NTNU
Norwegian University of
Science and Technology

Department of Mathematical Sciences

Examination paper for **MA2501 Numerical Methods**

Academic contact during examination: Kjetil André Johannessen

Phone: 91345314

Examination date: 16th of May 2017

Examination time (from–to): 09:00–13:00

Permitted examination support material: Support material code C

- Approved basic calculator.
- The textbook: Cheney & Kincaid, Numerical Mathematics and Computing, 6th or 7th edition, including the list of errata.
- Rottmann, Matematisk formulae.
- Handout: Fixed point iterations.

Other information:
All answers should be justified and include enough details to make it clear which methods and/or results have been used.

All the (sub-)problems are worth 5 points each. The total value is 70 points.

Language: English

Number of pages: 5

Number of pages enclosed: 0

Checked by:

Date Signature

Problem 1

a) The matrix is symmetric as $a_{ij} = a_{ji}\ \forall i, j = 1, 3$. Furthermore, all diagonal terms are positive and we have $a_{ii} > \sum_{j=1,j\neq i}^{3} |a_{ij}|$, i.e. the matrix is diagonally dominant. It then follows then from Gershgorin theorem that all the eigenvalues are real and positive, hence the matrix is positive definit.

b) The conditioning number $\kappa(A) = ||A||||A^{-1}||$. For matrices with large condition numbers (ill-condition equation system) may small perturbations in the right hand side result in large errors in the computed unknowns.

c) We fin and LU-factorization by Gauss elimination of A:

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

In each step we fill in the off-diagonal part of L with the coefficients used to eliminate the lower triangular part of A:

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 3 & 1 \end{pmatrix}$$

One can now verify that $A = LU$. Note that the LU-factorization is not unique. Another possible factorization is

$$L = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ \frac{1}{2} & -\frac{1}{7} & 1 \end{pmatrix} \quad , \quad U = \begin{pmatrix} 6 & 3 & 3 \\ 0 & \frac{7}{2} & -\frac{3}{2} \\ 0 & 0 & \frac{2}{7} \end{pmatrix}$$

Problem 2

a) The interpolating polynom of lowest order is: $p_2 = 1 + 4x + 2x^2$. This may be found by using the definition of Lagrange polynomial, but here the use of Newton interpolation formula and divided differences as it may be consider advantageous as that makes next question easier to solve.
Thus we get: $p_2(x) = 1 + 8(x - 0) + 2(x - 0)(x - 2) = 1 + 4x + 2x^2$.

b) We now just add one row and column to our table:
Thus we get: $p_2(x) = 1+8(x-0)+2(x-0)(x-2)+0(x-0)(x-2)(x-3) = 1 + 4x + 2x^2$.

x_i	$f[x_i]$	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$
0.0	1.0		
		8.0	
2.0	17.0		2.0
		14.0	
3.0	31.0		

(1)

x_i	$f[x_i]$	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$	
0.0	1.0			
		8.0		
2.0	17.0		2.0	
		14.0		0.0
3.0	31.0		2.0	
		12.0		
1.0	7.0			

(2)

c) A third order polynomial $p_3(x) = c_0 + c_1 x + c_2 x^2 c_3 x^3$ has four unknown coefficient that may be determined by four independent constraints. Here we choose the following constraints: $p(0) = f(0)$, $p'(0) = f'(0)$, $p(a) = f(a)$, and $p'(a) = f'(a)$, i.e for $f(x) = x^5$ we get the following system to solve:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & a & a^2 & a^3 \\ 0 & 1 & 2a & 3a^2 \end{bmatrix} \times \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ a^5 \\ 5a^4 \end{bmatrix} \tag{3}$$

We see immediately that $c_0 = c_1 = 0$ thus we end up with the following system:

$$\begin{bmatrix} a^2 & a^3 \\ 2a & 3a^2 \end{bmatrix} \times \begin{bmatrix} c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} a^5 \\ 5a^4 \end{bmatrix} \tag{4}$$

This gives $c_2 = -2a^3$ and $c_3 = 3a^2$.

Problem 3

a) Determine the values A_0, A_1, A_2 and x_1 such that the quadrature rule $Q(f) = A_0 f(-1) + A_1 f(x_1) + A_2 f(1)$ gives the correct value for the integral $\int_{-1}^{1} f(x)dx$ when f is any polynomial of degree 3 (since we have four "free variable" we should be able to solve exactly for p+1=4).

Direct construction gives:

$$\begin{bmatrix} 1 & 1 & 1 \\ -1 & x_1 & 1 \\ 1 & x_1^2 & 1 \\ -1 & x_2^3 & 1 \end{bmatrix} \times \begin{bmatrix} A_0 \\ A_1 \\ A_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 2/3 \\ 0 \end{bmatrix} \tag{5}$$

Trick: Due to symmetry we have that $A_0 = A_2$, which together with the second equation gives $x_1 = 0$. Finally we find the solution to be: $A_0 = \frac{1}{3}, A_1 = \frac{4}{3}, A_2 = \frac{1}{3}$, and $x_1 = 0$.

b) Determine the values A_0, A_1 and x_0, x_1 such that the Gaussian quadrature rule
$Q(f) = A_0 f(x_0) + A_1 f(x_1)$ gives the correct value for the integral $\int_{-1}^{1} f(x)dx$ when f is any polynomial of degree 3 (since we have four "free variable" we should be able to solve exactly for p+1=4).

Direct construction gives:

$$\begin{bmatrix} 1 & 1 \\ x_0 & x_1 \\ x_0^2 & x_1^2 \\ x_0^3 & x_1^3 \end{bmatrix} \times \begin{bmatrix} A_0 \\ A_1 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 2/3 \\ 0 \end{bmatrix} \tag{6}$$

Trick: Due to symmetry we have that $x_0 = -x_1$ and $A_0 = A_1$. From the first equation we then get $A_0 = A_1 = 1$ and the third equation then imply $2x_0^2 = 2/3$, i.e., $x_0 = 1/-\sqrt{3}$ and $x_1 = 1/\sqrt{3}$.

Lobatto rule gives for x^3: $\frac{1}{3}(-1)^3 + \frac{4}{3}(0)^3 + \frac{1}{3}(1)^3 = 0$
Gauss rule gives for x^3: $1(\frac{-1}{\sqrt{3}})^3 + 1(\frac{1}{\sqrt{3}})^3 = 0$
Thus, both rules gives the exact answer!

c) Lobatto rule gives for x^4: $\frac{1}{3}(-1)^4 + \frac{4}{3}(0)^3 + \frac{1}{3}(1)^4 = 2/3 = 0.6667$.
Gauss rule gives for x^4: $1(\frac{-1}{\sqrt{3}})^4 + 1(\frac{1}{\sqrt{3}})^4 = 2/9 = 0.2222$.
The exact answer is $2/5 = 0.4$
The error for Lobatto is 0.3667 and about half the size for the Gauss quadrature rule: $0,1778$.

Problem 4

a) We rewrite the system of equations as

$$F(X) = \begin{bmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{bmatrix} = \begin{bmatrix} x_1^2 + x_2^2 - 1 \\ x_1^3 - x_2 - 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

where $X = (x_1, x_2)^T$. The Jacobian matrix becomes

$$J(X) = \begin{bmatrix} \partial f_1/\partial x_1 & \partial f_1/\partial x_2 \\ \partial f_2/\partial x_1 & \partial f_2/\partial x_2 \end{bmatrix} = \begin{bmatrix} 2x_1 & 2x_2 \\ 3x_1^2 & -1 \end{bmatrix}$$

We can then write Newton's method as

$$X^{(n+1)} = X^{(n)} + H^{(n)}$$

where $H^{(n)}$ is implicitly given by

$$J(X^{(n)})H^{(n)} = -F(X^{(n)})$$

b) We must avoid initial values where the Jacobian is singular, i.e. when $\det(J(X)) = 0$:

$$\det(J(X)) = -2x_1 - 6x_1^2x_2 = -2x_1(1 + 3x_1x_2) = 0$$

Thus, we must keep away from the curves $x_1 = 0$ and $3x_1x_2 = -1$, and choose e.g. initial values $x_1^{(0)} = x_2^{(0)} = 0.5$. Thus

$$F(X^{(0)}) = \begin{bmatrix} -0.5 \\ -2.375 \end{bmatrix}$$

$$J(X^{(0)}) = \begin{bmatrix} 1 & 1 \\ 0.75 & -1 \end{bmatrix}$$

and to obtain $H^{(0)}$ we solve the linear system of equations

$$\begin{bmatrix} 1 & 1 \\ 0.75 & -1 \end{bmatrix} \begin{bmatrix} h_1^{(0)} \\ h_2^{(0)} \end{bmatrix} = \begin{bmatrix} -0.5 \\ -2.375 \end{bmatrix}$$

The solution is

$$H^{(0)} = \begin{bmatrix} 1.6429 \\ -1.1429 \end{bmatrix}$$

After one iteration we thus get $x_1^{(1)} = 2.1429$ and $x_2^{(1)} = -0.6429$.
In the same way, we obtain $x_1^{(2)} = 1.6288$ and $x_2^{(2)} = 0.7588$.
In our case, we can alternatively easily calculate J^{-1} by hand, which leads to the explicit iteration

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \mapsto \frac{1}{2x_1(1 + 3x_1x_2)} \begin{bmatrix} 4x_1^3x_2 + x_1^2 + x_2^2 + 4x_2 + 1 \\ 3x_1^2(x_2^2 - x_1^2 + 3) - 4x_1 \end{bmatrix}$$

Calculating J^{-1} as we have done will normally be very cumbersome. Instead one usually solves (1) numerically, e.g. with the conjugate gradient method. MATLAB does this for us if we solve (1) using the \ operator.

c) As we saw in a), the Jacobian is singular on the x_1-axis. This causes the algorithm to fail, since we don't get a unique solution when solving (1).

Problem 5

a) For explicit methods we don't have to solve an equation system, whereas we for implicit methods have to.

b) When the step-lenght for an explicit method is governed by the stability requirements, and not because of accuracy tolerances, it is usually more efficient to use implicit methods.