Numerical Analysis PhD Qualifying Exam
University of Vermont, Winter 2011

1. (a) Given an initial guess x_0, derive Newton's method to find a better guess x_1 for approximating the root of a function $f(x)$. (b) Apply Newton's method to the function $f(x) = 1/x$ using an initial guess of $x_0 = 1$ and find a (simple) analytical expression for x_{50}.

   Solution:

   (a) Newton's method suggests we find root of the line tangent to $f(x)$ at the point x_0, and use this root as our new guess. For an initial guess of x_0, we're looking for a line through the point $(x_0, f(x_0)$ with slope $f'(x_0)$. The point-slope form for this line is $y - f(x_0) = f'(x_0)(x - x_0)$. Substituting $y = 0$ into this line, we find

$$\begin{aligned} f'(x_0)(x - x_0) &= 0 - f(x) \\ x - x_0 &= -\frac{f(x_0)}{f'(x_0)} \\ x &= x_0 - \frac{f(x_0)}{f'(x_0)} \end{aligned}$$

   We label this better guess x for the root of $f(x)$ by x_1 and iterate.

   (b)

$$\begin{aligned} x_{i+1} &= x_i - \frac{f(x_i)}{f'(x_i)} \\ &= x_i - \frac{\frac{1}{x}}{-\frac{1}{x^2}} = 2x_i \end{aligned}$$

   Given an initial guess of $x_0 = 1$, we find $x_{50} = 2^{50}$.

   □

2. Solve the following linear system with naive Gaussian elimination (i.e. without partial pivoting)

$$\begin{bmatrix} \frac{\epsilon_{mach}}{10} & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

   using (1) infinite precision and (2) a computer whose machine epsilon is given by ϵ_{mach}. Label your solutions $\vec{x}_{true}$ and $\vec{x}_{comp}$ respectively. Why is there such large difference between the two? **Note** that the first pivot $\frac{\epsilon_{mach}}{10}$ is much larger than the smallest number the computer can represent.

   Solution:

   (1) infinite precision

$$\begin{bmatrix} \frac{\epsilon_{mach}}{10} & 1 \\ 0 & 1 - \frac{10}{\epsilon_{mach}} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -\frac{10}{\epsilon_{mach}} \end{bmatrix}$$

Backward substitution gives

$$\vec{x} = \begin{bmatrix} -\frac{10}{10-\epsilon_{mach}} \\ \frac{10}{10-\epsilon_{mach}} \end{bmatrix}$$

This is the correct answer, i.e. $\vec{x}_{true}$, which is approximately $[-1, 1]^\top$.
(2) On a double precision computer, the system reduces to the solution of

$$\begin{bmatrix} \frac{\epsilon_{mach}}{10} & 1 \\ 0 & -\frac{10}{\epsilon_{mach}} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -\frac{10}{\epsilon_{mach}} \end{bmatrix}$$

which has the solution $\vec{x}_{comp} = [0, 1]^\top$.

The difference $|\vec{x}_{true} - \vec{x}_{comp}|$ is quite large, $O(10^0)$. The reason is as follows. Assuming we're using a double precision computer, where $\epsilon_{mach} = O(10^{-16})$, the first pivot is $O(10^{-17})$. As a result, the multiplier used in naive Gaussian elimination is $O(10^{17})$ leading to swamping.

The difference occurs during back substitution. Letting $\alpha = \frac{10}{10-\epsilon_{mach}}$, back substitution in part (1) produces an equation for x_1 of the form

$$\frac{\epsilon_{mach}}{10} x_1 + \alpha = 1$$

whose solution involves calculating $1 - \alpha$. This calculation suffers from catastrophic cancellation in part (2).

□

3. Apply Gram-Schmidt to find a QR-factorization of the matrix.

$$A = \begin{bmatrix} 2 & 3 \\ -2 & -6 \\ 1 & 0 \end{bmatrix}$$

Solution:

Take $y_1 = \begin{bmatrix} 2 \\ -2 \\ 1 \end{bmatrix}$. Then $r_{11} = \|y_1\|_2 = 3$ and $q_1 = \dfrac{y_1}{r_{11}} = \begin{bmatrix} 2/3 \\ -2/3 \\ 1/3 \end{bmatrix}$. Then
$y_2 = v_2 - q_1(q_1^T \cdot v_2) = \begin{bmatrix} 3 \\ -6 \\ 0 \end{bmatrix} - \begin{bmatrix} 2/3 \\ -2/3 \\ 1/3 \end{bmatrix} (6) = \begin{bmatrix} -1 \\ -2 \\ -2 \end{bmatrix}$. $r_{12} = (q_1^T \cdot v_2) = 6$ and

$r_{22} = \|y_2\|_2 = 3$. So $q_2 = \begin{bmatrix} -1/3 \\ -2/3 \\ -2/3 \end{bmatrix}$. If we stop here, we have

$$A = \begin{bmatrix} 2/3 & -1/3 \\ -2/3 & -2/3 \\ 1/3 & -2/3 \end{bmatrix} \begin{bmatrix} 3 & 6 \\ 0 & 3 \end{bmatrix}.$$

If we wish to use QR for least squares, then we continue in this manner with an arbitrarily chosen $v_3 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ to generate q_3. A complete QR-factorization of A is

$A = \begin{bmatrix} 2/3 & -1/3 & 2/3 \\ -2/3 & -2/3 & 1/3 \\ 1/3 & -2/3 & -2/3 \end{bmatrix} \begin{bmatrix} 3 & 6 \\ 0 & 3 \\ 0 & 0 \end{bmatrix}$. Of course, there are several other QR decompositions up to a negative sign, e.g. negative entries in column 2 can be switched to positive provided $R_{22} = -3$.

□

4. Given an IVP $y' = f(t, y)$, methods for numerical integration are distinguished by their approximation of the integral in the formula $y(t+h) = y(t) + \int_{t_i}^{t_{i+1}} f(t, y)dt$. Derive the degree-2 Adam's Bashforth method (AB2) given by $w_{i+1} = w_i + \frac{h}{2}(3f_i - f_{i-1})$ in two steps:

   (1) Approximate $f(t, y)$ with a polynomial $P_n(t)$ of degree n interpolating the $n+1$ points $(t_{i-n}, f_{i-n}), ..., (t_{i-1}, f_{i-1}), (t_i, f_i)$. **Note** that you should determine the degree n based on the specified order of accuracy of AB2. It may help to label the constant step-size in time $h = t_i - t_{i-1}$.

   (2) Evaluate the integral $\int_{t_i}^{t_{i+1}} P_n(t)dt$

   Solution:

   The degree 2 Adams' Bashforth method requires a polynomial of degree 1. First, we use the Newton form of the interpolating polynomial to find $P_1(t)$, namely

$$\begin{aligned} P_1(t) &= f_{i-1} + \frac{f_i - f_{i-1}}{t_i - t_{i-1}}\Big(t - t_{i-1}\Big) \\ &= f_{i-1} + \frac{f_i - f_{i-1}}{h}\Big(t - t_{i-1}\Big) \end{aligned}$$

   Then we integrate

$$
\begin{aligned}
\int_{t_i}^{t_{i+1}} P_1(t)dt &= f_{i-1}h + \frac{(f_i - f_{i-1})\left[(t_{i+1} - t_{i-1})^2 - (t_i - t_{i-1})^2\right]}{2h} \\
&= f_{i-1}h + \frac{(f_i - f_{i-1})\left[(2h)^2 - h^2\right]}{2h} \\
&= f_{i-1}h + \frac{(f_i - f_{i-1})3h}{2} \\
w_{i+1} &= w_i + h\frac{3f_i - f_{i-1}}{2}
\end{aligned}
$$

□

5. Method

$$Y_{n+1} - Y_{n-1} \ = \ \frac{h}{8}\,(5f_{n+1} + 6f_n + 5f_{n-1}), \qquad \text{where} \quad f_n \equiv f(x_n, Y_n), \text{ etc.} \tag{1}$$

can be used to solve the initial-value problem

$$y' = f(x,y), \qquad y(x_0) = y_0. \tag{2}$$

Using the equation $y' = -\lambda y$ as a model problem ($\lambda > 0$), show that this method is A-stable. *Note*: A notation $h\lambda/8 \equiv z$, so that $z > 0$, should be helpful.

Solution:

Solutions for Math 337 (1)
portion of the 237/337 Qualifier.
Fall 2010.

#5 Let $Y_n = r^n$.
Sheme (1) is to be applied to $y' = -\lambda y$, $\lambda > 0$, whence $f_n = -\lambda Y_n$. Then from (1) we get:

$$r^2 - 1 = -\frac{h\lambda}{8}(5r^2 + 6r + 5)$$

Denoting $\frac{h\lambda}{8} \equiv z > 0$, we get:

$$r^2(1+5z) + 6rz - (1-5z) = 0$$

The solutions of this quadratic equation are:

$$r_{1,2} = \frac{-3z \pm \sqrt{9z^2 + (1-25z^2)}}{(1+5z)} = \frac{-3z \pm \sqrt{1-16z^2}}{1+5z}.$$

The condition that $|r_{1,2}| \le 1$ (stability) becomes:

(i) when $1 - 16z^2 > 0$, or $16z^2 < 1$:

$$\begin{cases} \dfrac{-3z + \sqrt{1-16z^2}}{1+5z} < 1 \\[2ex] \dfrac{-3z - \sqrt{1-16z^2}}{1+5z} > -1 \end{cases}$$

2

$$\Rightarrow \begin{cases} \sqrt{1-16z^2} < 1+5z+3z \\ 3z+\sqrt{1-16z^2} < 1+5z \end{cases}$$

$$\Rightarrow \begin{cases} \sqrt{1-16z^2} < 1+8z \\ \sqrt{1-16z^2} < 1+2z. \end{cases}$$

Both ~~equations~~ inequalities are true by inspection, since $z>0$.

(ii) when $\underline{1-16z^2<0}$, or $\underline{16z^2>1}$:

$$\left| \frac{-3z \pm i\sqrt{16z^2-1}}{1+5z} \right| \leq 1$$

$$\frac{9z^2+16z^2-1}{(1+5z)^2} \leq 1$$

$$25z^2-1 \leq (1+5z)^2$$

$$\cancel{25z^2}-1 \leq 1+10z+\cancel{25z^2}$$

Again, this is true since $z>0$.

Thus, in both cases (i) and (ii), $|r_{1,2}| \leq 1$, $\Rightarrow$ the method is A-stable.

□

6. Describe how you would solve a boundary-value problem on $x \in [a, b]$:

$$y'' = y^3 - x, \qquad y(a) = \alpha, \quad y(b) = \beta \tag{1}$$

with *second-order accuracy*.

If you choose to use a finite-difference discretization, do the following:

- Write the equation at an internal point.
- Write the equations at the boundary points.
- Write your system of equations in matrix (or matrix-vector) form.
- Describe what method you would use to solve (or attempt to solve) your system of equations. Provide only brief necessary details about the method's setup; do *not* go deeply into its workings.
  Note: If several alternative methods can be used, describe **only one** of them, **not all**. Also, your method does *not* have to be the best one; it should be just a reasonable method.

If you choose to use the shooting method, do the following:

- Write the equation (or equations) that you would be solving numerically.
- Explain what method(s) you would use to solve this equation (or these equations). You do *not* need to write the equations of the method(s); just write its (their) name(s) and, if needed, briefly justify your choice.

Note: You need to describe **just one** of the above methods of solution, **not both**.

Solution:

#6 <u>Discretization</u>:

- @ an internal point x_m,
  $$y_{m-1} - 2y_m + y_{m+1} = h^2(y_m^3 - x_m)$$

③

- @ the left boundary $x_0 = a$ & $x_1 = a+h$:

$$\underbrace{y_0}_{\alpha} - 2y_1 + y_2 = h^2\left(y_1^3 - (a+h)\right)$$

- @ the right boundary $x_{M-1} = b-h$, $x_M = b$:

$$y_{M-2} - 2y_{M-1} + \underbrace{y_M}_{\beta} = h^2\left(y_{M-1}^3 - (b-h)\right).$$

- In matrix-vector form:

$$A\underline{Y} = h^2\underline{Y}^3 + \underline{R}, \quad \text{where}$$

$$\underline{Y} \equiv \begin{pmatrix} y_1 \\ \vdots \\ y_{M-1} \end{pmatrix}, \quad \underline{Y}^3 \equiv \begin{pmatrix} y_1^3 \\ \vdots \\ y_{M-1}^3 \end{pmatrix}, \quad A = \underbrace{\begin{pmatrix} -2 & 1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & \ddots & \ddots & \vdots \\ 0 & \cdots & 1 & -2 \end{pmatrix}}_{\text{tridiagonal}}$$

$$R = -\begin{pmatrix} h^2(a+h) + \alpha \\ h^2(a+2h) \\ \vdots \\ h^2(b-2h) \\ h^2(b-h) + \beta \end{pmatrix}.$$

- Since this is a nonlinear system, it can be solved either by Picard's (or modified Picard's) or Newton's method.

For Picard's method, we solve

$$A\underline{Y}^{(k+1)} = h^2\left(\underline{Y}^{(k)}\right)^3 + \underline{R},$$

where $\underline{Y}^{(k)}$ is the solution at the k-th iteration.

Since A is tridiagonal, this system can be solved by the Thomas algorithm.

For Newton's method, we seek

$$\underline{Y}^{(k)} = \underline{Y} + \underline{\epsilon}, \qquad \underline{Y}^{(k)} \text{ is known},$$

where $\underline{Y}$ is the exact solution (unknown) and $\|\underline{\epsilon}\| \ll 1$. Substituting this into our matrix system and linearizing, we obtain:

$$\cancel{A\underline{Y}} + A\underline{\epsilon} = h^2 \cancel{\underline{Y}^3} + h^2 \cdot 3\underline{Y}^2\underline{\epsilon} + \cancel{\underline{R}} + O(\epsilon^2)$$

These terms cancel by virtue of $\underline{Y}$ being the exact solution.

$$A\underline{\epsilon} \approx 3h^2\,\underline{Y}^2\underline{\epsilon}$$

where $\underline{Y}^2\underline{\epsilon} \equiv \begin{pmatrix} y_1^2\,\epsilon_1 \\ \vdots \\ y_{M-1}^2\,\epsilon_{M-1} \end{pmatrix}$, $\quad y_m \approx y_m^{(k)}$.

This linear system is solved by the Thomas algorithm, whence we find

$$\underline{Y}^{(k+1)} \approx \underline{Y}^{(k)} - \underline{\epsilon},$$

and then repeat the process.

Shooting method :

- One should solve the following IVP numerically:

$$y'' = y^3 - x \,, \quad y(a) = \alpha, \quad y'(a) = \theta .$$

To have the second-order accuracy, one can represent this as a system of first-order equations and solve it by the modified Euler method, or simply use the central-difference method on the original 2nd-order equation.

- By solving the above IVP, one obtains the values $y(b)$ as functions of θ :

$$F(\theta) = y(b)\Big|_{y'(a)=\theta} .$$

Having two consequtive values $F(\theta_{n-1})$ and $F(\theta_n)$, one can estimate the solution of

$$F(\theta) = \beta$$

by the secant method.

$\square$

7. A method

$$U_j^{n+1} - U_j^n = \frac{\kappa}{h^2}\left(U_{j+1}^n - U_j^n - U_j^{n+1} + U_{j-1}^{n+1}\right) \tag{1}$$

is proposed by some people in the computational finance community in connection with solving the initial-boundary-value problem for the Heat equation:

$$u_t = u_{xx}, \qquad x \in [0,\, 1], \quad t \geq 0; \qquad u(0,t) = \alpha, \quad u(1,t) = \beta, \quad u(x,0) = \varphi(x). \tag{2}$$

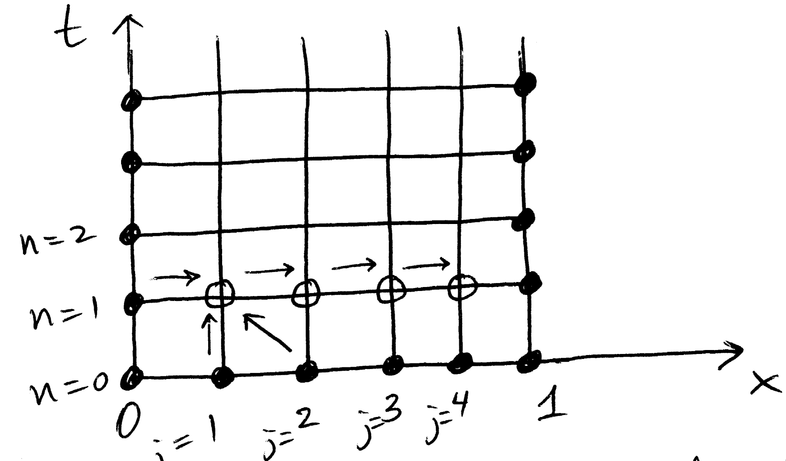
(In Eq. (1), κ and h are the temporal and spatial steps, and U_j^n is the numerical approximation to $u(jh,\, n\kappa)$.)

(a) Explain how this seemingly implicit scheme can be solved recursively, i.e. without inverting any matrix.
Hint: Draw the grid for the BVP (2) and try to find the solution node-by-node at the first time level. (The initial condition is prescribed at the zeroth time level.)

(b) Use the von Neumann analysis to show that this scheme is unconditionally stable.

Solution:

#7 (a)



In the above figure, the filled circles denote values u_j^n known from the initial and boundary conditions.

Rewrite the scheme:

$$u_j^{n+1}(1+r) - u_{j-1}^{n+1} \cdot r = u_j^n (1-r) + u_{j+1}^n \cdot r$$

$$r \equiv k/h^2.$$

Look at $\underline{n=0}$, $\underline{j=1}$ (so that $n+1=1$ is 1st time level):

$$u_1^1 (1+r) - \underbrace{u_0^1}_{\text{known @ left boundary}} \cdot r = \underbrace{u_1^0}_{\text{known @ } n=0} (1-r) + \underbrace{u_2^0}_{\text{known @ } n=0} \cdot r$$

$\Rightarrow u_1^1$ can be found.

Next, @ $n=0$, $j=2$:

7

$$u_2^1\cdot(1+r) - \underbrace{u_1^1}_{\text{known from prev. step}} r = \underbrace{u_1^0\cdot(1-r) + u_3^0}_{\text{known at } n=0}\cdot r$$

Thus u_2^1 can be found, etc. up to u_{M-1}^1. Thus, all values u_j^1, $j=1,\dots M-1$ have been found, and $u_{0,M}^1$ are known from the boundary conditions. Hence one knows all u_j^1, and then repeats to find u_j^2, etc.

(b) $u_j^n = \rho^n e^{i\beta hj}$. Then, substituting this into the scheme, we find:

$$\rho - 1 = r\left(e^{i\beta h} - 1 - \rho + \rho e^{-i\beta h}\right)$$

$$\rho\left(1 + r[1 - e^{-i\beta h}]\right) = 1 + r[e^{i\beta h} - 1]$$

$$\rho = \frac{1 + r[\cos\beta h - 1] + ir\cdot\sin\beta h}{1 + r[1 - \cos\beta h] + ir\cdot\sin\beta h}.$$

Let $r(1-\cos\beta h) = z > 0$.

Then condition $|\rho| \le 1$ becomes:

8

$$\left| \frac{(1-z) + ir \cdot \sin \beta h}{(1+z) + ir \cdot \sin \beta h} \right| \leq 1$$

$$(1-z)^2 + r^2 \cancel{\sin^2 \beta h} \leq (1+z)^2 + r^2 \cancel{\sin^2 \beta h}$$

$$\cancel{1} - 2z + \cancel{z^2} \leq \cancel{1} + 2z + \cancel{z^2}$$

$$-2z \leq 2z,$$

which is true because $z > 0$.

Thus, $|\rho| \leq 1$ for any r, i.e. the method is unconditionally stable.

$\square$