CS371/AMATH242 Winter 2019: Assignment 3

Instructor: Maryam Ghasemi

Due Sunday, March 24, 11:59 pm via Crowdmark

---

1. [12 marks] In the following parts of this question, write a MATLAB code to solve a linear system $Ax = b$ (A is a square nonsingular matrix) using Jacobi and Gauss-Seidel algorithms. Do not use the built-in Matlab functions for solving linear systems.

   (a) Write a Matlab function called Jacobi that consumes a square $n \times n$ matrix A, and an $n \times 1$ vector b, and uses the Jacobi technique to solve the system $Ax = b$, starting with the zero vector as the initial guess. Your function should stop when norm-2 of the difference between two successive iterates is less than 10^{-6}, or if your function has performed 1000 iterations without reaching the termination condition. Your function must produce the value of x when the function terminates, along with the number of iterations performed to that point.

   (b) Write a Matlab function called Gauss-Seidel, which meets all the same specifications as the function Jacobi described in part a), except that it uses the Gauss-Seidel technique to solve $Ax = b$.

   (c) Consider the following $n \times n$ system, for values of $n > 1$:

   $$\begin{bmatrix} n & 1 & \cdots & 1 \\ 1 & n & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

   Solve this system using the Jacobi and Gauss-Seidel functions for the following values of $n : 5, 250, 1250$. For this part of the problem, you may calculate the exact solution as $x_e = A/b$, to use in error calculation below.
   Produce a table of results, where each row contains the following information:

   - the value of n
   - the relative error in the Jacobi solution x_j compared to x_e, for n
   - it_j, the number of iterations required by Jacobi for n
   - the relative error in the Gauss-Seidel solution x_g compared to x_e, for n
   - it_g, the number of iterations required by Gauss-Seidel for n

   (d) Compare these techniques in terms of number of iterations and runtimes. To measure runtimes accurately, run both few times (say 10), disregarding the time for the first run, and take the average time (you may use tic,toc function to measure runtime).

2. [6 marks] Consider the three data points: (-2,0.25), (0,1), (2,4).

(a) Find the quadratic Lagrange polynomial that passes through each of the points. Be sure to show the individual Lagrange functions as well. Use your interpolating function to predict the value at x=-1.

(b) Find a piecewise linear interpolating function that passes through all three points. Use your interpolating function to predict the value at x=-1.

(c) Draw a simple plot (with x in the range [-2,2]) illustrating the above "fits": quadratic Lagrange function, and the piecewise linear function. You may draw the plot by hand or by using Matlab.

3. [9 mark] Suppose you want to construct a look-up table for $f(x) = x + ln(x)$ for x in [2,8], that is, you want to produce a table listing the actual values of $x + ln(x)$ at point $2, 2 + h, 2 + 2h, \cdots, 8$ for some fixed step-size h. When someone wants the value of f(x) for a value in this range that is not stored in the table, they will use linear Lagrange interpolation over the range containing the value of interest.

(a) Write a Matlab function *lookup* that consumes a step-size, h, and produces two vectors, x_{ref} and y_{ref}, both of length N, where

- x_{ref} contains the reference points $2, 2 + h, 2 + 2h, \cdots, 2 + Nh$
- y_{ref} contains the values of $x + ln(x)$ at the points $2, 2 + h, 2 + 2h, \cdots, 2 + Nh$

(b) Write a Matlab function LL_{interp} that consumes vectors of the form of x_{ref} and y_{ref}, and a point x (between 2 and 8), and produces the interpolated value of $x + ln(x)$, using linear Lagrange interpolation and the values in x_{ref} and y_{ref}.

(c) Write a Matlab program that uses your functions *lookup* and LL_{interp} to estimate the value of $x + ln(x)$ at 100 random values in the range [2,8]. Determine the average deviation (measured as the absolute value of the difference between the value $x+ln(x)$ and the interpolated value) over all the random evaluations. What average deviation did you observe? Do you expect the same average deviation each time you run your code? Why or why not? You may consider $N = 60$.
Note that you can create a vector of 100 values in the range [a,b] with the Matlab command: $a + (b - a) * rand(100, 1)$.

3. [3 marks] Consider the definite integral $\int_3^{3.5} \frac{x}{\sqrt{x^2-4}}\, dx$. The exact value of definite integral is 0.63621.

(a) Use the Midpoint method to approximate, and calculate the absolute error. State your answer to 4 significant digits.

(b) Use the Trapezoid method to approximate, and calculate the absolute error. State your answer to 4 significant digits.

(c) Use Simpson's method to approximate, and calculate the absolute error. State your answer to 4 significant digits.