Project 3: Resistor networks / Cubic spline interpolation

OVERVIEW

In this project, you will write a program to automatically solve two engineering tasks which can be represented as systems of linear equations. The challenges here are a) figuring out how to algorithmically map a problem instance to a linear system of the form $Ax = b$, and b) exploring linear system solvers that we didn’t explicitly cover in class.

TASKS

1. Resistor networks. Below is depicted a simple electrical circuit, along with a corresponding description of it in a subset of the netlist format used by the SPICE circuit simulator.

![Resistor Network Diagram]

\[
\begin{align*}
V_1 & 1 0 12 \\
R_1 & 1 2 1000 \\
R_2 & 2 0 2000 \\
R_3 & 2 0 2000
\end{align*}
\]

The `parse_netlist.py` code provided on the course website is capable of parsing the netlist format above into a set of Python data structures. Your job is to transform the output of that function into a linear system using Modified Nodal Analysis (MNA), which can subsequently be solved for the voltages across the resistors and the currents through the voltage supplies.


Next, write a function \((A, x, b) = \text{solve_netlist}(V, I, R, N)\) which transforms the output of `parse_netlist` into a solved linear system using MNA. To solve
the underlying matrix equation, your program should use the Gaussian elimination matrix solver that you or your partner created for Homework 4.

- Use your program to verify the solutions of the examples at http://www.swarthmore.edu/NatSci/echeeve1/Ref/mna/MNA4.html. Submit your netlists and program output.

- Under what circumstances is MNA incapable of finding a valid solution? Are there any nets for which MNA can provide multiple valid solutions? Provide example netlists to support your answers.

2. Cubic spline interpolation. In the olden days of Engineering design, a spline was a thin, springy strip of flexible rubber, wood, or metal which could be constrained to pass through specific points on a schematic drawing in order to trace out smooth curves, as shown in figure 1. Analogously, we can mathematically define smooth curves which pass through (or interpolate) two or more points. The resulting mathematical objects are also called splines.


Next, create a program which takes as input a list of \( n+1 \) floating point numbers \( y(t) \) for integer values \( t = 0, \ldots, n \), and outputs the tridiagonal matrix of coefficients \( A \) and right-hand side \( b \) for the linear system described in the MathWorld article.

Finally, solve the linear system both with your own solver, as well as with the scipy.linalg.solve_banded method, and make sure the solutions agree.
Also, please consider the following:

- Choose a set of \( n \geq 8 \) points \( y(t) \) and plot the resulting piecewise cubic curve using \texttt{matplotlib} after solving the tridiagonal system.

- Explain how you might extend the system you have here to solve for a curve passing through \( n \) points in the plane by reusing the \( A \) matrix and simply supplying two different \( b \) vectors for the \( x \) and \( y \) coordinates. Note that the final, 2D example in the Wikipedia page actually does a horrible job of this. It specifies \( y \) as a function of \( x \); in general, we would prefer to specify \( x \) and \( y \) as functions of a common independent variable \( t \). Why is this more general than Wikipedia’s approach?

**WHAT TO TURN IN**

You should submit all of your source code, and a PDF writeup containing the outputs of your programs and which addresses the questions at the end of each task outlined above.