Numerical Analysis Preliminary Exam
10.00AM–1.00PM, JANUARY 19, 2018

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

For the iterations (a)–(d) state (i) whether it converges to α (for initial conditions sufficiently close to the root), (ii) if it converges give the order of convergence (e.g. linear, quadratic, etc), and (iii) if it converges linearly give the rate of convergence, i.e. compute $\lim_{k\to\infty} |\alpha - x_{k+1}|/|\alpha - x_k|$. Justify your answers.

(a) $x_{k+1} = -1 + x_k + x_k^2$, $\alpha = 2$

(b) Newton's method for $f(x) = x(1-x)^2$, $\alpha = 1$.

(c) $x_{k+1} = x_k^2 + x_k^{-2} - 1$, $\alpha = 1$.

(d) Newton's method for $f(x) = \sin(x)$, $\alpha = \pi$.

(e) For what initial conditions x_0 does Newton's method for $f(x) = e^{-1/x}$ converge to the root $\alpha = 0$? Show that the iteration converges sublinearly; specifically, show that the error ratio x_{k+1}/x_k behaves asymptotically like e^{-x_k} in the sense that $\lim_{k\to\infty} \ln(x_{k+1}/x_k)^{1/x_k} = -1$

Solution:

(a) No convergence since $\alpha = 2$ is not a fixed point.

(b) The root has multiplicity 2, so for a smooth function Newton's method will converge linearly with rate 1/2.

(c) The fixed-point iteration function is $g(x) = x^2 + x^{-2} - 1$; at the fixed point we have $g'(1) = 0$ so convergence is at least linear. The second derivative is $g''(1) \neq 0$ so convergence will be quadratic.

(d) This is a smooth function with a simple root, so Newton's method will converge at least quadratically. However, the iteration function is $g(x) = x - \tan(x)$; the second derivative is $g''(\pi) = 0$ so convergence is actually faster than quadratic. The third derivative is $g^{(3)}(\pi) \neq 0$, so convergence is cubic.

(e) The iteration function that corresponds to Newton's method is $g(x) = x - x^2$. If $x_k < 0$ then $x_{k+1} < x_k$, so the iteration will not converge to 0 for $x_0 < 0$. If $x_k > 1$ then $x_{k+1} < 0$, so the iteration will not converge to 0 for $x_0 > 1$. If $0 < x_k < 1$ then $0 < x_{k+1} < x_k$, so the iteration is bounded below and decreasing $\Rightarrow$ it must converge to something; the only possible candidate is $x_k \to 0$. Finally, if $x_0 = 1$ then $x_1 = 0$, so the iteration will converge to 0 for all $0 \leq x_0 \leq 1$.

Now that we know it converges, what is the rate?

$$x_{k+1} - 0 = x_k - 0 - x_k^2 \Rightarrow \frac{|x_{k+1} - 0|}{|x_k - 0|} = |1 - x_k|.$$

If we take the limit $k \to \infty$ the RHS is 1, which means that the convergence is sublinear. Note that the absolute values above are superfluous since everything is positive. As directed, consider

$$\ln\left(\frac{x_{k+1}}{x_k}\right) = \ln(1 - x_k)$$

$$\frac{1}{x_k}\ln\left(\frac{x_{k+1}}{x_k}\right) = \frac{\ln(1 - x_k)}{x_k}$$

Take the limit as $k \to \infty$. The RHS is -1 (simple Taylor expansion), which proves the desired result.

Problem 2: Interpolation & Approximation

(a) Find the polynomial that interpolates $f(0) = 0,\ f(1) = 1, f(2) = 0, f(3) = 1, f(4) = 0$ using Newton divided differences. Use the Newton table to generate the necessary divided differences.

(b) Construct the generalized Newton table for the Hermite type data
$f(0) = 0, f'(0) = 1,\ f''(0) = 0, f(1) = 1, f'(1) = 0$. The two leftmost columns in this table are $(x_0, x_0, x_0, x_1, x_1)^T$ and $(f[x_0], f[x_0], f[x_0], f[x_1], f[x_1])^T$. To fill in the rest of the table you must relate the divided differences to the derivative data.
Hint: Let $f[x_0, x_0] = \lim_{x \to x_0} f[x_0, x]$.

(c) Let $f(x)$ be a smooth function and $p(x)$ be the unique Hermite interpolation polynomial satisfying

$$\left.\frac{d^l p(x)}{dx^l}\right|_{x=x_i} = \left.\frac{d^l f(x)}{dx^l}\right|_{x=x_i}, \quad l = 0, \ldots, m, \quad i = 0, 1.$$

Show that the error in the interpolation is orthogonal to the interpolant p in the (semi) inner product

$$(v, w)_{m+1} = \int_{x_0}^{x_1} \left(\frac{d^{m+1}v}{dx^{m+1}}\right)\left(\frac{d^{m+1}w}{dx^{m+1}}\right) dx.$$

Solution:
(a) The Newton table is:

0	0				
		1			
1	1		-1		
		-1		$2/3$	
2	0		1		$-1/3$
		1		$-2/3$	
3	1		-1		
		-1			
4	0				

Thus the interpolating polynomial is:

$$p(x) = 0 + 1(x - 0) - 1(x - 0)(x - 1) + 2/3(x - 0)(x - 1)(x - 2) - 1/3(x - 0)(x - 1)(x - 2)(x - 3).$$

(b) The error formula is

$$\begin{aligned} f(x) - p(x) &= [x_0, x_1, \ldots, x_{n-1}, x]f(x - x_0)(x - x_1)\cdots(x - x_{n-1}) \\ &= (x - x_1)\cdots(x - x_{n-1})\frac{f^{(n-1)}(\xi_x)}{(n-1)!}, \end{aligned}$$

where $\xi_x \in \text{int}(x, x_0, x_1, \ldots, x_n)$. By letting the points coalesce and evaluating the error at x_0 we thus see that $[x_0, x_0]f = f'(x_0)$, $[x_0, x_0, x_0]f = \frac{1}{2!}f''(x_0)$ etc.

The generalized Newton table in this case is:

$$\begin{array}{c|cccccc}
x_0 & [x_0]f & & & & \\
 & & f'(x_0) & & & \\
x_0 & [x_0]f & & \frac{1}{2}f''(x_0) & & \\
 & & f'(x_0) & & [x_0,x_0,x_0,x_1]f & \\
x_0 & [x_0]f & & [x_0,x_0,x_1]f & & [x_0,x_0,x_0,x_1,x_1]f \\
 & & [x_0,x_1]f & & [x_0,x_0,x_1,x_1]f & \\
x_1 & [x_0]f & & [x_0,x_1,x_1]f & & \\
 & & f'(x_1) & & & \\
x_1 & [x_1]f & & & &
\end{array}$$

Plugging in numbers and computing the divided differences gives

$$\begin{array}{c|ccccc}
0 & 0 & & & & \\
 & & 1 & & & \\
0 & 0 & & 0 & & \\
 & & 1 & & 0 & \\
0 & 0 & & 0 & & -1 \\
 & & 1 & & -1 & \\
1 & 1 & & -1 & & \\
 & & 0 & & & \\
1 & 1 & & & &
\end{array}$$

Thus the polynomial is $p(x) = 0 + 1x + 0x^2 + 0x^3 - 1x^3(x-1) = x - x^3(x-1)$.
(c) We want to show that $(p, f-p)_{m+1} = 0$. Note that the degree of p is $(2m+1)$ and thus

$$\frac{d^{2m+2}p}{dx^{2m+2}} = 0.$$

Now, repeated integration by parts together with the interpolation conditions yields

$$\begin{aligned}
(p, f-p)_{m+1} &= \int_{x_0}^{x_1} \left(\frac{d^{m+1}p}{dx^{m+1}}\right)\left(\frac{d^{m+1}(f-p)}{dx^{m+1}}\right) dx \\
&= -\int_{x_0}^{x_1} \left(\frac{d^{m+2}p}{dx^{m+2}}\right)\left(\frac{d^{m}(f-p)}{dx^{m}}\right) dx + \left[\left(\frac{d^{m+1}p}{dx^{m+1}}\right) \underbrace{\left(\frac{d^{m}(f-p)}{dx^{m}}\right)}_{\text{zero by interp. cond.}}\right]_{x_0}^{x_1} \\
&\vdots \\
&= -\int_{x_0}^{x_1} \underbrace{\left(\frac{d^{2m+2}p}{dx^{2m+2}}\right)}_{\text{zero by degree}}(f-p)\, dx + \left[\left(\frac{d^{2m+1}p}{dx^{2m+1}}\right) \underbrace{(f-p)}_{\text{zero by interp. cond.}}\right]_{x_0}^{x_1} = 0.
\end{aligned}$$

(c) [Alternative]. As above plus: Since $(p, f-p)_{m+1} = 0$, the Pythagorean theorem gives

$$(f,f)_{m+1} = (p,p)_{m+1} + (f-p, f-p)_{m+1},$$

thus $(f,f)_{m+1} \geq (p,p)_{m+1}$.

Problem 3: Quadrature

(a) Let the weights in the quadrature formula $\int_a^b f(x)\mathrm{d}x \approx \sum_{i=0}^n w_i f(x_i)$ with distinct nodes $x_0, \ldots, x_n$ be based on integrating the unique polynomial of degree $\leq n$ that interpolates the data. Give a formula relating the weights w_i to the Lagrange interpolating polynomials $\ell_i(x)$.

(b) Let the weights in the quadrature formula $\int_a^b f(x)\mathrm{d}x \approx \sum_{i=0}^n w_i f(x_i)$ with distinct nodes $x_0, \ldots, x_n$ be chosen so that the quadrature exactly integrates all polynomials up to degree $\leq n$. Show that the resulting weights are the same as in part (a).

(c) Find weights w_0, w_1, and w_2 and nodes $x_0, x_1, x_2 \in [-1, 1]$ such that the quadrature $\int_{-1}^1 f(x)\mathrm{d}x \approx \sum_i w_i f(x_i)$ integrates all quintic polynomials exactly.

Solution:

(a) The formula is $w_i = \int_a^b \ell_i(x)\mathrm{d}x$ where $\ell_i(x)$ is the i^{th} Lagrange interpolating polynomial.

(b) First recall that the Lagrange polynomials form a basis for all polynomials of degree $\leq n$, so it suffices to find weights w_i such that all the Lagrange polynomials are integrated exactly. These weights would then have to satisfy

$$\int_a^b \ell_i(x)\mathrm{d}x = \sum_i w_i \ell_i(x_i)$$

But the RHS is just w_i, which proves the result.

(c) This is a Gaussian quadrature problem with weight function $w(x) = 1$, i.e. Gauss-Legendre. The nodes must be the roots of the third Legendre polynomial, which is

$$P_3(x) = \frac{1}{2}(5x^3 - 3x).$$

you can find this by applying Gram-Schmidt to the monomials, though this is a lot of work. You can also recall that the third Legendre polynomial is cubic and antisymmetric, so it must have the form $x^3 - bx$ (un-normalized); the fact that it must be orthogonal to all polynomials of lower degree, and to x^2 in particular, sets the coefficient to $b = 3/5$. The roots are $x_0 = -\sqrt{3/5}$, $x_1 = 0$, $x_2 = \sqrt{3/5}$. The Lagrange polynomial $\ell_1(x)$ is

$$\ell_1(x) = \frac{x^2 - 3/5}{-3/5} = 1 - \frac{5x^2}{3}$$

The weight w_1 is the integral of this over $[-1, 1]$ which is $w_1 = 8/9$. The other weights are equal to each other (by symmetry), and the sum of the weights is 2 so $w_0 = w_2 = 5/9$.

Problem 4: Numerical Linear Algebra

Householder matrices form one of the most important 'building blocks' in several numerical linear algebra methods.

(a) Write down the general form (definition) of a Householder matrix $\mathbf{H}$.

(b) Show from this form that $\mathbf{H}$ is both Hermitian and unitary.

(c) Given two vectors $\vec{x}$ and $\vec{y}$, describe the condition(s) on these vectors such that one can find a Householder matrix $\mathbf{H}$ satisfying $\mathbf{H}\vec{x} = \vec{y}$. Show that these condition(s) indeed is (are) required.

(d) Assuming the condition(s) in part c is (are) satisfied, describe how one actually determines this matrix $\mathbf{H}$ when given $\vec{x}$ and $\vec{y}$.

(e) Describe how these Householder matrices can be used to similarity transform a square matrix to upper Hessenberg form.

Solution:

(a) $\mathbf{H} = \mathbf{I} - 2\vec{u}\vec{u}^*$ where $\vec{u}$ is a column vector of unit length.

(b) (i) Hermitian: $\mathbf{H}^* = \mathbf{I} - 2(\vec{u}\vec{u}^*)^* = \mathbf{I} - 2\vec{u}\vec{u}^* = \mathbf{H}$; (ii) Unitary: $\mathbf{H}^*\mathbf{H} = \mathbf{H}^2$ (by above) $= (\mathbf{I} - 2\vec{u}\vec{u}^*)^2 = \mathbf{I} - 4\vec{u}\vec{u}^* + 4\vec{u}\vec{u}^*\vec{u}\vec{u}^* = \mathbf{I}$ (since $\vec{u}^*\vec{u} = 1$).

(c) There are two conditions that need to be satisfied: (i) $\|\vec{y}\|_2 = \|\vec{x}\|_2$ (since $\vec{y}^*\vec{y} = \vec{x}^*\mathbf{H}^*\mathbf{H}\vec{x} = \vec{x}^*\vec{x}$), and (ii) $\vec{x}^*\vec{y}$ is real (since $\vec{x}^*\vec{y} = \vec{x}^*\mathbf{H}\vec{x}$ and $\mathbf{H} = \mathbf{H}^*$).

(d) $\vec{y} = \mathbf{H}\vec{x} = \vec{x} - 2\vec{u}\vec{u}^*\vec{x}$ where $\vec{u}^*\vec{x}$ is a scalar. Therefore the vector $\vec{u}$ must lie in the direction of $\vec{y} - \vec{x}$. Also, $\vec{u}$ is a unit vector. Therefore $\vec{u} = \pm(\vec{y} - \vec{x})/\|\vec{y} - \vec{x}\|$; either choice of sign is valid.

(e) We know from (d) how to find a Householder matrix $\mathbf{H}$ such that, with $\vec{x}$ given, we clear all entries but the first one in a column vector. Schematically:

$$\mathbf{H}\vec{x} = (*, 0, \cdots, 0)^T$$

Applying this idea to a full matrix $\mathbf{A}$ and using a $\mathbf{H}_1$ that is one step smaller allows us to introduce zeros from entry 3 and down in the first column:

$$\left[\begin{array}{c|c} 1 & \vec{0} \\ \hline \vec{0}^T & \mathbf{H}_1 \end{array}\right] \mathbf{A} = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ \vdots & * & * & * & * \\ 0 & * & * & * & * \end{bmatrix}$$

Making it a similarity transform by corresponding multiplication of the transpose from the left leaves the introduced zero pattern intact. Repeat this process with

$$\left[\begin{array}{c|c} \mathbf{I} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{H}_2 \end{array}\right]$$

where $\mathbf{I}$ is 2×2 from the left and the right, to get also the second column to Upper Hessenberg form. If $\mathbf{A}$ is $n \times n$ the process is completed after $n - 2$ steps.

Problem 5: ODEs

(a) Define the concept of *stability domain.*

(b) Determine the stability domain for the leap-frog scheme $y_{n+1} - y_{n-1} = 2kf(t_n, y_n)$ for solving the ODE $y' = f(t,y)$ (with k denoting the time step: $k = t_{n+1} - t_n$).

(c) Determine the leap-frog scheme's order of accuracy.

Consider next the following variation of the leap-frog scheme

$$y_{n+1} - y_{n-1} = k\left(\frac{7}{3}f(t_n, y_n) - \frac{2}{3}f(t_{n-1}, y_{n-1}) + \frac{1}{3}f(t_{n-2}, y_{n-2})\right).$$

(d) It can be shown that this scheme is third order accurate, and also that it entirely lacks a stability domain, apart from the single point at the origin. Can this scheme be used to solve $y' = y$ and/or $y' = -y$? Explain!

Solution:

(a) Consider the (scalar) test problem $y' = \lambda y$ (with analytic solution $y(t) = Ce^{\lambda t}$, decaying to zero as $t \to \infty$ if $\text{Re}\{\lambda\} < 0$). Then approximate $y' = \lambda y$ by a numerical scheme using time step $k > 0$ and define $\xi = k\lambda$. The stability domain is the set $\xi \in \mathbb{C}$ for which the numerical solution remains bounded as $t \to \infty$.

(b) Applied to $y' = \lambda y$, the leap-frog scheme becomes $y_{n+1} - y_n = 2\xi y_n$, with characteristic equation $r^2 - 2r\xi - 1 = 0$. The roots r_1 and r_2 must satisfy $r_1 + r_2 = 2\xi$ and $r_1 r_2 = -1$, the latter of which implies that to avoid growth, both roots must be of the form $r_1 = e^{i\alpha}$ and $r_2 = e^{-i\alpha}$. The equation $r_1 + r_2 = 2\xi$ then implies $\xi = i\sin(\alpha)$, i.e. the only values of ξ that avoid growth are those that are purely imaginary between $-i$ and i.

(c) One way to test the accuracy of a linear multi-step scheme is to apply it to the test functions $y = 1, t, t^2$, etc and see how far it remains exact. We get

$$\begin{aligned}
y = 1, y' = 0 &\qquad 1 - 1 = 2k \times 0 \qquad \text{OK}\\
y = t, y' = 1 &\qquad (t+k) - (t-k) = 2k \qquad \text{OK}\\
y = t^2, y' = 2t &\qquad (t+k)^2 - (t-k)^2 = 2k \times 2t \qquad \text{OK}\\
y = t^3, y' = 3t^2 &\qquad (t+k)^3 - (t-k)^3 \neq 2k \times (3t^2)\ \text{FAIL} \ \Rightarrow\ \text{second order.}
\end{aligned}$$

(d) The only requirements for convergence to the ODE solution are (i) consistency and (ii) root condition satisfied. Consistency is assured since the scheme is first order accurate (or better). Regarding the root condition, we get (as for the leap-frog scheme), when applied to the ODE $y' = 0$, the characteristic equation $r^2 - 1 = 0$, with roots $r_\pm = \pm 1$. There is no root outside the unit circle, and the roots on the periphery are simple. The proposed scheme will therefore converge to the true solution for both $y' = y$ and $y' = -y$ (as well as for the general ODE $y' = f(t,y)$).

Problem 6: PDEs

Consider the periodic initial boundary value problem

$$\begin{aligned} &u_t = u_x, \quad x \in [0, 2\pi],\, t > 0, \\ &u(x,t) = u(x+2\pi, t), \quad u(x,0) = e^{iKx}. \end{aligned}$$

Let v_j^n be a grid function approximating $u(x_j, t_n)$ on the equidistant space-time grid with nodes $(x_j, t_n) = (jh, nk)$, $h > 0, k > 0$, $j = 0, 1, \ldots, J$, $n = 0, 1, \ldots$

Find the coefficients c_{-1} and c_1 in the approximation

$$u_x(x_j, t_n) \approx \frac{1}{h}\left(c_1 v_{j+1}^n + c_{-1} v_{j-1}^n\right),$$

(a) so that the approximation is second order accurate,
(b) so that the approximation is exact for constants and for the initial data with $K = \frac{\pi}{2h}$.

For the temporal derivative consider the two approximations

$$u_t(x_j, t_n) \approx \frac{1}{k}(v_j^{n+1} - v_j^n), \qquad u_t(x_j, t_n) \approx \frac{1}{k}(v_j^{n+1} - A v_j^n),$$

where A is a spatial averaging operator defined as $Aw_j \equiv \frac{w_{j+1}+w_{j-1}}{2}$.
(c) The two spatial and two temporal approximations can be combined in four ways to approximate the PDE. Let $k = \lambda h$, with λ being a positive constant. In each case determine what values of λ yields a stable method.
(d) Which of the combinations results in consistent approximations to the PDE? You may simply state the result without deriving the local truncation error.

Solution:

For **(a)** use Tylor expansions around x_j with steps $\pm h$ to find $c_1 = -c_{-1} = \frac{1}{2}$.
The conditions in **(b)** requires that

$$\begin{aligned} c_1 + c_{-1} &= 0, \\ c_1 e^{ikh} + c_{-1} e^{-ikh} &= ikh, \end{aligned}$$

which simplifies to

$$\begin{aligned} c_1 + c_{-1} &= 0, \\ c_1 - c_{-1} &= \frac{\pi}{2}, \end{aligned}$$

when $kh = \pi/2$. The solution is $c_1 = -c_{-1} = \frac{\pi}{4} = \frac{\pi}{2}\frac{1}{2}$.
(c) Note that

$$\frac{\tau}{2h}\left(e^{ikh(j+1)} - e^{ikh(j-1)}\right) = i\frac{\tau}{h}\sin(kh)\, e^{ikhj}.$$

The first time discretization (Forward Euler) gives the amplification factor

$$\hat{g}(kh) = 1 + i\tau\lambda \sin(kh),$$

whose magnitude is maximized for the $\pm$ wave, i.e., $|\hat{g}(\pi/2)|^2 \sim (1 + (\tau\lambda)^2)$ leading to a solution with errors of size $\varepsilon_{\mathrm{m}}(1 + (\tau\lambda)^2)^{\frac{1}{k}}$ at time $t = 1$. The error thus grows without bound as $h, k \to 0$.

The second time discretization (Lax-Friedrich) gives the amplification factor

$$\hat{g}(kh) = \cos(kh) + i\tau\lambda \sin(kh),$$

$$\begin{aligned}
|\hat{g}(kh)|^2 &= \cos^2(kh) + (\tau\lambda)^2 \sin^2(kh) \\
&= 1 - \sin^2(kh) + (\tau\lambda)^2 \sin^2(kh) \\
&= 1 - (1 - (\tau\lambda)^2) \sin^2(kh).
\end{aligned}$$

Thus $|\hat{g}(kh)| < 1$ if $(1 - (\tau\lambda)^2) > 0$, that is $\tau\lambda < 1$, with $\tau = 1$ for the standard stencil and $\tau = \pi/2$ for the non-standard stencil.

(d) Using the standard stencil in space with either of the time discretizations gives consistent approximations to the PDE but only LxF gives a convergent approximation as the Forward Euler method is not stable. Note however that if we had not made the restriction that $k = \lambda h$ one may choose, say, $k = h^2$ to make this approximation stable and consistent in the Lax-Richtmyer sense. When using the non-standard spatial discretization scheme becomes consistent to the equation $u_t = \frac{\pi}{2} u_x$ and to the original equation, hence the scheme is not consistent with the PDE.

As an aside, you may wonder why one would ever want to use a scheme like this that is not even consistent! Here we made you minimize the error for a particular wave number as this gives straightforward calculations but it is possible to select the values of the coefficients by minimizing the dispersive error in an integrated sense over a range of wave numbers. This can be quite useful if one has a good understanding of the wave number content of the problem at hand. For example in computational aero acoustics the paper *Dispersion-relation-preserving finite difference schemes for computational acoustics* by Tam and Webb, JCP 1993, presents spatial discretizations that are quite popular as evidenced by the almost 1900 citations to that paper.