Tecnical University of Denmark

Written 4-hour exam, 19. Decemeber 2018

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

Aids and materials: All aids and materials permitted.

Weights: Multiple choice 25%, Question 1: 10%, Question 2: 20%, Question 3: 15%, Question 4: 10%, Question 5: 20%. The weight is only a guideline. The final grade is based on overall assessment.

Multiple choice (25%)

Each question has only ONE correct answer. In the pdf file with your answers, you just need type the number of the answer. You get 5% for a correct answer, 0% for no answer, and -2% for a wrong answer.

A – Gaussian elimination with partial pivoting. The reason for always performing partial pivoting in Gaussian elimination is:

1. It reduces the computational complexity by a factor 2.
2. We must make sure that the largest element in the solution to $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ lies in the first element in $\boldsymbol{x}$.
3. We must avoid that the solution has a large rounding error due to e.g. loss of significance.
4. To make sure that the matrix $\boldsymbol{A}$ has the full rank.

B – Interpolation. Consider the table

x	0	3	3	5
y	-2	-2	0	9

1. The table **does not satisfy** the condition to be an interpolation table because the values of x can not be 0 (zero).
2. The table **does not satisfy** the condition to be an interpolation table because the values of x must be equally spaced.
3. The table **does not satisfy** the condition to be an interpolation table because there cannot be two points with the same y values.
4. The table **does not satisfy** the condition to be an interpolation table because there cannot be two points with the same x values.
5. The table **satisfies** the condition to be an interpolation table.

C – Convergence of the secant method. Consider a continuous and differentiable function of one variable. With a starting point "close enough" to a simple root, the convergence of *secant method* is

1. Linear.
2. Superlinear.
3. Quadratic.
4. Something else.

D – Sensitivity analysis. We consider two systems of linear equations

$$\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{b}, \qquad \boldsymbol{A}\,\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{b}}, \qquad \tilde{\boldsymbol{b}} = \boldsymbol{b} + \delta\boldsymbol{b}. \tag{1}$$

where $\boldsymbol{A}$ is a 100×100 matrix that is invertible. We set $\delta\boldsymbol{x} = \tilde{\boldsymbol{x}} - \boldsymbol{x}$, the condition number of $\boldsymbol{A}$ is denoted by $\kappa(\boldsymbol{A})$, and we have

$$\kappa(\boldsymbol{A}) = 5.3, \qquad \|\boldsymbol{b}\|_2 = 1.4, \qquad \boldsymbol{x} = [1, -1, 1, -1, \cdots].$$

If the absolute error $\delta\boldsymbol{b}$ satisfies:

$$\|\delta\boldsymbol{b}\|_2 \leq 0.07, \tag{2}$$

we obtain the *upper bound* for the relative error $\|\delta\boldsymbol{x}\|_2/\|\boldsymbol{x}\|_2$ in the solution as

1. 0.371.
2. 0.265.
3. 2.65.
4. 7.42.

E – Numerical integration. Consider the function $f(x) = xe^x$ on the interval $I = [0, 4]$. Apply the composite Simpson's rule on n equally spaced subintervals to approximate the integral $\int_0^4 f(x)\,dx$. We know that the mth derivative of f is $f^{(m)}(x) = (m + x)e^x$. What is the smallest value n such that the largest approximation error is guaranteed to not exceed 10^{-4}?

1. $n = 71$
2. $n = 2090$
3. $n = 72$
4. $n = 84$

Question 1: Gaussian elimination with partial pivoting (10%)

We have the following system of linear equations:

$$\begin{aligned} 2x + 4y + 5z &= 13 \\ 3x + 6y - 3z &= 9 \\ x - 2y + 8z &= 4. \end{aligned}$$

Apply Gaussian elimination with partial pivoting to solve this linear system. This question does not need use Matlab. The derivations, intermediate results and explanations should be included in the pdf file with your answers.

1.1) State the final upper triangular matrix and the elimination coefficient in each step.

1.2) Use back substitution to obtain the solution of this linear system and state the solution.

Question 2. Data-fitting (20%)

Suppose that we want to fit the data (x_k, y_k), $k = 0, \cdots, 25$, by a function of the form

$$F(x) = a + b\,x + c\,\cos(3x) + d\,e^x. \tag{3}$$

The data can be found in `data.mat`, which is saved in the same folder as the exam paper.

2.1) Set up the normal equation for calculating the coefficients a, b, c and d, and state both the system matrix and the right-hand side in the normal equation.

2.2) Use Matlab to solve the normal equation, and state the solutions of a, b, c and d with 4 significant digits.

2.3) Use Matlab to calculate the sum of the absolute error $|y_k - F(x_k)|$ for $k = 0, \ldots, 25$ and state it with 3 decimal places.

2.4) Now we want to use the function $F(x)$ in (3) to fit several different data sets. For each data set, the x-values are the same, but the y-values are different, so we will obtain different a, b, c and d by solving the corresponding normal equation. Can we use the factorization of the system matrix of the normal equation to speed up the process? Explain your answer.

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

Consider a system of nonlinear equations $\boldsymbol{F}(\boldsymbol{X}) = \mathbf{0}$ with

$$\boldsymbol{X} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad \text{and} \qquad \boldsymbol{F}(\boldsymbol{X}) = \begin{bmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{bmatrix} = \begin{bmatrix} x_1^2 + x_2^2 - 6x_1 - 2x_2 + 1 \\ 4x_1^2 + 9x_2^2 - 8x_1 + 18x_2 - 23 \end{bmatrix}.$$

We use Newton's method to find the solution which is the most distant from the origin.

3.1) Derive the Jacobian matrix $\boldsymbol{F}'$ for the vector function $\boldsymbol{F}$, and write a Matlab function `funFdF_1` starting with:

```
[F,dF] = funFdF_1(X)
```

It returns a vector `F` with the values of $\boldsymbol{F}(\boldsymbol{X})$ and a matrix `dF` with the values of the Jacobian matrix $\boldsymbol{F}'(\boldsymbol{X})$. Include your Matlab code in the pdf file with your answers.

3.2) State a starting point $\boldsymbol{X}^{(0)}$ that gives the solution which is the most distant from the origin, and explain how you chose it. (Hint: In Matlab, `ezplot` is a function plotter. Example: You can easily plot the equation $xy - 2x + 1 = 0$ by calling `ezplot('x*y - 2*x + 1')`.)

3.3) Run 10 iterations of Newton's method with the starting point $\boldsymbol{X}^{(0)}$ from Question 3.2, and state your final solution $\boldsymbol{X}^{(10)}$. For this question, you will need the Matlab function `Newtonsys.m`, which is saved in the same folder as the exam paper.

Question 4. Polynomial approximation of a function (10%)

In this question, we will approximate the function $f(x) = (x+3)e^x$ on the interval $I = [0, 6]$ and study the approximation error. We know that the mth derivative of f is $f^{(m)}(x) = (x + 3 + m)\, e^x$.

4.1) Based on the second interpolation error theorem, how accurate will the interpolation be, if we use a polynomial of degree at most 16, named as p_{16}, on $16 + 1$ equally spaced nodes to approximate the function f over I?

4.2) Test numerically by taking 300 equally spaced points in the interval I and plot the error $f(x) - p_{16}(x)$. Include the plot in the pdf file with all your answers. For this question, you will need the Matlab function `LagrangeFormInterpolation.m`, which is saved in the same folder as the exam paper, or you can use your own implementation.

Question 5: Initial-value problem (20%)

We consider the initial-value problem

$$
\begin{aligned}
x'' &= x^2 - y + e^t \\
y'' &= x - y^2 - e^t \\
x(0) = 0, &\quad x'(0) = 0 \\
y(0) = 1, &\quad y'(0) = -2
\end{aligned}
\tag{4}
$$

5.1) We introduce the new variables $z_1 = x$, $z_2 = y$, $z_3 = x'$, and $z_4 = y'$. Rewrite the problem (4) into a system of first-order differential equations in the variables $Z = [z_1, z_2, z_3, z_4]^T$, and state the initial condition for Z at $t = 0$.

5.2) Implement a Matlab function that returns the right-hand side of the system for Z. Include your Matlab code in the pdf file with your answers.

5.3) Apply the Runge-Kutta method of order 4 to solve this system of differential equation in the interval $[0, 2]$. Use $h = 2^{-k}$ with $k = 4$ and 5.

- State the number of steps n and the number of function evaluations n_f for $k = 4$ and 5, respectively.

- State the results of $(x(2), y(2))$ at $t = 2$ for $k = 4$ and 5, respectively.

You will need the Matlab function `MyRK4System.m`, which is saved in the same folder as the exam paper, or you can use your own implementation.

END.