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 $\lim_{k\to\infty} \ln(x_{k+1}/x_k)^{1/x_k} = -1$

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

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

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.

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.

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!

Problem 6: PDEs

Consider the periodic initial boundary value problem

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

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

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.