3 - \sqrt{5}}{2} \right) w_1 + \left(\frac{3 + \sqrt{5}}{2} \right) w_2 = e$$

  with solution $w_1 = \frac{e}{2}\left(1 + \frac{1}{\sqrt{5}}\right)$ and $w_2 = \frac{e}{2}\left(1 - \frac{1}{\sqrt{5}}\right)$.

Problem 4: Numerical Linear Algebra
Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be a diagonalizable matrix with eigenvalues

$$|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n| > 0,$$

and corresponding linearly independent eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$, such that

$$\mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{v}_i, \quad \text{for } i = 1, 2, \ldots, n.$$

(a) State the condition(s) required for convergence of the Power Method to the dominant eigenvector. Write pseudocode for the Power Method to approximate the eigenvector associated with the largest magnitude eigenvalue of $\mathbf{A}$.

(b) Assume that the condition(s) in part (a) are satisfied. Show that the Power Method converges to the dominant eigenvector. (For simplicity, you can assume that $\lambda_1 > 0$.)

(c) Using your solution in part (b), show that the Rayleigh Quotient converges to the dominant eigenvalue.

(d) Suppose $|\lambda_{n-1}| > |\lambda_n|$. Modify the Power Method to approximate an eigenvector corresponding to the smallest magnitude eigenvalue. Explain why your modification works. State the condition(s) required for convergence and write pseudocode for this procedure.

Solution:

(a) The initial vector $\mathbf{x}^{(0)}$ needs to have a nonzero component in the direction of the dominant eigenvector.

Pseudocode: Starting with initial vector $\mathbf{x}^{(0)}$, repeat until convergence:

- Compute $\mathbf{y} = \mathbf{A}\mathbf{x}^{(k)}$
- Normalize: $\mathbf{x}^{(k+1)} = \frac{\mathbf{y}}{\|\mathbf{y}\|}$.

(b) Let the initial vector $\mathbf{x}^{(0)}$ be a linear combination of the eigenvectors so that

$$\mathbf{x}^{(0)} = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \cdots + c_n\mathbf{v}_n,$$

where $c_1 \neq 0$. At the k^{th} iteration, the Power Method update is

$$\mathbf{x}^{(k)} = \frac{\mathbf{A}^k\mathbf{x}^{(0)}}{\|\mathbf{A}^k\mathbf{x}^{(0)}\|}.$$

Focusing on $\mathbf{A}^k\mathbf{x}^{(0)}$, we have

$$\begin{aligned}
\mathbf{A}^k\mathbf{x}^{(0)} &= \mathbf{A}^k(c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \cdots + c_n\mathbf{v}_n) \\
&= c_1\lambda_1^k\mathbf{v}_1 + c_2\lambda_2^k\mathbf{v}_2 + \cdots + c_n\lambda_n^k\mathbf{v}_n \\
&= \lambda_1^k\left(c_1\mathbf{v}_1 + c_2\left(\frac{\lambda_2}{\lambda_1}\right)^k\mathbf{v}_2 + \cdots + c_n\left(\frac{\lambda_n}{\lambda_1}\right)^k\mathbf{v}_n\right).
\end{aligned}$$

As $k \to \infty$, all the $\left(\frac{\lambda_i}{\lambda_1}\right)^k \to 0$ for $i \geq 2$ since $|\lambda_i/\lambda_1| < 1$. Therefore,

$$\lim_{k\to\infty} \mathbf{x}^{(k)} = \lim_{k\to\infty} \frac{\mathbf{A}^k\mathbf{x}^{(0)}}{\|\mathbf{A}^k\mathbf{x}^{(0)}\|} = \lim_{k\to\infty} \frac{\lambda_1^k c_1\,\mathbf{v}_1}{|\lambda_1^k c_1|\,\|\mathbf{v}_1\|} = \text{sign}(c_1)\,\mathbf{v}_1.$$

(c) The Rayleigh quotient at the k^{th} iteration of the Power Method is

$$\mu^{(k)} = \frac{(\mathbf{x}^{(k)})^T \mathbf{A}\mathbf{x}^{(k)}}{(\mathbf{x}^{(k)})^T \mathbf{x}^{(k)}}.$$

Since $\mathbf{A}\mathbf{v}_1 = \lambda_1 \mathbf{v}_1$ and since we saw in part (b) that $\mathbf{x}^{(k)} \to \text{sign}(c_1)\, \mathbf{v}_1$, we have

$$\lim_{k\to\infty} \mu^{(k)} = \frac{\lambda_1 \|\mathbf{v}_1\|^2}{\|\mathbf{v}_1\|^2} = \lambda_1.$$

(d) We apply the Power Method to $\mathbf{A}^{-1}$. Since the eigenvalues of $\mathbf{A}^{-1}$ are $\frac{1}{\lambda_1}, \frac{1}{\lambda_2}, \ldots, \frac{1}{\lambda_n}$, the largest magnitude eigenvalue of $\mathbf{A}^{-1}$ is the smallest magnitude eigenvalue of $\mathbf{A}$.

Pseudocode: Starting with initial vector $x^{(0)}$, repeat until convergence:

- Solve $\mathbf{A}\mathbf{y} = \mathbf{x}^{(k)}$.
- Normalize: $\mathbf{x}^{(k+1)} = \frac{\mathbf{y}}{\|\mathbf{y}\|}$.

Problem 5: Numerical ODE (25 points)

We wish to solve an IVP for a system of N 1st order ODEs $\{\vec{y}'(t) = f(t, \vec{y}),\ \vec{y}(0) = \vec{y}_0\}$. We consider four linear, multistep methods (LMMs) derived from finite difference formulas:

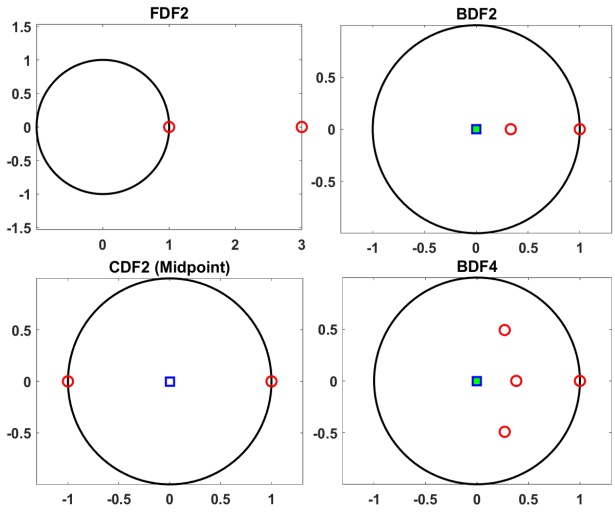
$$\text{FDF2}: \quad \frac{1}{2}\left(-3y_n + 4y_{n+1} - y_{n+2}\right) = \Delta t f(t_n, y_n)$$
$$\text{CDF2}: \quad \frac{1}{2}\left(-y_n + y_{n+2}\right) = \Delta t f(t_{n+1}, y_{n+1})$$
$$\text{BDF2}: \quad \frac{1}{2}\left(y_n - 4y_{n+1} + 3y_{n+2}\right) = \Delta t f(t_{n+2}, y_{n+2})$$
$$\text{BDF4}: \quad \frac{1}{12}\left(3y_n - 16y_{n+1} + 36y_{n+2} - 48y_{n+3} + 25y_{n+4}\right) = \Delta t f(t_{n+4}, y_{n+4})$$

(a) Write down a definition of the **truncation error** for the FDF2 multistep method. This method is obtained from a formula for the derivative of a smooth function $Y(t)$, satisfying:

$$Y'(t) - \frac{-3Y(t) + 4Y(t + \Delta t) - Y(t + 2\Delta t)}{2\Delta t} = O(\Delta t^2)$$

What can you conclude about the truncation errors for FDF2 from this? If FDF2 converges, what is expected for the global error?

(b) Recall that for LMM $\sum_{m=0}^{p} a_m y_{n+m} = \Delta t \sum_{m=0}^{p} b_m f(t_{n+m}, y_{n+m})$ we can use $\rho(w) = \sum_{m=0}^{p} a_m w^m$ and $\sigma(w) = \sum_{m=0}^{p} b_m w^m$ to analyze its properties.



In each of these plots, roots of $\rho(w)$ are denoted as circles, and roots of $\sigma(w)$ as squares. If the root is simple, the marker face is empty. If it is repeated, it is filled with color.

Assume all of the methods presented above are consistent. Using the information in these plots or otherwise, indicate what you know about each method's (i) stability, (ii) convergence and (iii) relative (strong) stability. Be sure to justify each of these conclusions.

(c) Write down a simple pseudocode for a function using BDF2 to integrate the IVP for the system of N ODEs from time 0 to time T taking n_t time steps. Be sure to indicate inputs and outputs required for this routine.

Proposed Solution:

(a) The definition of truncation (local) error is the error we make by taking one step in our method using *exact* initial data. Let $Y(t)$ denote the exact solution of our IVP. Plugging it into the FDF2 method, we get:

$$\begin{aligned} T_{n+2}(Y) &= (-3Y(t_n) + 4Y(t_{n+1}) - Y(t_{n+2})) - 2\Delta t f(t_n, Y(t_n)) \\ &= (-3Y(t_n) + 4Y(t_{n+1}) - Y(t_{n+2})) - 2\Delta t Y'(t_n) \end{aligned}$$

Multiplying both sides of the given result for the forward difference formula by $2\Delta t$, we see this implies that our truncation error $T_{n+2}(Y) = O(\Delta t^3)$. Given that we add $O(\Delta t^{-1})$ local errors integrating our IVP, if this method converges, the global error is expected to be one power less of Δt, that is, $O(\Delta t^2)$.

(b) We will use three main results from the theory of stability and convergence of LMMs:

1. We say an LMM satisfies the root condition (RC) if, for all roots ρ_i of $\rho(w)$, it is either true that $|\rho_i| < 1$ or $|\rho_i| = 1$ and it is a simple root ($\rho'(\rho_i) \neq 0$). A consistent LMM is stable if and only if it satisfies the RC.
2. **Dahlquist Equivalence Theorem:** a consistent LMM is convergent if and only if it satisfies RC.
3. We say an LMM satisfies a strong root condition (SRC) if it satisfies RC and the only root of magnitude 1 is the principal root $\rho_0 = 1$. SRC implies relative (strong) stability. If SRC is not satisfied, this is evidence that our method might not be relatively stable; we then would have to investigate further.

Armed with these three, we can then state what we know for each method based on the information given:

- **FDF2:** One of the roots of $\rho(w)$ is clearly outside the unit disk. By (1), it is unstable. By (2), it does not converge. Instability implies our method cannot be relatively stable.
- **CDF2:** This method has two simple roots on the boundary of the unit disk, so by (1), it is stable. By (2), it is convergent. Since it does not satisfy SRC, by (3), this is evidence it might not be relatively stable.

  Note: This is the famous Midpoint method, which is given as an example of a method which is not weakly stable, since the root of $\rho(w) - \lambda\Delta t\sigma(w)$ of the form $-1 + O(\Delta t^2)$ is always comparable to the principal one.
- **BDF2:** This method has two simple roots, one of which is magnitude 1 and one which is inside the disk. This means it satisfies both RC and SRC. By (1), it is stable. By (2), it is convergent. By (3), it is relatively stable.
- **BDF4:** This method has 4 simple roots, one of magnitude 1, and 3 inside the disk. This means it satisfies both RC and SRC. By (1), it is stable. By (2), it is convergent. By (3), it is relatively stable.

(c) BDF2 is an implicit method (A-stable, part of a family widely used to tackle stiff systems) of order 2. So, the step to go from data y_n, y_{n-1} to y_{n+1} will involve a (potentially non-linear) solve for a system of equations. We assume $f(t, \boldsymbol{y})$ is non-linear, access to a formula or routine to evaluate $\frac{\partial f}{\partial y}$, and that we have a routine for Newton, and use it as our solver.

- **Inputs:** Final time T, number of timesteps n_t, functions f and Jacobian $\frac{\partial f}{\partial y}$, initial guess $\boldsymbol{y}_0$.
- **Output:** Array containing $\boldsymbol{y_k}$ for $k = 0, \ldots, n_t$ as its columns.
- Initialize necessary quantities and arrays, e.g $\Delta t = T/n_t$.

- Take the first step to obtain y_1 using an implicit method with truncation error at least $O(\Delta t^2)$ (e.g. Backward Euler). Here it's acceptable (but unnecessary) to take a more expensive step to ensure y_1 is accurate to near machine precision.
- FOR $k = 1$ to $n_t - 1$:
- Define $g(y) = 3y - 2\Delta t f(t_{k+1}, y) + (y_{k-1} - 4y_k)$.
- Define $J_g(y) = 3\mathsf{I} - 2\frac{\partial f}{\partial y}(t_{k+1}, y)$.
- Set initial guess $z_0 = y_k$, TOL near machine precision and a reasonable MAXIT, e.g. 10-50 (maximum iterations).
- $y_{k+1} = \text{NewtonSolver}(g, Jg, z_0, \text{TOL}, \text{MAXIT})$
- END

Problem 6: Numerical PDE (25 points)

Consider the following elliptic two-point Boundary Value Problem (BVP) on $(0,1)$:

$$-au_{xx}(x) = -k^2 u(x) + f(x) \quad x \in (0,1)$$
$$u(0) = g_0 \quad u(1) = g_1$$

with $a > 0, k \geq 0$, $g_0, g_1 \in \mathbb{R}$ and f smooth.

(a) For a regular grid of points $x_i = ih$, with $i = 0, 1, \ldots, n$ and spacing $h = 1/n$, we denote our approximation $U_i \simeq u(x_i)$. Using second differences for u_{xx}, write down a system of equations for each unknown U_i. Indicate how the boundary data is incorporated.

(b) Note that by collecting all unknown entries in a vector $\boldsymbol{U}$, we can write the equations above as a linear system of the form $\mathsf{M}\boldsymbol{U} = \boldsymbol{b}$. Describe the entries and structure of matrix M. If possible, provide an estimate for its condition number $\kappa(\mathsf{M})$.

(c) Consider the associated parabolic problem:

$$u_t(x,t) = au_{xx}(x) - k^2 u(x) + f(x) \quad x \in (0,1), t \in (0,T)$$
$$u(0,t) = g_0 \quad u(0,t) = g_1, t \in (0,T)$$
$$u(x,0) = \phi(x) \quad x \in (0,1)$$

for ϕ smooth. Use the same discretization in space as in (a)-(b) to define a linear IVP on the vector $\boldsymbol{U}(t)$ with entries $U_i(t) \simeq u(x_i, t)$. Then, use a Backward Euler method in time to write down a time-stepping formula for $\boldsymbol{U}(t_{k+1})$ in terms of $\boldsymbol{U}(t_k)$, for $t_k = k\Delta t$, $\Delta t = T/n_t$.

(d) Assume M is Symmetric Positive Definite (SPD), with eigenvalues in the interval (η_1, η_2) where $\eta_1, \eta_2 \geq 0$, $\eta_1 \leq \lambda_{min}(\mathsf{M})$ and $\eta_2 \geq \lambda_{max}(\mathsf{M})$. Explain how you can use this information to perform a stability analysis on the scheme proposed in (c). Indicate what restriction (if any) there is on timestep size Δt.

Proposed Solution:

(a) We set $U_0 = g_0$ and $U_n = g_1$ to satisfy boundary conditions, so our $(n-1)$ unknowns are U_i for $i = 1, \ldots, n-1$. Using the standard second difference formula for u_{xx}, we obtain the following equation for unknown U_i:

$$\frac{a}{h^2}\left(-U_{i+1} + 2U_i - U_{i-1}\right) + k^2 U_i = f(x_i) \quad i = 1, \ldots, n-1$$

we note that for $i = 1$ and $i = n-1$, one of the terms is known (given by boundary data). We send it to the right hand side:

$$\frac{a}{h^2}\left(-U_2 + 2U_1\right) + k^2 U_1 = f(x_1) + \frac{ag_0}{h^2}$$
$$\frac{a}{h^2}\left(2U_{n-1} - U_{n-2}\right) + k^2 U_{n-1} = f(x_{n-1}) + \frac{ag_1}{h^2}$$

(b) Writing the $n-1$ linear equations in (a) in the form $M\boldsymbol{U} = \boldsymbol{b}$ gives us a system with associated matrix M which is tri-diagonal and SPD, and a right-hand-side containing data from $f(x_i)$ and boundary conditions. Our matrix is

$$\mathsf{M} = \frac{a}{h^2}\begin{bmatrix} 2+\frac{k^2h^2}{a} & -1 & 0 & \cdots & 0 & 0 \\ -1 & 2+\frac{k^2h^2}{a} & -1 & \cdots & 0 & 0 \\ 0 & -1 & 2+\frac{k^2h^2}{a} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2+\frac{k^2h^2}{a} & -1 \\ 0 & 0 & 0 & \cdots & -1 & 2+\frac{k^2h^2}{a} \end{bmatrix}$$

Using Gershgorin disks, we can conclude that all eigenvalues of M are contained on the interval $\frac{a}{h^2}(\frac{k^2h^2}{a}, 4+\frac{k^2h^2}{a}) = (k^2, \frac{4a}{h^2}+k^2)$. So, an upper bound for condition number would be $\kappa(\mathsf{M}) = \frac{\lambda_{max}}{\lambda_{min}} \leq \frac{\frac{4a}{h^2}+k^2}{k^2} = \frac{4a}{k^2h^2}+1$.

Note: We can write down formulas for the eigenvalues of M if we know those of the matrix for the second derivative matrix D_2, which were derived on class; these yield $\lambda_k = \frac{4}{h^2}\sin^2(\frac{k\pi}{2n})$ for $k = 1, \ldots, n-1$.

(c) Applying the discretization and notation used above, we first get the linear IVP:

$$\begin{aligned} \boldsymbol{U}'(t) &= -\mathsf{M}\boldsymbol{U}(t) + \boldsymbol{b} \quad t \in (0,T) \\ \boldsymbol{U}(0) &= \boldsymbol{\phi} \end{aligned}$$

where vector $\boldsymbol{\phi}$ has entries $\phi(x_i)$. Applying Backward Euler in time for timesteps $t_k = k\Delta t$, with $\Delta t = T/n_t$, $k = 0, 1, \ldots, n_t$ gives us:

$$\boldsymbol{U}^{k+1} = \boldsymbol{U}^k + \Delta t\left(-\mathsf{M}\boldsymbol{U}^{k+1} + \boldsymbol{b}\right)$$

We send all terms with superindex $k+1$ (corresponding to future time t_{k+1}) to the left-hand-side, which gives us a linear equation. Our time-stepping is given by

$$\begin{aligned} (\mathsf{I} + \Delta t\mathsf{M})\,\boldsymbol{U}^{k+1} &= \boldsymbol{U}^k + \Delta t\boldsymbol{b} \quad t \in (0,T) \\ \boldsymbol{U}^0 &= \boldsymbol{\phi} \end{aligned}$$

To implement this, we would use a linear system solver to solve the system of equations for $\boldsymbol{U}^{k+1}$ defined above.

(d) We note that both the discrete solution $\boldsymbol{U}^k$ and the corresponding error at discretization points $\boldsymbol{E}^k$ satisfy equations of the form

$$\begin{aligned} \boldsymbol{U}^{k+1} &= (\mathsf{I} + \Delta t\mathsf{M})^{-1}\left(\boldsymbol{U}^k + \Delta t\boldsymbol{b}\right) \\ \boldsymbol{E}^{k+1} &= (\mathsf{I} + \Delta t\mathsf{M})^{-1}\left(\boldsymbol{E}^k + \Delta t\boldsymbol{T}^k\right) \end{aligned}$$

with $\boldsymbol{T}^k$ vector of truncation errors. Given eigenvalues $\lambda_i(\mathsf{M}) > 0$, the eigenvalues of the matrix "propagating" the error from time t_k to time t_{k+1} are $\frac{1}{1+\Delta t\lambda_i(\mathsf{M})}$ which are also positive, and smaller than 1 in magnitude for all $\Delta t \geq 0$. This means the method is unconditionally stable, and so, we have no restriction on Δt in terms of h^2 (as we would had we used an explicit method like Forward Euler).