Numerical Analysis Preliminary Exam
10 AM TO 1 PM, AUGUST 20, 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. **Do not write your name on your exam.** Write your student ID number.

Problem 1: Rootfinding

Iterative sequences very reminiscent of those arising when using Newtons method for root finding (either for a single equation or for a system of equations) can arise also in a number of other contexts. For exanple, if one starts with

$$a_0 = \sqrt{2} - 1, \quad b_0 = 6 - 4\sqrt{2}$$

and then iterates for $k = 0, 1, 2, \ldots$

$$a_{k+1} = \frac{1 - (1 - a_k^4)^{1/4}}{1 + (1 - a_k^4)^{1/4}}, \quad b_{k+1} = b_k(1 + a_{k+1})^4 - 2^{2k+3} a_{k+1}(1 + a_{k+1} + a_{k+1}^2)$$

it will turn out that $a_k \to 0$ and $b_k \to 1/\pi$.

(a) In the same sense as we describe typical Newton iteration convergence as quadratic, determine the convergence rate of each of the two sequences above.

(b) Give a rough estimate of how high we need k to be if we want the approximation for π to become correct to over 1,000 decimal places.

Solution: (a) We consider first a_k since it doesn't depend on b_k. Let $a_k - 0 = \epsilon$ be the error relative to the limit. Then, using the binomial theorem,

$$a_{k+1} = \frac{1 - (1 - \epsilon^4)^{1/4}}{1 + (1 - \epsilon^4)^{1/4}} = \frac{1 - (1 - \epsilon^4/4 + \mathcal{O}(\epsilon^8))}{1 + (1 - \epsilon^4/4 + \mathcal{O}(\epsilon^8))} = \frac{\epsilon^4/4 + \mathcal{O}(\epsilon^8)}{2 + \mathcal{O}(\epsilon^4)} = \frac{1}{8}\epsilon^4 + \mathcal{O}(\epsilon^8)$$

With typical Newton iterations having quadratic convergence (a leading term proportional to ϵ^2), the leading term ϵ^4 tells us that the convergence in this case has become *quartic*. The number of correct digits becomes about four times larger with each iteration.

With this information we turn to the b_k sequence. With b_k approaching a constant, the update from b_k to b_{k+1} provided by the term $b_k(1 + a_k + 1)^4$ will decrease to zero at about the same rate as a_{k+1} goes to zero. The second term $2^{2k+3} a_{k+1}(1 + a_{k+1} + a_{k+1}^2)$ will go to zero at a quartic rate as well, being proportional to a_{k+1}. We further note that its factor 2^{2k+3} is more than off-set by the factor $1/8$ in $a_{k+1} = \epsilon^4/8 + \mathcal{O}(\epsilon^8)$. Also, the b_k sequence therefore converges at a *quartic* rate.

(b) As a rough way to estimate the number of iterations needed to get 1,000 digit accuracy, we note that $b_0 \approx 1/3$, which is correct to one digit. Then quartic convergence suggests $k = 1 \Rightarrow 4$ digits, $k = 2 \Rightarrow 16$ digits, $k = 3 \Rightarrow 64$ digits, $k = 4 \Rightarrow 256$ digits, and finally $k = 5 \Rightarrow 1024$ digits.

Problem 2: Interpolation & Approximation

(a) Let the entries of $\mathbf{x} = [x_0, x_1, \ldots, x_{N-1}]^T$ be N discrete samples of a continuous function f, observed at timeponts $t_k = 2\pi k/N$, $k = 0, 1, \ldots, N-1$. What is the connection between the trigonometric interpolation of f at (t_k, x_k), $k = 0, 1, \ldots, N-1$, and the discrete Fourier transform $\mathbf{X} = F_N\mathbf{x}$, where $F_N = [f_{pq}]$ is the Vandermonde matrix with

$$\begin{aligned} f_{pq} &= \omega_N^{pq} \\ \omega_N &= e^{-2\pi i/N}\,. \end{aligned}$$

(b) Denote the DFT operator as $\mathcal{F}$, the inverse DFT operator as $\mathcal{F}^{-1}$, and a time series of data as $\boldsymbol{x}$. Assuming the existence of a software that efficiently computes a DFT, mathematically explain a way that this code can be used to efficiently compute an inverse DFT. In other words, how can you compute $\mathcal{F}^{-1}(\boldsymbol{x})$ using only the code that computes $\mathcal{F}$ and the data $\boldsymbol{x}$?

Solution: (a) The kth coefficient of the trigonometric interpolation is precisely proportional to the kth element of the discrete Fourier transform, i.e., X_k. The discrete Fourier transform of uniformly spaced points $t_k = \frac{2\pi k}{N}$ is

$$X_k = \sum_{j=0}^{N} x_j e^{-i\frac{2\pi k}{N}j}\ ;\ k = 0, 1, \ldots, N-1\,,$$

and the trigonometric interpolation of f at (t_k, x_k), $k = 0, 1, \ldots, N-1$ is

$$p_N(t) = \sum_{k=0}^{N-1} c_k e^{ijt}$$

where

$$c_k = \frac{1}{N}\sum_{j=0}^{N-1} x_k e^{-i\frac{2\pi k}{N}j}\,.$$

Thus, the relationship is

$$c_k = \frac{X_k}{N}\,.$$

Note that, depending on which book the student studies from, the $\frac{1}{N}$ normalization factor might be in front of the inverse DFT (I've even seen a book with $\frac{1}{\sqrt{N}}$). The important thing is the connection between c_k and X_k.

(b) We list three possible answers (providing any one of them will earn full credit).

1. Reverse the inputs:
$$\mathcal{F}^{-1}(\{x_n\}) = \mathcal{F}(\{x_{N-n}\})/N$$
where the indices are interpreted modulo N, i.e., $x_{N-0} = x_0$.

2. Conjugate the inputs and outputs:
$$\mathcal{F}^{-1}(\boldsymbol{x}) = \mathcal{F}(\boldsymbol{x}^*)^*/N$$

3. Swap real and imaginary parts:
$$\mathcal{F}^{-1}(\boldsymbol{x}) = \mathrm{swap}(\mathcal{F}(\mathrm{swap}(\boldsymbol{x})))/N$$
where for $a, b \in \mathbb{R}$, $\mathrm{swap}(a + ib) = b + ia$.

Problem 3: Quadrature

(a) Explain how to find weights w_i and nodes x_i such that the quadrature $\sum_{i=0}^{n} w_i f(x_i)$ gives the exact solution to $\int_0^\infty e^{-x} f(x)\mathrm{d}x$ whenever f is a polynomial of degree $\leq n$.

(b) Find nodes x_0 and x_1 and weights w_0 and w_1 such that the quadrature $\sum_{i=0}^{n} w_i f(x_i)$ gives the exact solution to $\int_0^\infty e^{-x} f(x)\mathrm{d}x$ whenever f is a polynomial of degree ≤ 1.

(c) Formulate a convergent quadrature for the integral $\int_0^1 e^x/\sqrt{x}\mathrm{d}x$.

(d) Let $I[f] = \int_a^b f(x)\mathrm{d}x$, and let $I_n[f] = \sum_{i=0}^{n} w_i f(x_i)$. Prove that if I_n integrates polynomials up to degree n exactly and the weights w_i are all positive then the quadrature is convergent for any $f \in C([a,b])$, i.e. $\lim_{n\to\infty} I_n[f] = I[f]$.

Solution: (a) This is Gauss-Laguerre quadrature. The weight function is $w(x) = e^{-x}$. The nodes x_i are the roots of the n^{th} order orthogonal polynomial associated with this weight function (which can be obtained via, e.g., Gram-Schmidt). The weights are, as usual, the integrals of the Lagrange polynomials

$$w_i = \int_0^\infty e^{-x} \ell_i(x)\mathrm{d}x.$$

(b) First find the second-order Laguerre polynomial using Gram-Schmidt.

$$\phi_0(x) = 1, \quad \phi_1(x) = x - \frac{\int_0^\infty x e^{-x}\mathrm{d}x}{\int_0^\infty e^{-x}\mathrm{d}x} = x - 1$$

$$\phi_2(x) = x^2 - \frac{\int_0^\infty x^2 e^{-x}\mathrm{d}x}{\int_0^\infty e^{-x}\mathrm{d}x} - \frac{\int_0^\infty x^2(x-1) e^{-x}\mathrm{d}x}{\int_0^\infty (x-1)^2 e^{-x}\mathrm{d}x}(x-1)$$

$$\phi_2(x) = x^2 - 2 - 4(x-1).$$

The roots are $x_0 = 2 - \sqrt{2}$ and $x_1 = 2 + \sqrt{2}$.
The weights can be obtained either by integrating the Lagrange polynomials (with the weight function) or by solving the following 2×2 linear system:

$$w_0 + w_1 = 1, \quad x_0 w_0 + x_1 w_1 = 1$$

$$w_0 = \frac{2+\sqrt{2}}{4}, \quad w_1 = \frac{2-\sqrt{2}}{4}.$$

(c) Our quadratures usually assume that the integrand is continuous on the closed interval, which is not the case here since the integrand is singular. We can either set $w(x) = 1/\sqrt{x}$ and then formulate a Gaussian quadrature, or we can simply make a change of variables. For example, letting $\sqrt{x} = t$ yields the integral

$$\int_0^1 \frac{e^x}{\sqrt{x}}\mathrm{d}x = 2\int_0^1 e^{t^2}\mathrm{d}t.$$

We can then apply any convergent quadrature to the transformed integral, e.g. trapezoid rule.

(d) Let p_* be the minimax approximation to $f(x)$ of degree $\leq n$. Note that

$$|I[f] - I_n[f]| = |I[f] - I[p_*] + I_n[p_*] - I_n[f]| \leq |I[f - p_*]| + |I_n[f - p_*]|$$

because $I_n[p_*] = I[p_*]$. Now notice that

$$|I[f - p_*]| = |\int_a^b f(x) - p_*(x)\mathrm{d}x| \leq (b - a)\rho_n(f)$$

where $\rho_n(f)$ is the minimax error. Recall that the Weierstrass approximation theorem says $\rho_n(f) \to 0$ as $n \to \infty$ for functions $f \in C([a, b])$.
Next note that $I_n[\cdot]$ is a bounded linear operator whose ∞-norm operator norm is

$$\|I_n\|_\infty = \sum_i |w_i| = \sum_i w_i = b - a.$$

We can therefore conclude

$$|I_n[f - p_*]| \leq \|I_n\|_\infty \|f - p_*\|_\infty = (b - a)\rho_n(f) \to 0 \text{ as } n \to \infty.$$

Problem 4: Numerical Linear Algebra

(a) Prove that the Gauss-Jacobi iteration is convergent whenever the coefficient matrix is strictly diagonally dominant.

(b) Formulate the Modified Gram-Schmidt algorithm to produce an orthogonal basis for the range of a matrix $\mathbf{A}$. You may assume that $\mathbf{A}$ has full column rank.

(c) Suppose that $\bar{\lambda}$ is a very good approximation (but not exact) to a simple eigenvalue λ of the matrix $\mathbf{A}$. Formulate an iterative method that will obtain a good approximation to the associated eigenvector after a very small number of iterations.

Solution:

(a) Let $\mathbf{A} = \mathbf{D} - \mathbf{E} - \mathbf{F}$ where $-\mathbf{E}$ and $-\mathbf{F}$ are the lower- and upper-triangular parts of $\mathbf{A}$. The Gauss-Jacobi iteration is

$$\boldsymbol{x}_{k+1} = \mathbf{D}^{-1}(\mathbf{E} + \mathbf{F})\boldsymbol{x}_k + \mathbf{D}^{-1}\boldsymbol{b}.$$

The iteration matrix is $\mathbf{B} = \mathbf{D}^{-1}(\mathbf{E} + \mathbf{F})$ and we know that the iteration will be convergent when $\|\mathbf{B}\| \leq 1$ (sufficient condition). If $\mathbf{A}$ is strictly diagonally dominant by rows, then the ∞-norm of $\mathbf{B}$ is less than 1 because diagonal dominance implies that each absolute row sum of $\mathbf{B}$ is less than 1. The proof for column-wise diagonal dominance is similar, but uses the 1-norm of $\mathbf{B}$.

(b) Let $\boldsymbol{a}_i$ be the i^{th} column of $\mathbf{A}$. The MGS algorithm iteratively produces orthonormal vectors $\boldsymbol{q}_i$ as follows: Set $\boldsymbol{a}_i^{(1)} = \boldsymbol{a}_i$ for all i.

- For $k = 1, \ldots, n$ do

$$\boldsymbol{q}_k = \frac{\boldsymbol{a}_k^{(k)}}{\|\boldsymbol{a}_k^{(k)}\|}$$

  - For $j = k+1, \ldots, n$ do

$$\boldsymbol{a}_j^{(k+1)} = \boldsymbol{a}_j^{(k)} - (\boldsymbol{q}_k \cdot \boldsymbol{a}_j^{(k)})\boldsymbol{q}_k$$

The basic idea is that once a $\boldsymbol{q}_k$ vector is available, it is projected out of all the remaining columns of $\mathbf{A}$. In classical Gram-Schmidt, by contrast, at each $\boldsymbol{a}_i$ one projects out all of the $\boldsymbol{q}_k$ for $k = 1, \ldots, i$. The algorithms are equivalent in exact arithmetic, but MGS is more numerically stable.

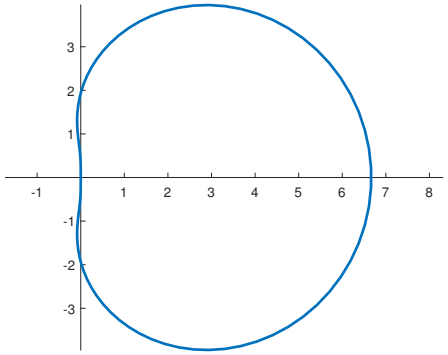
(c) The matrix $\mathbf{A} - \bar{\lambda}\mathbf{I}$ has a simple eigenvalue very close to 0 (closer than all others, if $\bar{\lambda}$ is a very good approximation to λ). We can find this eigenvalue and the associated eigenvector by performing an inverse power iteration on the matrix $\mathbf{A} - \bar{\lambda}\mathbf{I}$, which is the same as a shifted inverse power iteration on $\mathbf{A}$. The rate of convergence depends on how close $\lambda - \bar{\lambda}$ is to zero, as compared to the other eigenvalues; by assumption $\lambda - \bar{\lambda}$ is extremely close to zero so the iteration should converge extremely quickly. The iteration is

$$(\mathbf{A} - \bar{\lambda}\mathbf{I})\tilde{\boldsymbol{z}}_{k+1} = \boldsymbol{z}_k, \quad \boldsymbol{z}_{k+1} = \frac{\tilde{\boldsymbol{z}}_{k+1}}{\|\tilde{\boldsymbol{z}}_{k+1}\|}.$$

Problem 5: ODEs

The following matlab code produces the following figure

```
r = exp(complex(0,linspace(0,2*pi)));
xi = (11/6*r.^3-3*r.^2+3/2*r-1/3)./r.^3;
plot(xi,'LineWidth',2)
axis equal; ax = gca; box off;
ax.XAxisLocation = 'origin';
ax.YAxisLocation = 'origin';
```



This figure displays the boundary of the stability domain for a certain consistent linear multistep method (LMM).

(a) Write down the formula for this LMM in the conventional form of coefficients for $y(t)$ and $y'(t) = f(t)$ at a sequence of equispaced t levels. Does this scheme go under a well-known name?

(b) Determine if the stability domain is given by the inside or the outside (or neither) of the shown curve.

(c) Determine if the scheme satisfies the root condition.

Solution: (a) After re-writing the equation for ξ as

$$r^3\left(\xi - \frac{11}{6}\right) + 3r^2 - \frac{3}{2}r + \frac{1}{3} = 0 \tag{1}$$

with $\xi = \lambda k$ ("xi" in the code), we can directly read off the scheme as

$$y'(t+k) = \frac{1}{k}\left(\frac{11}{6}y(t+k) - 3y(t) + \frac{3}{2}y(t-k) - \frac{1}{3}y(t-2k)\right)$$

(since applying the scheme to $y' = \lambda y$ leads to a linear recursion with (1) as its characteristic equation). This is a *backwards differentiation (BD) scheme.* you are not asked to check its order, but it is the standard third order accurate scheme of this kind, sometimes abbreviated as BD3.

(b) By general theory, immediately to the right of the origin is always outside the stability domain. For values of $\xi = \lambda k$ very large in magnitude, all roots $r_{1,2,3}$ to (1) will clearly be forced in towards $r = 0$, so the stability domain must be the outside of the curve.

(c) The equation to check for the root condition is (1) after we have set $\xi = 0$, i.e.

$$-\frac{11}{6}r^3 + 3r^2 - \frac{3}{2}r + \frac{1}{3} = 0$$

which we write as

$$r^3 - \frac{18}{11}r^2 + \frac{9}{11}r - \frac{2}{11} = 0.$$

For any consistent scheme $r = 1$ has to be a root. Dividing this away leaves

$$r^2 - \frac{7}{11}r + \frac{2}{11} = 0$$

with two roots

$$r_\pm = \frac{7}{22} \pm \frac{\sqrt{39}}{22}\mathrm{i}.$$

Since these form a complex conjugate pair, their magnitudes are equal. We also see from the constant term of the quadratic equation that the product of the roots is $2/11$, which (in magnitude) is less than one. Thus, both of these two roots are inside the unit circle, and the root condition is satisfied.

Problem 6: PDEs

Consider the partial differential equation defined in the (t, x)-domain

$$\begin{aligned} u_{tt} &= 2u_{xx} \\ -1 \leq\ & x \leq 1 \\ 0 <\ & t \end{aligned}$$

with boundary and initial conditions

$$\begin{aligned} u(t,-1) = u(t,1) = 0 \quad &; \quad t > 0 \\ u(0,x) = e^{-x^2} - e^{-1} \quad &; \quad -1 \leq x \leq 1 \\ u_t(0,x) = (x+1)(x-1) \quad &; \quad -1 \leq x \leq 1 \end{aligned}$$

and spatial discretization with stepsize h and time discretization with stepsize k.

(a) Create an explicit $\mathcal{O}(h^2 + k^2)$ finite difference approximation to the solution.

(b) How would one accurately compute the solution at the first timepoint $t_1 = k$?

(c) How would one choose the sizes of h and k sufficient to maintain stability?

Solution: (a) A finite difference approximation of this order would be created by using centered differences in space and time. That is, at gridpoint (t_i, x_j), the PDE would discretize to

$$\frac{u(t_{i+1},x_j) - 2u(t_i,x_j) + u(t_{i-1},x_j)}{k^2} = 2\frac{u(t_i,x_{j+1}) - 2u(t_i,x_j) + u(t_i,x_{j-1})}{h^2}\,.$$

Answer must communicate (using a Taylor expansion) that centered differences are 2nd order for full credit.

(b) At the first timepoint, there is a problem in that the central difference discretization requires the value of the solution at two previous timepoints. At $t_1 = k$, the only information to draw upon is the value of the solution on the $t = 0$ boundary. Thus, the discretization is altered to include derivative information on the initial boundary. The left side of the discretization is altered in the following manner:

$$\begin{aligned} \frac{u(t_1,x_j) - 2u(t_0,x_j) + u(t_{-1},x_j)}{k^2} &= \frac{u(t_1,x_j) - u(t_0,x_j)}{k^2} - \frac{1}{k}\left(\frac{u(t_0,x_j) - u(t_{-1},x_j)}{k}\right) \\ &\approx \frac{u(t_1,x_j) - u(t_0,x_j)}{k^2} - \frac{1}{k}u_t(t_0,x_j) \\ &\approx \frac{u(t_1,x_j) - u(t_0,x_j)}{k^2} - \frac{1}{k}(x_j+1)(x_j-1) \end{aligned}$$

and thus the computation at t_1 becomes

$$u(t_1,x_j) = u(t_0,x_j) + k^2\left(\frac{1}{k}(x_j+1)(x_j-1) + 2\frac{u(t_0,x_{j+1}) - 2u(t_0,x_j) + u(t_0,x_{j-1})}{h^2}\right)\,.$$

(c) The CFL condition states that

$$\frac{k}{h} < \frac{1}{\sqrt{2}}$$

must be satisfied to maintain stability. Satisfying this condition ensures that the domain of dependence for the PDE is within the domain of dependence of the explicit finite difference scheme. However, the CFL condition is only necessary and not sufficient.

For full credit, an answer must illustrate a von Neumann analysis. which yields a form for the approximate solution of

$$r^{t/k} e^{i\omega x} ,$$

and leading (after some straightforward algebra) to the characteristic equation

$$r^2 + [4\frac{k^2}{h^2}(1 - \cos(\omega h) - 2]r + 1 = 0 .$$

The condition to keep $|r| < 1$ is then

$$\frac{k}{h} < \frac{1}{\sqrt{2}} ,$$

which is (coincidentally) the CFL condition.