Problem Set 5 - Complete version

Unless otherwise specified, you may use MATLAB to assist with computations. Please provide a print-out of the code used and its output with your assignment.

1. Positive definiteness.
   (a) Without doing any calculations, determine whether the following matrices are positive definite (and explain your reasoning):

   $A_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad A_2 = \begin{bmatrix} -1 & 8 & 3 \\ 1 & 4 & -7 \\ -2 & -3 & 6 \end{bmatrix}, \quad A_3 = \begin{bmatrix} 4 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 9 \end{bmatrix},$

   (b) Show that if a $n \times n$ symmetric matrix $B$ is not full rank, it cannot be positive definite.

   (c) Show that if $C$ is an $n \times n$ matrix such that $u^T C u > 0$ for every $u \neq 0$, then all eigenvalues of $C$ are positive.

   (d) Show that if the symmetric $n \times n$ matrices $D_1$ and $D_2$ are positive definite, then so is $D_1 + D_2$.

2. SVD. This problem is to be done by hand. Find the singular value decomposition of the following matrices:

   (a) $A = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}$,

   (b) $B = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}$,

   (c) $C = \begin{bmatrix} 6 & 3 \\ 8 & 4 \end{bmatrix}$.

   (Hint: Use the result from part (a))

3. Error. Consider the matrix:

   $A = \begin{bmatrix} 1.2143 & 0.8054 \\ 0.8128 & 0.5391 \end{bmatrix}$.

   (a) Set $b_1 = [1.2143; 0.8128]$ and compute $x = A \backslash b_1$ in MATLAB.

   (b) Repeat the process but with a vector $b_2$ obtained from $b_1$ by rounding each value to three decimal places. Why do the two answers differ by so much?

   (c) Now compute $x = B \backslash b_1$ where $B$ is obtained from $A$ by rounding each value to three decimal places. As you will see, our solution is also sensitive to small changes in the matrix $A$. 
(d) Errors from ill-conditioned problems are not unique to solving \( Ax = b \). As an example, compute \( \tan(10^{100}) \) three ways

i. Google: search for \( \tan(10^{^100}) \)

ii. MATLAB: \( \tan(10 \cdot 100) \)

iii. MATLAB: Use \( \text{vpa}('\tan(1e100)', p) \) to compute with \( p \) digits of precision for \( p = 5, 10, 25, 50, \) and 100.

Explain the results you see. Why is this a hard problem to solve?

4. Condition number. Suppose the condition number of an invertible \( n \times n \) matrix \( A \) is \( \kappa \).

(a) Find the condition number of \( A^\top A \) in terms of \( \kappa \). What does this imply about solving the normal equations by inverting \( A^\top A \)?

(b) Find the condition number of \( A^{-1} \) in terms of \( \kappa \).

(c) What is the value of \( \kappa \) if \( A \) is an orthogonal matrix?

(d) What is the smallest possible value for \( \kappa \)? Construct a \( 3 \times 3 \) matrix with this minimum condition number.

5. Initial Value Problem. Consider the following initial value problem for \( x(t) \):

\[
\frac{dx}{dt} = x^2
\]

with initial condition \( x(0) = -1 \).

(a) Find the exact solution for \( x(t) \). (Hint: Separate variables and integrate both sides). Compute the exact value of \( x(1) \).

(b) We would like to setup this problem to be solved numerically with a time step \( \Delta t \). Recall that \( x_n = x(n\Delta t) \). Find an explicit formula for \( x_{n+1} \) as a function of \( x_n \) and \( \Delta t \) using a Forward Euler method.

(c) Implement this explicit numerical method in MATLAB and find the approximation of \( x(1) \) using \( \Delta t = \frac{1}{10} \). Calculate the difference between the exact value and this approximate value.

(d) If we reduce the time step from \( \Delta t = \frac{1}{10} \) to \( \Delta t = \frac{1}{100} \), how much do you expect the error to be reduced?

(e) Find the approximation of \( x(1) \) with \( \Delta t = \frac{1}{100} \). Calculate the difference between the exact value and this approximate value. By what factor is the error at \( x(1) \) reduced by switching from \( \Delta t = \frac{1}{10} \) to \( \Delta t = \frac{1}{100} \)?

(f) Derive an explicit formula for \( x_{n+1} \) as a function of \( x_n \) and \( \Delta t \) using a:

i. Backward Euler method,

ii. Trapezoidal Rule method.
6. The exact solution of the differential equation

\[ \frac{dx}{dt} = -x + \cos t, \tag{1} \]

with the initial condition \( x(0) = x_0 \) is \( x(t) = (x_0 - \frac{1}{2})e^{-t} + \frac{1}{\sqrt{2}} \cos (t - \pi/4) \). Implement the Forward Euler method in MATLAB to solve this equation numerically. Use three different time steps: \( \Delta t = 0.1, \Delta t = 0.5 \) and \( \Delta t = 1.0 \). In a single plot, show the exact solution and your three numerical solutions for \( x_0 = 0 \).

7. Newton’s method (This problem is to be done by hand.)

(a) Use two steps of Newton’s method to approximate \( x^* = \sqrt{3} \) starting from an initial guess \( x_0 = 2 \). How many correct digits are there at each step? How many iterations do you expect it to take to find \( x^* \) to sixteen digits (the limit of MATLAB’s double-precision arithmetic)?

(b) Consider solving the system of equations

\[ \frac{dx}{dt} = 1 - txy \]
\[ \frac{dy}{dt} = 1 - ty^2 - x \]

with initial condition \((x(0), y(0)) = (1, 1)\). Set up the system of equations for the backwards Euler approximation to \((x(1), y(1))\) with a timestep of \( \Delta t = 1 \). Use the forward Euler approximation as your initial guess and improve it with one step of Newton’s method.

(c) A common place Newton’s method shows up is in finding minima of functions. Use it to find the minimum of

\[ f(x, y) = x^2 + y^2 + xy - 3x - 3y \]

near \((x, y) = (0, 0)\). How many steps do you need? Why? (Hint: what do you want to set to zero?)