UNIVERSITY OF OSLO

Faculty of mathematics and natural sciences

Exam in: MAT3110 — Introduction to numerical analysis

Day of examination: 14 December 2018

Examination hours: 0900 – 1300

This problem set consists of 5 pages.

Appendices: None

Permitted aids: None

Please make sure that your copy of the problem set is complete before you attempt to answer anything.

All 8 part questions will be weighted equally.

Problem 1 Matlab function

Write a Matlab (or Python) function `[L,U] = mylu(A)` that computes the LU factorization of an $n \times n$ matrix A, assuming that no pivoting is required.

Answer:

```
function [L,U] = mylu(A)

n = size(A,1);
L=zeros(n,n);
U=zeros(n,n);
B = A;

for k=1:n
  U(k,:) = B(k,:);
  L(:,k) = B(:,k) / B(k,k);
  B = B - L(:,k) * U(k,:);
end
```

Problem 2 Cholesky

Use the Cholesky algorithm to determine whether the matrix

$$A = \begin{bmatrix} 2 & 6 & -4 \\ 6 & 17 & 7 \\ -4 & 7 & 10 \end{bmatrix}$$

(Continued on page 2.)

is positive-definite.

Answer By a theorem from the course, A is positive-definite if and only if it has an LDL^T factorization where the diagonal elements of D are positive.

We initialize $A_0 = A$. In the first step we use the first column of A_0:

$$\mathbf{l}_1 = \frac{1}{2}\begin{bmatrix} 2 \\ 6 \\ -4 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ -2 \end{bmatrix},$$

and the first diagonal element of A_0:

$$D_{1,1} = 2.$$

Then we find

$$\mathbf{l}_1\mathbf{l}_1^T = \begin{bmatrix} 1 & 3 & -2 \\ 3 & 9 & -6 \\ -2 & -6 & 4 \end{bmatrix},$$

and so

$$D_{1,1}\mathbf{l}_1\mathbf{l}_1^T = \begin{bmatrix} 2 & 6 & -4 \\ 6 & 18 & -12 \\ -4 & -12 & 8 \end{bmatrix},$$

and we let

$$A_1 = A_0 - D_{1,1}\mathbf{l}_1\mathbf{l}_1^T = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 19 \\ 0 & 19 & 2 \end{bmatrix}.$$

In the second step of the algorithm we set $D_{2,2}$ to be the second element of the second column of A_1. Thus $D_{2,2} = -1$. Thus A is not positive-definite.

Problem 3 Matrix norm and SVD

3a

Recall that the 2-norm of a vector $\mathbf{x} = (x_1, \ldots, x_n)$ in $\mathbb{R}^n$ is

$$\|\mathbf{x}\|_2 = \left(\sum_{i=1}^{n} x_i^2\right)^{1/2}.$$

If U is an orthogonal $n \times n$ matrix, what is the 2-norm of $U\mathbf{x}$?

Answer: If U is orthogonal then $U^TU = I$. Therefore

$$\|U\mathbf{x}\|_2^2 = (U\mathbf{x})^T(U\mathbf{x}) = \mathbf{x}^TU^TU\mathbf{x} = \mathbf{x}^T\mathbf{x} = \|\mathbf{x}\|_2^2,$$

and so the 2-norm of $U\mathbf{x}$ equals the 2-norm of $\mathbf{x}$.

(Continued on page 3.)

3b

What is the definition of the 2-norm of a real $n \times n$ matrix A?

Answer:

$$\|A\|_2 = \max_{\mathbf{x} \neq 0} \frac{\|A\mathbf{x}\|_2}{\|\mathbf{x}\|_2}.$$

3c

Using the singular value decomposition of A, show that the 2-norm of A equals σ_1, the largest singular value of A.

Answer: The SVD of A is

$$A = USV^T,$$

where $U, S, V \in \mathbb{R}^{n,n}$, U, V are orthogonal, and S is diagonal with diagonal elements $\sigma_1, \ldots, \sigma_n$, the singular values of A, where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$. Therefore,

$$\|A\|_2 = \max_{\mathbf{x} \neq 0} \frac{\|USV^T\mathbf{x}\|_2}{\|\mathbf{x}\|_2} = \max_{\mathbf{x} \neq 0} \frac{\|SV^T\mathbf{x}\|_2}{\|\mathbf{x}\|_2},$$

because U is orthogonal. Letting $\mathbf{x} = V\mathbf{y}$,

$$\|A\|_2 = \max_{V\mathbf{y} \neq 0} \frac{\|S\mathbf{y}\|_2}{\|V\mathbf{y}\|_2} = \max_{\mathbf{y} \neq 0} \frac{\|S\mathbf{y}\|_2}{\|\mathbf{y}\|_2},$$

because V is also orthogonal. Therefore

$$\|A\|_2 = \max_{\mathbf{y} \neq 0} \left(\frac{\sum_{i=1}^n \sigma_i^2 y_i^2}{\sum_{i=1}^n y_i^2} \right)^{1/2}.$$

Therefore,

$$\|A\|_2 \leq \sigma_1$$

and the inequality is equality because we can choose $\mathbf{y} = (1, 0, \ldots, 0)$.

Problem 4 Iterative scheme

Consider the linear system $A\mathbf{x} = \mathbf{b}$ where A is a non-singular $n \times n$ matrix and $\mathbf{b}$ is a vector in $\mathbb{R}^n$. Suppose we try to find the solution $\mathbf{x}$ using the iterative scheme

$$(A - B)\mathbf{x}^{(k+1)} = -B\mathbf{x}^{(k)} + \mathbf{b}, \qquad k = 0, 1, 2,$$

with $\mathbf{x}^{(0)}$ some initial guess. Assuming $A - B$ is non-singular, what condition on A, B, and $\mathbf{b}$ is both necessary and sufficient for the sequence $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots$ to converge to $\mathbf{x}$? Explain your answer.

Answer: Consider the k-th error,

$$\mathbf{e}^{(k)} := \mathbf{x}^{(k)} - \mathbf{x}.$$

(Continued on page 4.)

The equation

$$(A - B)\mathbf{x} = -B\mathbf{x} + \mathbf{b},$$

is also satisfied by $\mathbf{x}$ and subtracting it from

$$(A - B)\mathbf{x}^{(k+1)} = -B\mathbf{x}^{(k)} + \mathbf{b}$$

implies

$$(A - B)\mathbf{e}^{(k+1)} = -B\mathbf{e}^{(k)}.$$

Under our assumption that $A - B$ is non-singular this means that

$$\mathbf{e}^{(k+1)} = H\mathbf{e}^{(k)}, \qquad k = 0, 1, 2, \ldots, \tag{1}$$

where

$$H = -(A - B)^{-1}B$$

By a theorem from the notes, the error vectors $\mathbf{e}^{(k)}$ converge to zero as $k \to \infty$ if and only if $\rho(H) < 1$, where $\rho(H)$ is the spectral radius of H,

$$\rho(H) = \max_{i=1,\ldots,n} |\lambda_i|,$$

and the λ_i are the eigenvalues of H.

Problem 5 Polynomial interpolation

The polynomial of degree $\leq n$ that interpolates a function $f : [-1, 1] \to \mathbb{R}$ at distinct points $x_0, x_1, \ldots, x_n \in [-1, 1]$ can be expressed as

$$p_n(x) = \sum_{i=0}^{n} \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j} f(x_i).$$

For approximating f, what is a good choice of the points x_i when n is large?

Answer:

The interpolation error is given by

$$e(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\omega(x), \qquad x \in [-1, 1],$$

where $\xi \in [-1, 1]$ and

$$\omega(x) = \prod_{i=0}^{n}(x - x_i).$$

We can minimize the absolute value of ω in $[-1, 1]$ by choosing

$$\omega(x) = 2^{-n}T_{n+1}(x),$$

where T_{n+1} is the Chebyshev polynomial

$$T_{n+1}(x) = \cos((n + 1) \arccos(x)).$$

Thus a good choice of $x_0, \ldots, x_n$ is the zeros of T_{n+1} which are

$$x_i = \cos\left(\frac{\pi}{2}\frac{2i + 1}{n + 1}\right), \quad i = 0, 1, \ldots, n.$$

(Continued on page 5.)

Problem 6 Non-linear least squares

Suppose we want to minimize a function $f : \mathbb{R}^n \to \mathbb{R}$ of the form

$$f(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{m} r_i(\mathbf{x})^2, \quad \mathbf{x} \in \mathbb{R}^n, \tag{2}$$

where $r_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, n$, are the so-called residuals. What are the Newton and Gauss-Newton methods for this problem, and what are their advantages and disadvantages?

Answer: The Newton method for minimizing f is Newton's method for root-finding applied to $\nabla f(\mathbf{x})$, which is the iteration

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - (\nabla^2 f(\mathbf{x}^{(k)}))^{-1}\nabla f(\mathbf{x}^{(k)}).$$

An advantage of this method is that it has quadratic convergence if it converges.

The Gauss-Newton method is a simplification of the Newton method. Differentiating (2) with respect to x_j gives

$$\frac{\partial f}{\partial x_j} = \sum_{i=1}^{m} \frac{\partial r_i}{\partial x_j} r_i,$$

and so the gradient of f is

$$\nabla f = J_r^T \mathbf{r},$$

where $\mathbf{r} = [r_1, \ldots, r_m]^T$ and $J_r \in \mathbb{R}^{m,n}$ is the Jacobian of $\mathbf{r}$,

$$J_r = \left[\frac{\partial r_i}{\partial x_j}\right]_{i=1,\ldots,m, j=1,\ldots,n}.$$

Differentiating again, with respect to x_k, gives

$$\frac{\partial^2 f}{\partial x_j \partial x_k} = \sum_{i=1}^{m}\left(\frac{\partial r_i}{\partial x_j}\frac{\partial r_i}{\partial x_k} + r_i \frac{\partial^2 r_i}{\partial x_j \partial x_k}\right),$$

and so the Hessian of f is

$$\nabla^2 f = J_r^T J_r + Q,$$

where

$$Q = \sum_{i=1}^{m} r_i \nabla^2 r_i.$$

The Gauss-Newton method is the result of neglecting the term Q, i.e., making the approximation

$$\nabla^2 f \approx J_r^T J_r. \tag{3}$$

Thus the Gauss-Newton iteration is

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - (J_r(\mathbf{x}^{(k)})^T J_r(\mathbf{x}^{(k)}))^{-1} J_r(\mathbf{x}^{(k)})^T \mathbf{r}(\mathbf{x}^{(k)}).$$

In general the Gauss-Newton method will not converge quadratically. On the other hand it is easier to implement and it is more robust than Newton's method because the search direction is always a descent direction.

Good luck!