Solution to the written exam

Tecnical University of Denmark

Written 4-hour exam, 17. Dec. 2020

Course name: Introduction to Numerical Algorithms Course nr.: 02601

Aids and materials: All aids and materials permitted.

Question 1: Approximation of a function (20%)

The Matlab code:

```
 Nvalue=9;
I=[1,2];
x=linspace(I(1),I(2),400);

f=@(x) log(x);

nodes=linspace(I(1),I(2),Nvalue+1);
ndata=f(nodes);

ydata=InterpolerLagrangeForm(nodes,ndata,x);
err =f(x)-ydata;
max_err=max(abs(err))

Theoreticalmaxerr=@(n)1/(4*(n+1))*factorial(n)*(1/n)^(n+1);
tbound=Theoreticalmaxerr(Nvalue)
```

1.1) The solution is 3.63×10^{-8}.

1.2) Based on the second interpolation error theorem, we have

$$|f(x) - p_9(x)| \leq \frac{1}{4(n+1)} Mh^{n+1} = \frac{1}{4 \times 10} \cdot 9! \cdot \left(\frac{2-1}{9}\right)^{10} \approx 2.60 \times 10^{-6}$$

with M as the upper bound of $|f^{n+1}(x)|$.

1.3) Same as in **1.1)** except setting `Nvalue=25` in the code. The solution is 1.83×10^{-11}.

1.4) Based on the second interpolation error theorem, we have

$$|f(x) - p_{25}(x)| \leq \frac{1}{4(n+1)} Mh^{n+1} \approx 6.72 \times 10^{-14},$$

which is smaller than the one we got in **1.4)**. The reason is loss of significant in subtraction.

Question 2. Data fitting (20%)

The Matlab code:

```
xk = [-2:1:2]';
yk = [0 15 25 50 110]';

m = length(xk);
A = [ ones(m,1), xk, xk.^2-1];

At=A';
At*A  % system matrix
At*yk  % right-hand side
c=(At*A)\(At*yk) % solution

norm(yk-A*c)

ub = cond(At*A)*0.01
```

2.1) The system matrix $\boldsymbol{A}^T\boldsymbol{A}$ and the right-hand side $\boldsymbol{A}^T y$ of the normal equation are

$$\begin{bmatrix} 5 & 0 & 5 \\ 0 & 10 & 0 \\ 5 & 0 & 19 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 200 \\ 255 \\ 305 \end{bmatrix}.$$

2.2) The solution is obtained by `c = (At'*A)\(At'*yk)`:

$$a = 32.5, \quad b = 25.5, \quad c = 7.50\ .$$

2.3) The absolute errors is obtained by `e = norm(A*c-yk)`: 12.65.
2.4) According to the sensitivity analysis, the upper bound of the relative error in the solution is

$$\kappa(\boldsymbol{A}^T\boldsymbol{A})\frac{\|\boldsymbol{A}^T\tilde{\boldsymbol{y}} - \boldsymbol{A}^T\boldsymbol{y}\|_2}{\|\boldsymbol{A}^T\boldsymbol{y}\|_2} = 0.061.$$

Question 3: Newton's method for a system of nonlinear equations (15%)

3.1)

```
function [F,dF] = funFdF(X)

F = [X(1)^2+2*X(2)-2;...
     X(1)+4*X(2)^2-4];

dF = [2*X(1)      2;...
      1       8*X(2)];
```

3.2) Newton iteration is

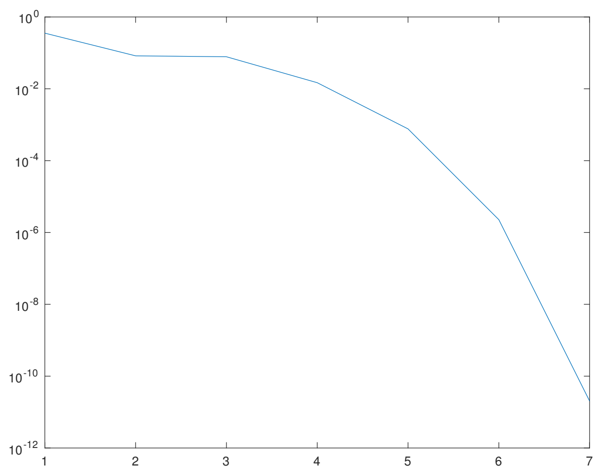
$$\boldsymbol{X}^{(k+1)} = \boldsymbol{X}^{(k)} - (F'(\boldsymbol{X}^{(k)}))^{-1}F(\boldsymbol{X}^{(k)}).$$

3.3) The final solution is $[-5.93 \times 10^{-17}, 1]^T$.
The Matlab code:

```
X0=[1;2];
Xiterations = Newtonsys(@funFdF,X0,8);
Xiterations(:,end)

Xexa=Xiterations(:,end);
err=sqrt((Xiterations(1,1:end-1)-Xexa(1)).^2+(Xiterations(2,1:end-1)-Xexa(2)).^2)
figure, semilogy(err)
```

3.3)



Question 4: Integration (10%)

4.1) Set $f(x) = \frac{1}{x}$, then we have $f^{(4)}(x) = 24x^{-5}$ and $|f^{(4)}| \leq 24$ in the interval $[1, 2]$. According to the error term for the composite Simpson's rule, we need

$$\left| \frac{1}{180} \cdot h^4 \cdot 24 \right| \leq 10^{-4},$$

that is $h < 0.16549$. Then, use $h = 1/n$ and we obtain $n > 6.04$.

4.2) Since n need be an even integer, we set $n = 8$. Run `MySimpson(@(x)1/x, 1, 2, 8)` and we will get 0.69315.

Question 5: LU factorization with pivoting (10%)

The Matlab code:

```
A=[2 -2 3 -3;1 -1 2 -1; 1 -1 4 3; 1 1 1 0];
[L,U,P]=lu(A)
```

5.1) The matrix $\boldsymbol{P}$ is

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

It shows that we need interchange row 2 and row 4.

5.2) The reasons to use pivoting are:

- Mathematically: avoid dividing by 0.
- Numerically: avoid large multipliers in row eliminations, which can lead to loss of significance.

Question 6: A boundary-value problem (20%)

6.1) Set

$$Y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} x \\ x' \end{bmatrix},$$

then we have

$$Y' = \begin{bmatrix} x' \\ x'' \end{bmatrix} = \begin{bmatrix} y_2 \\ y_2 + y_1 - (2t-1)e^t \end{bmatrix},$$

The Matlab code

```
function dydt = odefun_sol(t,Y)
dydt=[Y(2); Y(2)+Y(1)-(2*t-1)*exp(t)];
```

6.2) The value of $x(2)$ in these two shootings are 29.71 and 59.93.
The Matlab code:

```
tspan=[1 2];
Y1=[3*exp(1); 10];
Y2=[3*exp(1); 25];

[tODE1,YODE1]=ode45(@odefun_sol,tspan,Y1);
[tODE2,YODE2]=ode45(@odefun_sol,tspan,Y2);

YODE1(end,1)
YODE2(end,1)
```

6.3) We need find the root of the equation $f(z) = \varphi(z) - 5e^2 = 0$. Use the secant method, we get

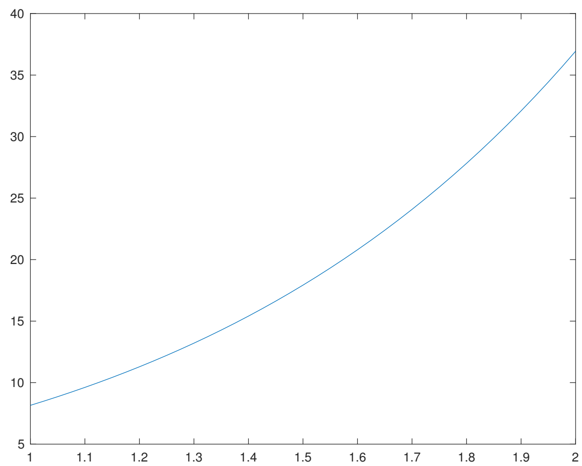
$$\hat{z} = z_2 - \left(\frac{z_2 - z_1}{f(z_2) - f(z_1)} \right) f(z_2) = 13.59$$

The Matlab code for the plot:

```
f=@(y)y-5*exp(2);
Y3_2=Y2(2)-(Y1(2)-Y2(2))/(f(YODE1(end,1))-f(YODE2(end,1)))*f(YODE2(end,1))

%%
```

```
tspan=[1 2];
Y3=[3*exp(1); Y3_2];
[tODE3,YODE3]=ode45(@odefun_sol,tspan,Y3);
YODE3(end,1)
figure, plot(tODE3,YODE3(:,1))
```



End