Scientific Computing Comprehensive Exam

January 2015

Answer **5 questions** *of your choice. Include a full explanation of the process by which each answer is derived, including any assumptions you make that are not stated in the problem. Partial credit will be awarded for presenting a viable solution strategy. No credit will be given for computations that are presented without an explanation of the approach.*

1. (a) Use the backward Euler method to transform the predator-prey model

$$u' = (2 - v)\,u,$$
$$v' = (u - 2)\,v,$$

into a discrete nonlinear system.

(b) Advance the solution by time step $\Delta t = 1$ from initial condition $u(0) = 1$, $v(0) = 1$ by application of Newton's method.

2. In Gauss-Radau quadrature, one of the quadrature nodes is fixed, say the left endpoint, so that

$$\int_{-1}^{1} f(x)\,\mathrm{d}x = w_0 f(-1) + \sum_{i=1}^{n} w_i f(x_i) + R(f). \qquad (1)$$

Determine the quadrature weights $w_0, \ldots, w_n$ and quadrature nodes $x_1, \ldots, x_n$ to ensure that (1) has the highest possible degree of precision. Deduce the form of the remainder $R(f)$.

3. Consider the following scheme to solve $y'(x) = f(x, y)$ with initial condition $y(0) = y_0$:

$$y_{n+1} = \frac{3}{2}y_n - \frac{1}{2}y_{n-1} + \frac{h}{4}\left(5f_n - 3f_{n-1}\right).$$

What is the order of this scheme? Is this method convergent?

4. Let $I(x)$ be the linear approximation of the function $\cos(ax)$ over the interval $[0, \pi/2]$ with minimum error in the infinity norm. Consider the operator $F[a]$ that maps a to $I(x)$. Define the condition number $\kappa(F)$ of $F : a \mapsto I(x)$, and compute $\kappa(F)$ for $a \in (1 - \varepsilon, 1 + \varepsilon)$, $\varepsilon \in \mathbb{R}_+$.

5. (a) Show by example that Gaussian elimination without pivoting is numerically unstable.

   (b) Is Gaussian elimination with partial pivoting stable? If so, provide a proof; if not, provide an example demonstrating its instability.

6. Consider the least squares problem $\min_x ||Ax - b||_2$ with $A \in \mathbb{R}^{m\times n}$, $m > n$, and $b \in \mathbb{R}^m$.

   (a) Assume A has full rank. Describe how to solve the least squares problem using (i) the normal equations, (ii) the QR factorization, and (iii) the SVD decomposition. Discuss the condition number and the number of operations for each method.

   (b) Discuss what happens when the rank of A is less than n.

7. Consider a dense matrix $A \in \mathbb{C}^{n\times n}$ and the series $S = I + \sum_{k=1}^{\infty} A^k$.

   (a) Show that the series is convergent if and only if the spectral radius of A, $\rho(A)$, is strictly smaller than 1. Show that S is invertible and $S = (I - A)^{-1}$.

   (b) When $\rho(A) \ll 1$, can we use (a) to produce an effective method for solving $(I - A)x = b$? Compare this approach to Gaussian elimination.

   (c) Assume that *most* of the eigenvalues of A satisfy $|\lambda_i| \ll 1$, but that there exist a small number of eigenvalues with $|\lambda_k| > 1$. Discuss effective methods to solve $(I - A)x = b$ using ideas from (a), (b), and what you learned from MATH 662.

Problem 1

$$\begin{cases} u' = (2-v)u \\ v' = (u-2)v \end{cases}$$

a) Recall backward Euler: $y_{n+1} = y_n + h f(y_{n+1}, t_{n+1})$

$$\Rightarrow \begin{cases} u_{n+1} = u_n + h(2 - v_{n+1})u_{n+1} \\ v_{n+1} = v_n + h(u_{n+1} - 2)v_{n+1} \end{cases} \Rightarrow \boxed{\begin{cases} u_n = (1 - 2h + h v_{n+1})\, u_{n+1} \\ v_n = (1 + 2h - h u_{n+1})\, v_{n+1} \end{cases}}$$

b) Let $h = 1$ and $\vec{x}_0 = \begin{pmatrix} u(0) \\ v(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ and recall multi-dimentional Newton's method:

$$\vec{x}_{n+1} = \vec{x}_n - (DF(x_n))^{-1} F(x_n)$$

$$F(\vec{x}_n) = F\begin{pmatrix} u_{n+1} \\ v_{n+1} \end{pmatrix} = \begin{bmatrix} (1 - 2h + h v_{n+1})u_{n+1} - u_n \\ (1 + 2h - h u_{n+1}) v_{n+1} - v_n \end{bmatrix} = \begin{bmatrix} (-1 + v_{n+1}) u_{n+1} - u_n \\ (3 - u_{n+1}) v_{n+1} - v_n \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$$

$$DF(\vec{x}_n) = \begin{bmatrix} \nabla f_1 \\ \nabla f_2 \end{bmatrix} = \begin{bmatrix} v_{n+1} - 1 & u_{n+1} \\ -v_{n+1} & 3 - u_{n+1} \end{bmatrix}$$

$$\Rightarrow F(x_0) = \begin{bmatrix} (-1+1)1 - 1 \\ (3-1)1 - 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

$$DF(\vec{x}_0) = \begin{bmatrix} 0 & 1 \\ -1 & 2 \end{bmatrix} \Rightarrow (DF(\vec{x}_0))^{-1} = \begin{bmatrix} 2 & -1 \\ 1 & 0 \end{bmatrix}$$

$$\Rightarrow \vec{x}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 2 & -1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} -3 \\ -1 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$$

Problem 2

Define an inner product: $\langle p,q \rangle = \int_{-1}^{1} (x+1)\, pq \, dx$

To find n nodes, we need to find roots of an orthogonal, n^{th} degree polynomial. To find this polynomial, we start with the basis $\{1, x, x^2, \dots, x^n\}$ and use Gramm Schmidt (and the inner product defined above) to get the orthogonal polynomials.

Next we need to find n+1 weights. To do this, solve $\int_{-1}^{1} (x+1) f(x)\, dx$ for $f(x) = 1, x, \dots, x^n$ and set it equal to $w_0 f(-1) + \sum_{i=1}^{n} w_i f(x_i)$. Since we found x_i's in the previous step, this generates a system of n+1 eqns w/ n+1 unknowns that we can solve.

The precision of Guassian quadrature with n+1 nodes is:

$$2(n+1) - 1 = 2n+1$$

So Guass-Radau has precision 2n.

The error for Guassian quadrature w/ n+1 nodes is:

$$E_G = \frac{f^{(2(n+1))}(\xi)}{(2(n+1))!} \int_a^b q^2(x)\, w(x)\, dx = C_G f^{(2n+2)}(\xi_G)$$

so the error for Gaussian-Radau should be of the form:

$$E_{G\text{-}R} = C_{G\text{-}R} f^{(2n+1)}(\xi_{GR})$$

Jan 2015

Problem 3

$$y_{n+1} = \frac{3}{2} y_n - \frac{1}{2} y_{n-1} + \frac{h}{4}(5 f_n - 3 f_{n-1})$$

$$y_{n+1} = y_n + h y_n' + \frac{h^2}{2} y_n'' + \frac{h^3}{3!} y_n''' + O(h^4)$$

$$= \frac{3}{2} y_n$$

$$- \frac{1}{2}\left(y_n - h y_n' + \frac{h^2}{2} y_n'' - \frac{h^3}{3!} y_n''' + O(h^4)\right)$$

$$+ \frac{5h}{4}\left(y_n' \right)$$

$$- \frac{3h}{4}\left(y_n' - h y_n'' + \frac{h^2}{2!} y''' + O(h^3)\right)$$

$O(1)$: $1 = \frac{3}{2} - \frac{1}{2}$ ✓

$O(h)$: $1 = \frac{1}{2} + \frac{5}{4} - \frac{3}{4}$ ✓

$O(h^2)$: $\frac{1}{2} = -\frac{1}{4} + \frac{3}{4}$ ✓

$O(h^3)$: $\frac{1}{6} = \frac{2 \cdot 1}{2 \cdot 12} - \frac{3 \cdot 3}{3 \cdot 8} = -\frac{8}{24} = -\frac{1}{3}$:(

Thus (if the method converges) it is <u>second order</u>.

To have convergence, we need stability.

<u>Zero stable</u>

To do this, let $y' = 0 \Rightarrow y_{n+1} - \frac{3}{2} y_n + \frac{1}{2} y_{n-1} = 0$. Let $y_n = r^n$

$\Rightarrow r^{n-1}\left(r^2 - \frac{3}{2} r + \frac{1}{2}\right) = 0 \Rightarrow r^{n-1}(2r^2 - 3r + 1) = 0 \Rightarrow r = \frac{1}{2}, 1$

$(2r-1)(r-1)$

$\Rightarrow y_n = c_1 (1)^n + c_2 (\frac{1}{2})^n \Rightarrow \lim_{n\to\infty} y_n < \infty$

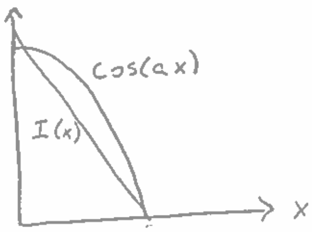
$\Rightarrow$ Our method is zero stable

$\therefore$ Our method is convergent

Problem 4

Recall the condition number formula:

$$K(F) = \sup_{\|\delta a\|} \frac{\|\delta F\| / \|F\|}{\|\delta a\| / \|a\|} = \frac{\|J\|}{\|F\| / \|a\|} \qquad (1)$$



$F: a \to I(x)$ a will affect the slope of I so:

$$I = -\lambda x + b \qquad m, b > 0 \qquad (2)$$

Depend on a

Let $f(x) = \cos(ax)$. Error should be maximized at $x=0$.

For $x=0$: error $= I(0) - f(0) = b - 1$

$\exists\, x_0 \in (0, \frac{\pi}{2})$: error $= I(x_0) - f(x_0) = \lambda x_0 + b - \cos(ax_0) = -E = 1-b$

$$g(x_0) = -\lambda x_0 + b - \cos(ax_0) \qquad (3)$$

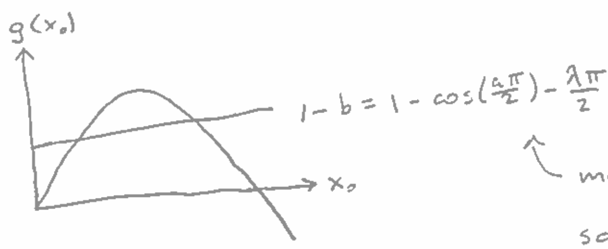
Note: $I(\frac{\pi}{2}) = \cos(a\frac{\pi}{2})$

$-\lambda(\frac{\pi}{2}) + b = \cos(\frac{a\pi}{2})$

$$b = \cos\left(\frac{a\pi}{2}\right) + \frac{\lambda\pi}{2} \qquad (4)$$

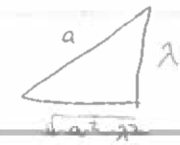
$$-\lambda x_0 + 2\cos(a\tfrac{\pi}{2}) + \lambda\pi = \cos(ax_0)$$

$$g(x_0) = -\lambda x_0 + \cos\left(\frac{a\pi}{2}\right) + \frac{\lambda\pi}{2} - \cos(ax_0)$$



max error and $g(x_0)$ = error so $1-b$ must be at peak of $g(x_0)$

$$g'(x_0) = -\lambda + a\sin(a x_0) = 0$$



$$x_0 = \frac{\arcsin(\lambda/a)}{a}$$

$$g(x_0) = -\lambda\left(\frac{\arcsin(\lambda/a)}{a}\right) + \cos\left(\frac{a\pi}{2}\right) + \frac{\pi\lambda}{2} - \cos\left(\alpha\,\frac{\arcsin\left(\frac{\lambda}{a}\right)}{\alpha}\right)$$

$$= -\frac{\lambda}{a}\arcsin\left(\frac{\lambda}{a}\right) + \cos\left(\frac{a\pi}{2}\right) + \frac{\pi\lambda}{2} - \frac{\sqrt{a^2-\lambda^2}}{a} = 1 - \cos\left(\frac{a\pi}{2}\right) - \frac{\lambda\pi}{2}$$

Need to solve this mess for λ then sub λ into (4) to get b. This gives us the opperator $F: a \mapsto I(x)$. Once we have that we can find the Jacobian, J, and use (1) to find our condition number.

Problem 5

a) Consider the matrix $A = \begin{bmatrix} 10^{-16} & 1 \\ 1 & 1 \end{bmatrix}$ which has the

LU-decomp: $L = \begin{bmatrix} 1 & 0 \\ 10^{16} & 1 \end{bmatrix}$ $U = \begin{bmatrix} 10^{-16} & 1 \\ 0 & 1-10^{16} \end{bmatrix} \approx \begin{bmatrix} 10^{-16} & 1 \\ 0 & -10^{16} \end{bmatrix}$

$\Rightarrow LU \approx \begin{bmatrix} 10^{-16} & 1 \\ 1 & 0 \end{bmatrix} \not\approx A$

b) Consider any matrix of the form: $A = \begin{bmatrix} 1 & & 0 & 1 \\ & \ddots & & \vdots \\ -1 & & \ddots & 1 \end{bmatrix}$

ie: $A = \begin{bmatrix} 1 & 0 & 1 \\ -1 & 1 & 1 \\ -1 & -1 & 1 \end{bmatrix}$ which has the LU-decomp:

$L = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ $U = \begin{bmatrix} 1 & & 1 \\ 0 & & 2 \\ 0 & 0 & 4 \end{bmatrix}$

It is technically stable but in practice, its not.

(Read Trefetham)

Problem 6

a) Minimizing Ax-b is the same thing as saying it's orthogonal to the column space of A or:

$$A^T(Ax-b)=0$$

$$\Rightarrow A^TAx - A^Tb = 0$$

i) Normal equations

Alg: 1) form A^TA matrix and A^Tb vector

2) Compute LU-decomp: $A^TA = LU$

3) solve $Ly = A^Tb$ for y

4) solve $Ux = y$ for x

Operation Count: A^TA + LU-decomp $\sim mn^2 + \frac{n^3}{3}$ flops

Cond #: $\text{cond}(A^TA) = \text{cond}(L)\cdot\text{cond}(U)$

Conclusion: fast, but sensitive to rounding errors

ii) QR factorization:

Let $A = QR$ $\Rightarrow$ b can be projected onto range(A) by QQ^T

$\Rightarrow$ Ax = projected b $\Rightarrow$ $QRx = QQ^Tb$ $\Rightarrow$ $Rx = Q^Tb$

Alg: 1) compute $A = QR$

2) Compute Q^Tb

3) solve $Rx = Q^Tb$ for x

Operation count: $2mn^2 - 2n^3/3$ flops

Cond #: $\text{cond}(A) = \text{cond}(Q)\,\text{cond}(R) = \text{cond}(R)$

Conclusion: fast enough, good stability

iii) SVD decomp:

Let $A = U\Sigma V^T \Rightarrow b$ can be projected onto range(A) by UU^T

$\Rightarrow Ax =$ projected $b \Rightarrow U\Sigma V^T x = UU^T b \Rightarrow \Sigma V^T = U^T b$

Alg: 1) Compute $A = U\Sigma V^T$

2) compute $U^T b$

3) solve $\Sigma y = U^T b$ for y

4) compute $x = Vy$

Operation Count: SVD $\sim 2mn^2 + 11n^3$ flops

Cond #: $\text{cond}(A) = \text{cond}(U)\,\text{cond}(\Sigma)\,\text{cond}(V^T) = \text{cond}(\Sigma) = \dfrac{|\lambda_{max}|}{|\lambda_{min}|}$

Conclusion: Slowest option but very stable

b) If A doesn't have full rank, we should worry about the stability of the method we choose. For A without full rank, we would minimize $Ax-b$ using SVD decomp because it is the most stable.