Department of Applied Mathematics

Preliminary Examination in Numerical Analysis

August 17, 2016 , 10 am – 1 pm.

Submit solutions to four (and no more) of the following six problems. Show all your work, and justify all your answers. No calculators allowed.

Problems and solutions given below:

----

1. Root finding / Nonlinear equations

Consider the scalar equation $F(x)=0$. Assume α is a root of the equation.

a. Give the recursion for the Newton method for approximating a root.

b. Give conditions on $F(x)$ near α that guarantee convergence for x_0 sufficiently close to α.

c. Consider the following Taylor expansion: $F(\alpha)=F(x)+F'(x)(\alpha-x)+\frac{1}{2}F''(w)(\alpha-x)^2$, where $w=\beta x+(1-\beta)\alpha$ for some $\beta\in[0,1]$. Using the Taylor expansion, derive a relationship between the error at step $j+1$ in terms of the error at step j.

d. Show how the conditions stated in part (b), together with the Taylor expansion above, can be used to bound the error at step $j+1$ in terms of the error in step j.

e. Finally, show how the development above can be used to establish convergence. With what order does the iteration converge?

Solution: Nonlinear Equations

(a) Newton:

$$x_{j+1} = x_j - \frac{F(x_j)}{F'(x_j)}. \tag{1}$$

(b). Assume $F(x), F'(x), F''(x)$ are all continuous in some neighborhood of α; assume $F(\alpha) = 0$, $F'(\alpha) \neq 0$. Then, if x_0 is sufficiently close to α, the iterates x_j, for $j > 0$, will converge to α.

(c) Note that $F(\alpha) = 0$ and solve for α to get

$$\alpha = x - \frac{F(x)}{F'(x)} - \frac{1}{2}\frac{F''(w)}{F'(x)}(\alpha - x)^2.$$

Let $x = x_j$ and use the recursion for Newton's method to get

$$\alpha - x_{j+1} = -\frac{1}{2}\frac{F''(w)}{F'(x)}(\alpha - x_j)^2.$$

(d) By continuity, we may assume $F'(x) \neq 0$ in a neighborhood $\mathcal{N} = (\alpha - \eta, \alpha + \eta)$. Let

$$M = \frac{\max_{x\in\mathcal{N}} F''(x)}{2\min_{x\in\mathcal{N}} F'(x)}.$$

If x_0 is sufficiently close to α, we have the bound

$$|\alpha - x_{j+1}| \leq M|\alpha - x_j|^2$$

(e) If $M|\alpha - x_j| < 1$, then

$$|\alpha - x_{j+1}| < |\alpha - x_j|$$

If x_0 is chosen so that $|\alpha - x_0| < \eta$ and $M|\alpha - x_0| < 1$, then

$$M|\alpha - x_1| \leq (M|\alpha - x_0|)^2,$$

which implies $M|\alpha - x_1| < M|\alpha - x_0|$, and, thus, $x_1 \in \mathcal{N}$. By induction, this also implies $x_j \in \mathcal{N}$ for $j > 1$ and

$$M|\alpha - x_j| \leq (M|\alpha - x_{j-1}|)^2 \leq (M|\alpha - x_0|)^{2^j}.$$

Thus, $x_j \to \alpha$ and convergence is quadratic.

2. Quadrature

Determine the nodes and the weights in the 2-node Gaussian quadrature formula

$$\int_0^\infty f(x)\, e^{-x} dx = w_1 f(x_1) + w_2 f(x_2).$$

Solution:

One approach is to find the first three orthogonal polynomials $\varphi_i(x),\ i = 0,1,2$ for e^{-x} over $[0,\infty]$ and then read off the nodes as the two zeros of $\varphi_2(x)$, after which the weights follow by requiring the exact result when applied to $f(x) \equiv 1$ and $f(x) \equiv x.$ Alternatively, we can start by writing down the nonlinear system that arises from enforcing the exact result for the functions $1, x, x^2, x^3$:

$$w_1 + w_2 = \int_0^\infty (1)\, e^{-x} dx = 1 \qquad (1)$$

$$w_1 x_1 + w_2 x_2 = \int_0^\infty (x)\, e^{-x} dx = 1 \qquad (2)$$

$$w_1 x_1^2 + w_2 x_2^2 = \int_0^\infty (x^2)\, e^{-x} dx = 2 \qquad (3)$$

$$w_1 x_1^3 + w_2 x_2^3 = \int_0^\infty (x^3)\, e^{-x} dx = 6 \qquad (4)$$

This type of nonlinear systems have a special structure that readily allows them to be solved. Let $p(x) = c_0 + c_1 x + 1 \cdot x^2$ be the quadratic polynomial that has x_1 and x_2 as its roots. We next form

$$c_0 \cdot \{eq.(1)\} + c_1 \cdot \{eq.(2)\} + 1 \cdot \{eq.(3)\} \quad \Rightarrow \quad 0 = c_0 + c_1 + 2$$

$$c_0 \cdot \{eq.(2)\} + c_1 \cdot \{eq.(3)\} + 1 \cdot \{eq.(4)\} \quad \Rightarrow \quad 0 = c_0 + 2c_2 + 6$$

This system has the solution $c_0 = 2$, $c_1 = 4$, i.e. $p(x) = x^2 - 4x + 2$ with roots (nodes) $x_{1,2} = 2 \mp \sqrt{2}.$

To get the weights, it suffices to require the exact result for $f(x) \equiv 1$ and $f(x) \equiv x.$ This leads to the linear system $\begin{bmatrix} 1 & 1 \\ 2-\sqrt{2} & 2+\sqrt{2} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$ with the solution $w_{1,2} = \dfrac{1}{2} \pm \dfrac{\sqrt{2}}{4}.$

<u>3.</u> <u>Interpolation / Approximation</u>

a. Define what is meant by *cubic splines* and, for these, *natural* and *not-a-knot* conditions.

b. Determine the *not-a-knot* cubic spline $s(x)$ that satisfies the data $\begin{array}{c|cccc} x & -1 & 0 & 1 & 2 \\ \hline y & -2 & -3 & -4 & 1 \end{array}$.

c. If, at the nodes $x = -h, 0, h$, one has function values y_{-h}, y_0, y_h and forms a quadratic interpolant $s(x)$, one obtains $s'(0) = [-\frac{1}{2} y_{-h} + \frac{1}{2} y_h]/h$, i.e. the finite difference weights can be written as $[-\frac{1}{2}, 0, +\frac{1}{2}]/h.$ It might be tempting to replace the quadratic interpolant here with a natural cubic spline (hoping to increase the approximation's order of accuracy). Work out the weights you get in this case.

<u>Solution:</u>

a. A *cubic spline* is a cubic polynomial between adjacent nodes, and features continuous function, first and second derivative at the nodes – i.e. the third derivative may be discontinuous at the nodes. Without additional end conditions, a cubic spline will have two free parameters. A *natural* cubic spline adds the two extra conditions that $s''(x) = 0$ at each end point. The *not-a-knot* cubic spline instead removes two 'freedoms', i.e. the cubic spline is not allowed to have a jump in its third derivative one node point in from each boundary.

b. With four node points, and jumps in the third derivative not allowed at either of the two internal nodes, the spline becomes a single cubic, i.e. we can immediately find it, for ex., by Lagrange's or Newton's interpolation formulas. Choosing, for ex., the Newton approach, the divided difference table becomes

-1	-2			
		-1		
0	-3		0	
		-1		1
1	-4		3	
		5		
2	1			

from which we read off the polynomial as $s(x) = -2 - 1\cdot(x+1) + 0\cdot(x+1)x + 1\cdot(x-1)x(x+1) =$ $x^3 - 2x - 3.$

c. Since the spline $s(x)$ is not discontinuous at $x = 0$ until in the third derivative, we can write it:

$$[-h, 0] \qquad a + bx + cx^2 + dx^3$$
$$[0, +h] \qquad a + bx + cx^2 + ex^3$$

The natural end conditions give $2c - 6dh = 0$ and $2c + 6eh = 0$, resp., i.e. $e = -d.$ Enforcing the values at the nodes now give

$$\begin{cases} a - bh + ch^2 - dh^3 & = y_{-h} \\ a & = y_0 \\ a + bh + ch^2 - dh^3 & = y_h \end{cases}$$

Subtracting the top equation from the bottom one gives $2bh = y_h - y_h,$ and we obtain the same approximation for $s'(0)$ as before.

4. Linear Algebra

Consider the linear system $A\underline{x} = \underline{b}$, where $A_{n\times m}, \underline{x}_{m\times 1}, \underline{b}_{n\times 1}$.

a. Describe the three possible cases for existence and uniqueness of a solution of the linear system. Give criteria on $A, \underline{b}$ that distinguish each case.

b. Let $\underline{x}_{LS}$ be a minimizer of the least squares functional, that is, let

$$\| A\underline{x}_{LS} - \underline{b} \|_2 = \min_{\underline{x}} \| A\underline{x} - \underline{b} \|_2 .$$

(i) Does $\underline{x}_{LS}$ always exist? Explain your answer.

(ii) Give conditions on $A, \underline{b}$ such that $\underline{x}_{LS}$ is unique.

(iii) In the case of a unique solution, give an expression for the least squares solution $\underline{x}_{LS}$.

(iv) If there is an infinite number of solutions to the least squares problem, find the solution of minimal norm.

c. The minimal norm solution can be computed by using the singular value decomposition (SVD) of A. Define the singular value decomposition and show how it can be used to compute the minimal norm least squares solution.

Solution: Linear Algebra

(a) The three cases are:

There exists a unique solution if:

(i) The columns of A are linearly independent (or any equivalent statement).

(ii) $\underline{b} \in$ Range A (equivalently, $\underline{b}$ is orthogonal to the null space of A^*).

There exists an infinite number of solutions if:

(i) The columns of A are linearly dependent (or any equivalent statement).

(ii) $\underline{b} \in$ Range A (equivalently, $\underline{b}$ is orthogonal to the null space of A^*).

There exists no solution if:

(i) $\underline{b} \notin$ Range of A.

(b)

(i) There is always at least one minimizer of the quadratic functional. To see this, take the gradient of the quadratic functional and set it equal to zero:

$$A^* A\underline{x} - A^*\underline{b} = 0.$$

The question is if $A^*\underline{b}$ is always in the range of A^*A. Suppose $\underline{v} \in$ Null Space of $A^*A =$ Null Space of A. Consider

$$\langle A^*\underline{b}, \underline{v}\rangle = \langle \underline{b}, A\underline{v}\rangle = 0.$$

Thus, $A^*\underline{b}$ is orthogonal to the Null Space of A^*A, which implies it is in the Range of A^*A.

(ii) The following equivalent conditions yield a unique least squares solution

(a) The columns of A are linearly independent.

(b) The square system A^*A is nonsingular.

(iii) If (A^*A) is nonsingular, then the unique least squares solution is given by

$$\underline{x}_{LS} = (A^*A)^{-1}A^*\underline{b}.$$

(iv) The minimal length solution is given by

$$\underline{x}_{min} = (A^*A)^{\dagger}A^*\underline{b} = A^{\dagger}\underline{b}.$$

where $(A^*A)^{\dagger}$ is the pseudo-inverse of (A^*A) ($A^{\dagger}$ is the pseudo-invers of A).

(c) The pseudo-inverse can be computed by using the singular value decomposition (SVD) of A,

$$A = U\Sigma V^*,$$

where $U_{n\times n}, V_{m\times m}$ are unitary and $\Sigma_{n\times m}$ is diagonal:

$$\Sigma_{n\times m} = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ 0 & 0 & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots \end{bmatrix}$$

and σ_j are the singular values. Then,

$$(A^*A)^\dagger A^* = A^\dagger = V\Sigma^\dagger U^*,$$

where

$$\Sigma^\dagger_{m\times n} = \text{diag}\{..., \sigma_j^\dagger, ...\}$$

and

$$\sigma_j^\dagger = \begin{cases} \frac{1}{\sigma_j} & \sigma_j \neq 0, \\ 0 & \sigma_j = 0. \end{cases}$$

---

5. Numerical ODE

The following are two different linear multistep methods for solving the ODE $y'(t) = f(t, y(t))$:

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

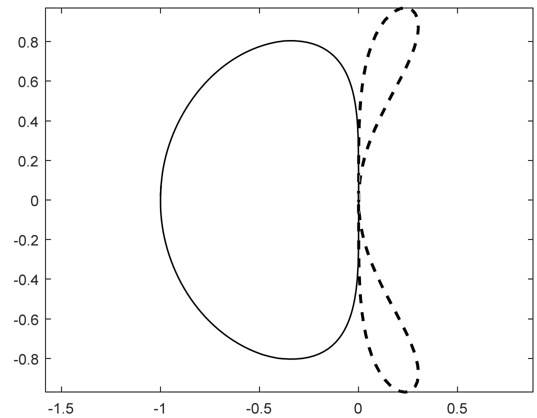
(ii) $$y_{n+1} = y_{n-1} + \frac{k}{3}(7f(t_n, y_n) - 2f(t_{n-1}, y_{-1}) + f(t_{n-2}, y_{n-2}))$$

In order to assess the basic properties of these two schemes, we run the Matlab code

```
r = exp(pi*2i*linspace(0,1));
plot(r.*(r-1)./((3*r-1)/2),'k-'); hold on
plot((r.^3-r)./((7*r.^2-2*r+1)/3),'k--');
axis equal
xi = 0.2+0.8i; roots([1,-7*xi/3,2*xi/3-1,-xi/3])
```

and obtain the plot shown to the right, together with the output

```
ans =
  -0.3082 + 1.1297i
   0.7431 + 0.9314i
   0.0318 - 0.1944i
```



For each of the two schemes, determine

a. Will the schemes convergence to the ODE solution in the limit of $k \to 0$, or diverge?
b. What is their formal order of accuracy?
c. Identify their stability domains,
d. Do the schemes feature A-stability?

Solution:

(a) The requirements are consistency and root condition. Consistency follows from part (b), where we show the order to be one or higher. For the root condition, apply the methods to the ODE $y' = 0$ and check the characteristic equation:

(i) $r - 1 = 0 \quad \Rightarrow r = 1$ root condition satisfied $\Rightarrow$ convergence

(ii) $r^2 - 1 = 0 \Rightarrow r = \pm 1$ root condition satisfied $\Rightarrow$ convergence

(b) Apply the scheme, at $t = 0$, to the functions $1, t, t^2, \ldots$ and see how far the approximations remain exact:

(i) $[1]: 1 = 1 + 0$, OK,

$[t]: k = 0 + \frac{k}{2}[3 - 1]$, OK,

$[t^2]: k^2 = 0 + \frac{k}{2}[0 + 2k]$, OK,

$[t^3]: k^3 = 0 + \frac{k}{2}[0 - 3k^2]$, Fail; Hence, Second order.

(ii) $[1]: 1 = 1 + 0$, OK,

$[t]: k = -k + \frac{k}{3}[7 - 2 + 1]$, OK,

$[t^2]: k^2 = k^2 + \frac{k}{3}(0 + 4k - 4k)$, OK,

$[t^3]: k^3 = -k^3 + \frac{k}{3}(0 - 6k^2 + 12k^2),$ OK,

$[t^4]: k^4 = k^4 + \frac{k}{3}(0 + 8k^3 - 32k^3),$ Fail; Hence, Third order.

(c) When applied to $y' = \lambda y,$ the stability domains are the regions in the complex $\xi = \lambda k$ plane such that all roots to the characteristic equation, are inside (or on) the unit circle.

(i) The polynomial to inspect becomes $r^2 + (\frac{3}{2}\xi - 1)r - \frac{1}{2}\xi = 0.$ The solid curve in the figure shows where a root crosses the unit circle, so it suffices to inspect one point outside and one point inside this curve.
Outside: Consider some ξ that is massively large. Since the products of the roots equals the constant term of the quadratic, $r_1 \cdot r_2 = -\frac{1}{2}\xi.$ Hence, at least one root is outside the unit circle.
Inside: Take for ex. $\xi = \frac{1}{2}.$ Solving the quadratic gives $r_{1,2} = \frac{1}{8}[1 \pm \sqrt{17}] \approx \frac{1}{8}[1 \pm 4.1],$ i.e. both roots are well inside the unit circle. This inside of the curve in the figure therefore shows the stability domain.

(ii) The polynomial to inspect becomes $r^3 - \frac{7}{3}\xi r^2 + (\frac{2}{3}\xi - 1)r - \frac{1}{3}\xi = 0.$ The dashed curve shows where a root crosses the unit circle.
Outside: For the same reason as above, this is outside the stability domain.
Inside: The numerical root evaluation in the program shows two roots outside the unit circle. This ODE solver therefore has the unusual property of altogether lacking a stability domain (with the exception of the single point $\xi = 0$).
Note: The plotted (dashed) curve shows where a root *r* crosses the unit circle. However, for this equation, there turns out to be another root that is outside, so the curve is irrelevant as far as the stability domain is concerned. *All roots* must be inside the unit circle for the $\xi -$ value to correspond to the inside of the stability domain.

(d) Neither scheme is A-stable (as this requires the complete left half-plane to fall within the methods stability domain).

6. Numerical PDE

Consider the parabolic equation

$$\frac{\partial u}{\partial t} = a\frac{\partial^2 u}{\partial x^2} + f$$

where *a* is a constant.

a. Give the formula for the following finite difference approximations.
   (i) Forward Euler: Centered differences in space, forward difference in time.
   (ii) Backward Euler: Centered differences in space, backward difference in time.
   (iii) Leapfrog: Centered difference in space and centered difference in time.

b. What is the order of accuracy of each method?

c. Use a von Neumann analysis (or any appropriate analysis) to determine the stability of each method.

Solution: PDEs

(a),(b) Let $u(x_j, t_k)$ be approximated by u_j^k. The schemes are as follows.

Forward Euler:

$$u_j^k = u_j^{k-1} + \frac{a\delta t}{\delta x^2}\left[u_{j-1}^{k-1} - 2u_j^{k-1} + u_{j+1}^{k-1}\right] + \delta t f_j^k,$$

with truncation error

$$O(\delta t) + O(\delta x^2).$$

Backward Euler:

$$u_j^k - \frac{a\delta t}{\delta x^2}\left[u_{j-1}^{k} - 2u_j^{k} + u_{j+1}^{k}\right] = u_j^{k-1} + \delta t f_j^k,$$

with truncation error

$$O(\delta t) + O(\delta x^2).$$

Leapfrog:

$$u_j^k = u_j^{k-2} + \frac{2a\delta t}{\delta x^2}\left[u_{j-1}^{k-1} - 2u_j^{k-1} + u_{j+1}^{k-1}\right] + 2\delta t f_j^k,$$

with truncation error

$$O(\delta t^2) + O(\delta x^2).$$

(c) Let $u_j^k = \lambda^k e^{ij\theta}$ and plug into each scheme with $f = 0$.

Forward Euler:

$$\lambda^k e^{ij\theta} = \lambda^{k-1} e^{ij\theta} + \lambda^{k-1}\frac{a\delta t}{\delta x^2}\left[e^{i(j-1)}\theta - 2e^{ij\theta} + e^{i(j+1)\theta}\right] a^{(k-1)}.$$

Divide by $\lambda^{k-1} e^{ij\theta}$ to get

$$\lambda = 1 + \frac{2a\delta t}{\delta x^2}(\cos(i\theta) - 1).$$

We see that $|\lambda| \leq 1.0$ for all values of θ if

$$\frac{a\delta t}{\delta x^2} \leq 1/2.$$

Forward Euler is stable for $\delta t \leq \frac{\delta x^2}{2a}$.

Backward Euler Repeating the process we obtain

$$\lambda^k e^{ij\theta} - \frac{a\delta t}{\delta x^2}\left[\lambda^k e^{i(j-1)\theta} - 2\lambda^k e^{ij\theta} + \lambda^k e^{i(j+1)\theta}\right] = \lambda^{k-1} e^{ij\theta}$$

Again, dividing by $\lambda^{k-1}e^{ij\theta}$ yields

$$\lambda(1 - 2\frac{a\delta t}{\delta x^2}(\cos(i\theta) - 1)) = 1,$$

or

$$\lambda = \frac{1}{1 + 2\frac{a\delta t}{\delta x^2}(1 - \cos(i\theta))} \leq 1.$$

Thus, Backward Euler is stable for all δt and δx, that is, it is *unconditionally stable*.

Leapfrog In this case, we have

$$\lambda^{k+1} e^{ij\theta} = \lambda^{k-1} e^{ij\theta} + \frac{a\delta t}{\delta x^2}\left[\lambda^k e^{i(j-1)\theta} - 2\lambda^k e^{ij\theta} + \lambda^k e^{i(j+1)\theta}\right]$$

Dividing by $\lambda^{k-1}e^{ij\theta}$ yields

$$\lambda^2 = 1 + \lambda(2a\frac{\delta t}{\delta x^2}(\cos(i\theta) - 1)).$$

or

$$\lambda^2 + \lambda(\frac{2a\delta t}{\delta x^2}(1 - \cos(i\theta)) - 1 = 0$$

Setting

$$B = (\frac{2a\delta t}{\delta x^2}(1 - \cos(i\theta)) \geq 0,$$

we see

$$\lambda = \frac{-B \pm \sqrt{B^2 + 4}}{2}$$

The negative root yields

$$|\lambda| > 1.0$$

for some θ for every choice of δt and δx. Thus, the Leapfrog method is *unconditioally unstable*.