Homework 4 Introduction to Computational Finance Spring 2023

Solutions due Friday March 31, 2023

Answers to the homework problems and programming tasks should be submitted using the class canvas page.

You should submit pdf files. **Do not sent Word files or any other text processing tool's input file**.

As with all homework assignments you are allowed and encouraged to consult the relevant literature. You are also expected **to cite all literature that is used to generate your solutions and your solutions must make clear your understanding of the work cited.**

Programming Assignment

Codes

1. Implement the LU factorization of a strictly diagonally dominant tridiagonal matrix and the associated routines to solve $Ly = b$ and $Ux = y$ using the resulting factors to solve $Tx = b$. Recall, a tridiagonal matrix has 0 everywhere except on the main diagonal, i.e., elements $\alpha_{i,i}$, and the first subdiagonal and superdiagonal, i.e., elements $\alpha_{i+1,i}$ and $\alpha_{i,i+1}$ respectively. T is said to be strictly diagonally dominant by rows if the magnitude of the diagonal element in each row is greater than the sum of the magnitude of the rest of elements in the same row. In this case, $|\alpha_{ii}| > |\alpha_{i,i+1}| + |\alpha_{i+1,i}|$. The definition can also be applied to columns. For the spline code the tridiagonal matrix using s_i'' as parameters with boundary conditions specifying s_0'' and s_n'' is strictly diagonally dominant in both row and column senses.

2. Implement an interpolatory cubic spline algorithm that accepts data on a **nonuniform mesh**. (Of course, this can then be used to specify a uniform mesh also.) You should use the parameterization of $s(x)$ in terms of the s_i''. Your code should support the boundary conditions where the pair of values $(s''(x_0), s''(x_n)) = (c_0, c_n)$ are specified.

3. Implement a code that evaluates the spline $s(x)$ produced by your code from the previous part of the problem.

4. Implement the supporting code required to empirically validate the correct functioning of your codes and the tasks below.

Tasks

Task 1

Empirically validate the correct functioning of your codes.

- You must design experiments and describe the outcomes that provide evidence that your code is working. Evidence must also be given that you code works correctly for each of the required boundary conditions.

- For the LU factorization, the simplest set of tests is to generate a tridiagonal strictly diagonally dominant matrix, $T \in \mathbb{R}^{n\times n}$, select a vector $x \in \mathbb{R}^n$ to be the solution of the system, and finally compute the corresponding righthand-side vector $b = Tx$ since you have selected T and x. This should be done for several systems of various dimensions n. Compare the computed solution $\tilde{x}$ to the true solution x using the relative error $\|\tilde{x} - x\|_2/\|x\|_2$. (Make sure you generate an x that has $\|x\|_2 \geq 1$.) Additionally, given T you can compare the product of the computed L and U matrices to T, i.e., $\|(LU) - T\|_F/\|T\|_F$, where the matrix norm is the Frobenius norm:

$$\|A\|_F^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_{ij}^2$$

  where the elements of $A \in \mathbb{R}^{n\times n}$ are denoted α_{ij}.

- For the spline code, this should include running your code to approximate carefully selected functions $f(x)$ for which the results are known. For example, if $f(x)$ is any polynomial of degree $d \leq 3$, the spline, $s(x)$, should reproduce its value at any x. Similarly, if $f(x)$ is a piecewise cubic with intervals defined by the same mesh as $s(x)$ then $s(x)$ should match $f(x)$ for any x between x_0 and x_n.

- Once this has been done for specific functions where $s(x) = f(x)$, move on to parameterized functions like the rational functions seen when discussing Runge's phenomenon and observe the convergence behavior as the mesh becomes finer. The rates are given in the notes but they can be difficult to observe due to the high order. However, at the very least some observations of convergence must be made by approximating

$$\|f(x) - s(x)\|_\infty = \max_{a\leq x\leq b} |f(x) - s(x)|.$$

  The approximation is done by evaluating the difference on a fine mesh of points, z_j, $0 \leq j \leq m$ where $m \gg n$,

$$\|f(x) - s(x)\|_\infty \approx \max_{z_j} |f(z_j) - s(z_j)| = E.$$

  This is done for a series of $s(x)$ produced by using an increasing number of intervals to define the spline where the maximal interval size is going to 0. You should use a family of uniform meshes and also get more sophisticated in your choice of meshes, e.g., using nonuniform intervals with a size distribution chosen based on the shape of $f(x)$. In any case, covergence to roundoff should be observed.

- For all problems where $f(x)$ is known use $(s''(x_0), s''(x_n)) = (c_0, c_n) = (f''(x_0), f''(x_n))$, i.e., you will have to determine $f''(x)$.

Task 2

Suppose, in practice $y(t)$ is available as data in the form of discrete values of (t_i, y_i), for $0 \le i \le n$ rather than as a continuous function. Specifically, consider the following data (t_i, y_i):

t_i	0.5	1.0	2.0	4.0	5.0	10.0	15.0	20.0
$y(t_i)$	0.0552	0.06	0.0682	0.0801	0.0843	0.0931	0.0912	0.0857

Note the mesh in t is nonuniform.

Use a natural boundary condition interpolatory cubic spline, $s(t)$, i.e., $s''(x_0) = 0$ and $s''(x_n) = 0$ based on the data (x_i, y_i) to estimate $y(t)$ for other x. Tabulate your estimates from $t_1 = 0.5$ to t_{40} with increment $\Delta t = 0.5$.

Written Exercises

There are no written problems in this assignment.