

Homework Project #3

Math 365

1. (**Fibonacci Series.**) Linear systems often arise in the context of solving recurrence relations. For example, the famous Fibonacci sequence, named after the 13th century Italian mathematician Fibonacci, is generated from the recurrence relation

$$F_{j+1} = F_j + F_{j-1}, \quad j = 1, 2, 3, \dots$$

Starting values for generating the sequence are given as $F_0 = 0$ and $F_1 = 1$, so that the first few numbers in the sequence are 0, 1, 1, 2, 3, 5, 8, ...

We can easily generate this sequence using a `for-loop`, but for this problem, we are going to solve for the F_j by solving a linear system of the form $A\mathbf{F} = \mathbf{b}$, where A is a square matrix, \mathbf{F} is a vector of solution values F_j and \mathbf{b} is a vector of starting values.

- Construct a linear system that relates F_j , $j = 2, 3, \dots, 10$ to known starting values F_0 and F_1 .
- Using the starting values $F_0 = 0$ and $F_1 = 1$, solve the linear system for the unknown values F_j , $j = 2, 3, \dots, 10$.
- Using the same recursion relation, generate a second sequence G_j , using starting values $G_0 = 1$, $G_1 = 4$.
- Produce a table of values for F_j and G_j , for $j = 0, 1, 2, \dots, 10$.

Hint: Use the recursion relation to write out equations for the unknowns F_2, F_3, \dots, F_{10} . with unknowns on the left hand side of each equation, and known values F_0 and F_1 on the right hand side. Construct the matrix A using either `diag` or `spdiags`.

2. (**Electric circuits**) Find the currents \mathbf{y} along each edge and the voltages \mathbf{x} at each node of the circuit shown in Figure 1. To do this, you will need to set up an “adjacency matrix” A , a diagonal “capacitance matrix” C , and any voltage (i.e. battery) sources \mathbf{b} .

To find the currents and voltages, we use Ohm’s Law and Kirchhoff’s Current Law to obtain the equations for the node voltages and edge currents.

$$\begin{aligned} \mathbf{y} &= C(\mathbf{b} - A\mathbf{x}), & (\text{Ohm's Law}) \\ A^T\mathbf{y} &= \mathbf{0}, & (\text{Kirschhoff's Law}) \end{aligned}$$

Substituting the expression for \mathbf{y} from the first equation into the second equation, we can eliminate \mathbf{y} from the equations to obtain the following system for node values \mathbf{x} .

$$A^T C A \mathbf{x} = A^T C \mathbf{b}.$$

Solve this equation for the nodes voltages \mathbf{x} . Then, use Ohm’s Law to compute the edge currents \mathbf{y} . Use `write_file` to write your voltage values \mathbf{x} to the file `voltages.out` and your current values \mathbf{y} to the file `currents.out`. For this problem, you will use the Matlabcode backslash “`\`” operator. Do **not** use `inv(A)` or `A^(-1)` !!!

Check your solution : The voltage at node 1 is 9.771 V and the current along edge 7 is 14.393 mA (1 mA = 0.001 A).

More details on this problem can be found in the linear algebra text *Linear Algebra and its Applications*, by Gilbert Strang. The relevant chapter is posted on the course website.

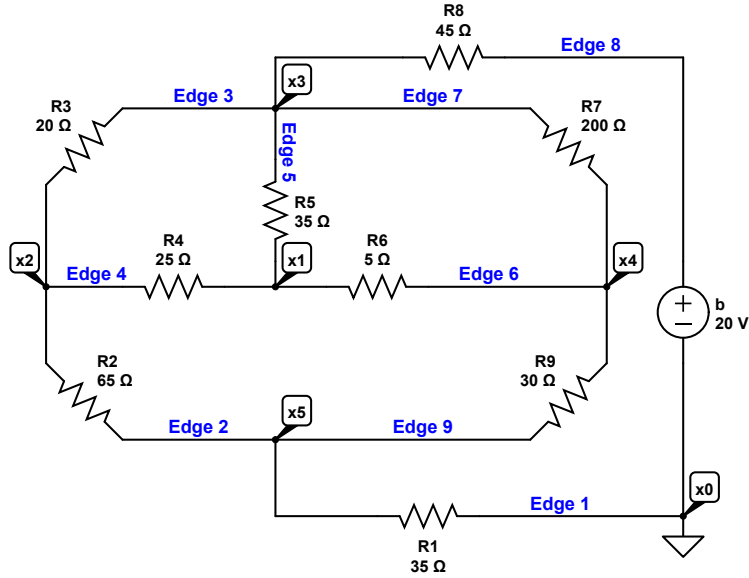


Figure 1: Circuit for Problem 2

3. **(Temperature conductivity)** This problem illustrates a common situation in which you will need to solve very large linear systems. Imagine that we have a metal rod of length L (m) whose temperature at each end is held fixed at $20C^\circ$ and which is being heated by a flame held at the midpoint. We can approximate the steady state temperature at equally spaced points along the rod using the following model. We divide the rod into N intervals of equal length. This partitioning of the rod gives us $N + 1$ equally spaced points

$$x_j = hj, \quad h = \frac{L}{N}, \quad j = 0, 1, 2, \dots, N$$

The steady state temperature T_j (K°) at interior points x_j in the rod can be modeled using a system of equations given by

$$T_{j-1} - 2T_j + T_{j+1} = -h^2 f(x_j), \quad j = 1, 2, \dots, N - 1$$

The flame $f(x)$ is given by

$$f(x) = \frac{S}{\rho c_p \beta} \exp\left(-\left(\frac{x - x_0}{\varepsilon}\right)^2\right)$$

where

- S is the heating rate for the flame (W/m^3)
- β is the thermal diffusivity of the metal (m^2/s)
- c_p is the heat capacity of the metal ($J/(kg \cdot K^\circ)$)
- ρ is the density of the metal (kg/m^3)
- ε is a scaling factor specifying the flame width (m)
- x_0 is the position of the flame (m)

Set up a linear system to solve for the steady state temperature distribution in the rod of length $L = 1$. Solve the system for a thermal conductivity of $\beta = 10^{-3}$, heat rate $S = 1 \times 10^4$, heat capacity of $c_p = 200$, density of $\rho = 10$ and a scaling factor $\varepsilon = 5 \times 10^{-2}$. The flame is located at $x_0 = L/2$.

For this problem, use `linspace` to create an array of x values and the `spdiags` command to construct the matrix that you need to solve for T .

Check your answer: Your solution for T_j for $j = 5$ should be

$$T(5) = 31.0778365681593058$$

- (a) Compute the result at equally spaced points $x_1 = 0.01, x_2 = 0.02, \dots, x_{N-1} = 0.99$ on the rod. Plot the results and be sure to add labels and a legend to your graph. Write your solution T to the file `temp.out`.
- (b) In steady state, the flux of heat in or out of the endpoints of the rod should be balanced exactly by the heat that the flame is adding to the rod. Mathematically, we have that

$$T'(L) - T'(0) = \int_0^L f(x)dx \quad (1)$$

We can show that this holds in our case by using finite difference approximations to the derivatives (or *fluxes*) of $T(x)$ at the endpoints as

$$T'(0) \approx \frac{T_1 - T_0}{h}$$
$$T'(L) \approx \frac{T_N - T_{N-1}}{h}$$

where T_0 and T_N are the end point temperatures that we imposed. We can approximate the integral as

$$\int_0^L f(x)dx \approx h \sum_{j=1}^{N-1} f(x_j) \quad (2)$$

- i. What fact from Calculus I assures us that (1) holds for the equation

$$T''(x) = f(x)$$

over the interval $[0, L]$? Answer this in your Matlab script.

- ii. Use these approximations in (2) to compute the fluxes and total heat input, and show that the difference of the fluxes balances the total heat added to the rod. Use `fprintf` to print both your fluxes and integral approximation to 6 decimal places. Print the difference using scientific notation in your `fprintf` statement.