Scientific Computation Comprehensive Examination

Fall 2016

Answer 5 questions of your choice explaining all steps that lead to a solution. Partial credit will be awarded for presenting a viable solution strategy. No credit will be given to computations presented without motivation.

I. Suppose the $n \times n$ non-singular matrix A is factored as $A = LH$ where L is lower triangular with ones on its diagonal and H is upper Hessenberg (i.e., all elements $h_{i,j} = 0$ if $j < i - 1$). Design an efficient algorithm to compute the LU decomposition of A from L and H. Discuss the number of operations and memory usage of your algorithm. You may assume pivoting is not necessary.

II. Using the singular value decomposition, one can determine the numerical rank of a matrix by studying the singular values and can approximate the original matrix by a truncated low rank matrix. Describe an algorithm based on the modified Gram-Schmidt (MGS) scheme and proper pivoting technique, so that the modified QR algorithm is also rank revealing (i.e., the diagonal entries of the R matrix play the same role as the singular values).

III. For a function $f(x), x \in [0, 1]$, consider the composite midpoint rule for computing

$$I(f) = \int_0^1 f(x)dx \approx h \sum_{i=1}^{n} f((i - \frac{1}{2})h) = Q(f, n)$$

where $h = \frac{1}{n}$,
(1) Suppose that $f \in C^\infty[0, 1]$, prove that

$$I(f) - Q(f, n) = a_1 h^2 + O(h^3).$$

(2) Consider $f(x) = \frac{1}{x^\alpha}$, with $0 < \alpha < 1$, notice that there is a singularity at $x = 0$. Could you find β in the formula

$$I(f) - Q(f, n) = ch^\beta?$$

(3) ("midpoint rule" with end-point corrections.) Now consider $f(x) = \frac{1}{\sqrt{x}} g(x)$, describe how to modify the midpoint rule and get higher order by adding a "local correction", i.e., changing the weight for the function value $f(\frac{h}{2}) = \frac{g(\frac{h}{2})}{\sqrt{x}}$.

IV. Find a solution of the system

$$\sin(x) + \cos(y) + \exp(xy) = 1.5, \; \arctan(x + y) - xy = 0,$$

to two significant digits of accuracy.

V. Find the best approximation in the inf-norm on interval [-1,1] of $\cosh(x)$ by a quadratic polynomial.

VI. Construct the third order Runge-Kutta formula that approximates solutions of the ordinary differential equation $x'(t) = f(t, x)$ and uses evaluations of f at intermediate steps $t, t + h/2, t + 3h/4$.

Problem 1

Aug 2016

First find LU-decomp of H: $H = \tilde{L}\tilde{U}$

$\hat{L} = L \cdot \tilde{L}$ is lower Δ'lar matrix

$\Rightarrow A = \hat{L}\tilde{U}$ is the LU-decomp of A

Best Version

Algorithm

$\tilde{U} = H$; $\tilde{L} = I$;

for $k = 1:n-1$

  for $j = k+1:n$

    $L_{j,k} = U_{j,k} \backslash U_{k,k}$

    $U_{j,k:n} = U_{j,k:n} - L_{j,k} U_{k,k:n}$

  end

end *

$\hat{L} = L\tilde{L}$ * *

Operation Count

* - $\frac{2}{3} n^3$ ops

** - n^2 ops $\because$ lower triangular is an order of mag. less

$\Rightarrow \frac{2}{3} n^3$ ops

If we make $H = \tilde{L}\,\tilde{U}$ and mult $L \cdot \tilde{L}$ we get an LU-decomp of A.

Algorithm

$U = H \quad \tilde{L} = I$

for $k = 1, \dots, n-1$ (size $A = n \times n$)

$\alpha = U_{k+1,k} / U_{k,k}$ 1 operation 1 #

$\tilde{L}_{k+1,k} = \tilde{L}_{k+1,k} + \alpha$ 1 operation n^2 #

$\tilde{U}_{k+1,k:n} = U_{k+1,k:n} - \alpha U_{k,k:n}$ $2(n-k)$ n^2 #

end

Op Count: $\sum_{k=1}^{n-1} 2 + 2(n-k) \sim 2n^2 - n^2 = n^2$

Memory Usage: $2n^2$ "doubles" $\longleftrightarrow$ floating pt #'s

$\Rightarrow A = \underbrace{L\tilde{L}}_{\text{lower } \Delta}\ \overbrace{\tilde{U}}^{\text{upper } \Delta}$

Problem 1 Aug 2016

too long

$$A = \mathcal{L}H = \begin{bmatrix} 1 & & 0 \\ a_{21} & 1 & \\ & \ddots & \ddots \\ a_{n1} & \cdots \; a_{n,n-1} & 1 \end{bmatrix} \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1n} \\ h_{21} & h_{22} & \cdots & h_{2n} \\ & \ddots & \ddots & \vdots \\ 0 & & h_{n,n-1} & h_{nn} \end{bmatrix}$$

$H = LU$

$Hx = b$

$LUx = b$

$Ly = b$ (solve for L)

$Ux = y$ (solve for H)

Let $\tilde{L} = \mathcal{L}L$ which is lower triangular $\because$ the prod of two lower triangular matricies is itself, lower triangular.

$\Rightarrow A = \tilde{L}U$

Algorithm:

computing $H = LU$

```
for i=1:n
    u_1i = h_1i
end
for i=1:n-1
    l_i = h_{i+1,i} / u_ii
    for j=i+1:n
        u_{i+1,j} = h_{i+1,j} - l_i * u_{i,j}
    end
end
```

computational cost: n^2+2n-1

Take 2

Algorithm

$H = LU$

```
U = H ;
L = I ;

for k=1:n-1
    for j=k+1:n
        L(j,k) = U(j,k) / U(k,k)
        U(j,k:m) = U(j,k:n) - L(j,k) U(k,k:n)
    end
end
```

operation count: $O(\frac{2}{3}n^3)$

← can use ∵ pivoting is not necessar

$\bar{L} = \mathcal{L} L$

```
sum = 0
for i=1:n
    for j=1:n
        sum = 0
        for k=1:n
            sum = sum + ℒ(i,k) * L(k,j)
        end
        L̄(i,j) = sum
    end
end
```

operation count:
n mults
$n-1$ adds
n^2 entries

$\Rightarrow A = \bar{L} U$

Total operation count:

Problem 3

$$I(f) = \int_0^1 f(x)\,dx \approx h \sum_{i=1}^{n} f((i-1/2)h) = Q(f,n)$$

$h = \frac{1}{n}$ $\quad x_i = ih$

a) $Q(f,n) = h\sum_{i=1}^{n} f(x_{i-1/2}) = \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} p_i(x)\,dx$ where $p(x)$ is the interpolating polynomial thru the pt $x_{i-1/2}$

Then

$$I(f) - Q(f,n) = \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} (f(x) - p_i(x))\,dx$$

$$\cancel{= \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} \frac{f'(\xi)}{2} (x - x_{i-1/2})\,dx}$$

depends on x so we can't pull out of integr also can't use MVT so this doesn't work

Taylor series expand about $x_{i-1/2}$

$$= \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} \left(\cancel{f(x_{i-1/2})} + f'(x_{i-1/2})(x - x_{i-1/2}) + \dots - \cancel{p_i(x)}\right) dx$$

$$= \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} \left(\cancel{f'(x_{i-1/2})(x - x_{i-1/2})}^{\text{odd}} + \dots\right) dx$$

$$= \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} \left(\frac{f''(x_{i-1/2})}{2}(x - x_{i-1/2})^2 + \cancel{\frac{f'''(x_{i-1/2})}{3!}(x - x_{i-1/2})^3}^{\text{odd}} + \dots\right) d$$

$$= \sum_{i=1}^{n} \frac{f''(x_{i-1/2})}{2 \cdot 3} (x - x_{i-1/2})^3 \Big|_{x_{i-1}}^{x_{i+1}}$$

$$= \sum_{i=1}^{n} \frac{f''(x_{i-1/2})}{3!} \left(((i+1)h - (i-1/2)h)^3 - ((i-1)h - (i-1/2)h)^3 \right)$$

$$= \sum_{i=1}^{n} \frac{f''(x_{i-1/2})}{3!} \left(\frac{h^3}{8} + \frac{h^3}{8} \right)$$

$$= \sum_{i=1}^{n} a_i h^3$$

$$= O(h^2) \quad \because h = \frac{1}{n}$$

b) $f(x) = \frac{1}{x^\alpha} \qquad 0 < \alpha < 1$

Since $f \in C^\infty[h,1]$, need to focus on $\tilde{I} = \int_0^{x_1=h} f(x)\,dx \approx h f(\frac{h}{2}) = \tilde{Q}$

$$\left.\begin{aligned} \tilde{I} &= \int_0^h \frac{1}{x^\alpha}\,dx = \frac{x^{1-\alpha}}{1-\alpha}\Big|_0^h = \frac{h^{1-\alpha}}{1-\alpha} \\ \tilde{Q} &= h f\left(\tfrac{h}{2}\right) = \frac{h}{(h/2)^\alpha} = 2^\alpha h^{1-\alpha} \end{aligned}\right\} \tilde{I} - \tilde{Q} = h^{1-\alpha}\left(\frac{1}{1-\alpha} - 2^\alpha\right)$$

$\Rightarrow$ In $I(f) - Q(f,n) = c h^\beta$ we have $\beta = 1-\alpha$

c) $f(x) = \frac{1}{\sqrt{x}} g(x)$

Two methods
- Newton-Coates (will do)
- Gaussian Quad (exercise)

$\tilde{Q}(f) = w_0 f(h/2)$

$g(x) = 1$: $\quad \int_0^h \frac{1}{\sqrt{x}}\,dx = 2\sqrt{h} = w_0\, g(h/2) = w_0$

$\Rightarrow w_0 = 2\sqrt{h}$

$$\Rightarrow Q(f,n) = 2\sqrt{h}\, g\left(\tfrac{h}{2}\right) + \sum_{i=2}^{n} h f\left((i-\tfrac{1}{2})h\right)$$

Problem 4

Aug 2016

$$\begin{cases} \sin x + \cos y + e^{xy} = \frac{3}{2} \\ \arctan(x+y) - xy = 0 \end{cases} \Rightarrow \begin{cases} f_1 = \sin x + \cos y + e^{xy} - \frac{3}{2} = 0 \\ f_2 = \arctan(x+y) - xy = 0 \end{cases}$$

We can use multidim. Newton's method:

$$\vec{x}_{n+1} = \vec{x}_n - \left(DF(x_n) \right)^{-1} F(\vec{x}_n) \qquad \vec{x}_n = \begin{bmatrix} x_n \\ y_n \end{bmatrix}$$

$$DF = \begin{bmatrix} \nabla f_1 \\ \nabla f_2 \end{bmatrix} = \begin{bmatrix} \cos x + y e^{xy} & -\sin y + x e^{xy} \\ \frac{1}{1+(x+y)^2} - y & \frac{1}{1+(x+y)^2} - x \end{bmatrix}$$

Initial guess: $\vec{x}_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

$$* \begin{cases} D(F(\vec{x}_0)) = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \Rightarrow \left(DF(\vec{x}_0) \right)^{-1} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \\ F(\vec{x}_0) = \begin{bmatrix} 1+1-3/2 \\ 0 \end{bmatrix} = \begin{bmatrix} 1/2 \\ 0 \end{bmatrix} \end{cases}$$

$$\vec{x}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1/2 \\ 0 \end{bmatrix} = \begin{bmatrix} -1/2 \\ 0 \end{bmatrix}$$

Iterate $*$ to get $\vec{x}_2$. Iterate until first 2 decimal places are consistent and problem will be done.

Problem 5

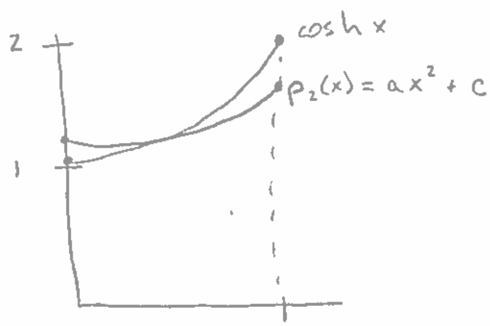
Aug 2016

Goal: Minimize $\max_{x \in [-1,1]} \left| \cosh(x) - (ax^2 + bx + c) \right| = \| \cosh(x) - p_2(x) \|_\infty$

Since cosh is symmetric about $x = 0$, we can assume $p_2(x)$ is too so $b = 0$.

$\Rightarrow p_2(x) = ax^2 + c$

Also, since they are both symmetric about $x = 0$, we need only consider the interval $[0, 1]$.



Max difference will be achieved at the end points. The difference should be same in magnitude but opposite in sign.

$$\Rightarrow \quad p_2(0) - \cosh(0) = \cosh(1) - p_2(1)$$

$$c - 1 = \cosh(1) - (a + c)$$

$$a + 2c = \cosh(1) + 1$$

Also want slope at $x = 1$ to be the same so:

$$\sinh(x)\Big|_{x=1} = 2ax\Big|_{x=1}$$

$$\sinh(1) = 2a$$

$$a = \frac{\sinh(1)}{2} \quad \Rightarrow \quad c = \frac{1}{2}\left(\cos(1) + 1 - \frac{1}{2}\sinh(1)\right)$$

$$\Rightarrow \boxed{p_2(x) = \frac{\sinh(1)}{2} x^2 + \frac{\cosh(1) + 1}{2} - \frac{\sinh(1)}{4}}$$

<u>Problem 6</u> Aug 2016

$$y_{n+1} = y_n + (b_1 K_1 + b_2 K_2 + b_3 K_3) h \qquad (1)$$

Given: $c_1 = 0$, $c_2 = \frac{1}{2}$, $c_3 = \frac{3}{4}$

$$K_1 = f(t_n, y_n) = f_n$$

Use Taylor series!

$$K_2 = f\left(t + \tfrac{h}{2}, y_n + h a_{21} K_1\right)$$

$$K_3 = f\left(t + \tfrac{3}{4}h, y_n + h a_{32} K_2\right)$$

$$K_1 = f_n$$

$$K_2 = f_n + \frac{h}{2}\partial_t f_n + h a_{21} \overset{f_n}{K_1} \partial_y f_n + \frac{(h/2)^2}{2!}\partial_{tt} f_n + \frac{(h a_{21} \overset{f_n}{K_1})^2}{2!}\partial_{yy} f_n + 2\left(\frac{(h/2) h a_{21} \overset{f_n}{K_1}}{2}\right)\partial_{yt} f_n + O(h^3)$$

$$K_3 = f_n + \frac{3h}{4}\partial_t f_n + (h a_{31} K_1 + h a_{32} K_2)\partial_y f_n + \frac{(3h/4)^2}{2}\partial_{tt} f_n + \frac{1}{2}(h a_{31} K_1 + h a_{32} K_2)^2 \partial_{yy} f_n + \frac{3h}{4}(h a_{31} K_1 + h a_{32} K_2)\partial_{ty} f_n + O(h^3)$$

$$= f_n + \frac{3h}{4}\partial_t f_n + \left(h a_{31} f_n + h a_{32}\left(f_n + \frac{h}{2}\partial_t f_n + h a_{21} f_n \partial_y f_n + \ldots\right)\right)\partial_y f_n$$

$$+ \frac{9h^2}{32}\partial_{tt} f_n + \frac{3h^2}{4} f_n \partial_{ty} f_n + \frac{3h^2}{4} a_{32} f_n \partial_{ty} f_n$$

$$+ \frac{h^2}{2}(a_{31} f_n + a_{32} f_n)^2 \partial_{yy} f_n$$

$$y_{n+1} = y_n + h \overset{f_n}{y_n'} + \frac{h^2}{2}(\partial_t f_n + \partial_y f_n f_n) + O(h^3)$$

Sub y_{n+1}, K_1, K_2 and K_3 into (1) and create a system of equations by matching terms on either side of the eqn. Solve to get coeffs.