

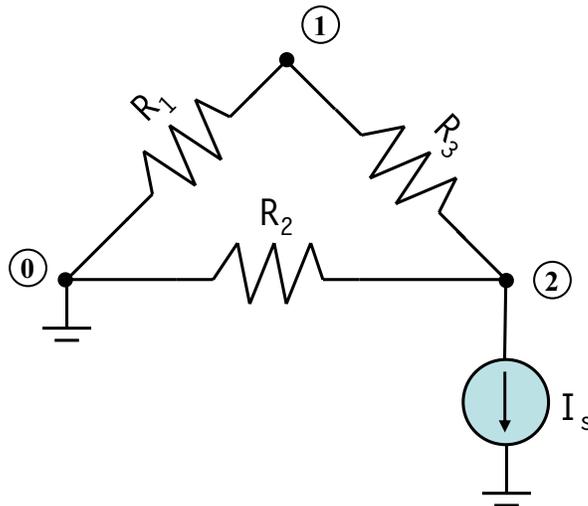
Introduction to Numerical Simulation (Fall 2003)
Problem Set #1 - due September 12

This problem set is mainly intended to familiarize you with the algorithms associated with formulating a system of equations from a given problem description, and also to remind you of eigenvalues and eigen vectors. We have used circuits and heat flow examples, and will use struts and joints later in the term when we discuss nonlinear systems.

In problems (2) and (3), you will be using Matlab and modifying scripts that we have written for you. You can download the files from the course web page.

1) This problem is intended to reinforce your understanding of the nodal analysis and node-branch equation formulation techniques.

Consider the following simple circuit.



a) Apply nodal analysis to generate a linear system of equations which can be used to compute the circuit node voltages. Where appropriate, please give matrix or vector entries as analytical formulas in terms of R_1 , R_2 , R_3 and I_s .

b) Use the node-branch approach to form a linear system of equations which can be used to compute the circuit node voltages and resistor currents. Where appropriate, please give matrix or vector entries as analytical formulas in terms of R_1 , R_2 , R_3 and I_s .

2) In order to help you better understand how to create a program which reads in schematics and generates systems of equations, we have written a set of matlab functions and scripts which can be used to read in a file containing a circuit description and set up the associated nodal analysis

matrix and the appropriate right-hand side. In this problem, you will modify these functions and scripts, so bear with us while we describe what they do.

The matlab script assumes that the circuit description is given as a list of statements in a file, and that the element types are described in the following format.

resistors:

$$rlabel \quad node1 \quad node2 \quad val$$

where *label* is an arbitrary label, *node1* and *node2* are integer circuit node numbers, and *val* is the resistor's resistance (a floating point number).

current sources:

$$ilabel \quad node1 \quad node2 \quad val$$

where *label* is an arbitrary label, *node1* and *node2* are integer circuit node numbers, and *val* is the current flowing from *node1* to *node2*.

voltage sources:

$$vlabel \quad node1 \quad node2 \quad val$$

where *label* is an arbitrary label, *node1* and *node2* are integer circuit node numbers, and *val* is the voltage between *node1* and *node2*.

voltage-controlled current sources:

$$clabel \quad node1 \quad node2 \quad node3 \quad node4 \quad val$$

where *label* is an arbitrary label, *node1*, *node2*, *node3*, and *node4* are integer circuit node numbers, and *val* is the controlled source's transconductance (denoted g_m), which means that the current flowing from *node1* to *node2* is *val* times the voltage difference between *node3* and *node4* (i.e. $i = g_m(v_3 - v_4)$). Note, ground is always node 0.

To use the supplied scripts to solve our supplied example circuit file, test.ckt, first start up matlab. When in matlab, type $file = 'test.ckt'$ to specify the name of the input file. Then type *readckt* to run the script in the file readckt.m. This will read the file and put the information in arrays. Finally, type *loadMatrix* to run the function in loadMatrix.m; this will create the conductance matrix and the right-hand side associated with the circuit in test.ckt. To determine the vector of node voltages, type $v = G \backslash b$.

The scripts we have provided have an implementation of nodal analysis for resistors, current sources and voltage-controlled current sources. Your job will be to extend the implementation to include voltage sources for the special case where one terminal of the voltage source connected to ground. To accomplish this, you will need to modify only the file loadMatrix.m.

You could, of course, switch the simulator so that it uses node-branch analysis. However, for networks of two-terminal positive resistors, the conductance matrix has a structure that is advantageous for numerical calculations (positive diagonals, negative off-diagonals, symmetry, diagonal dominance). Assuming that one terminal of the voltage source is connected to ground, it is possible to implement voltage sources in your Matlab simulator in such a way that you will still preserve the

above properties of the conductance matrix. Please implement such a scheme in your Matlab simulator. Only implement what is needed for resistors, current sources, and grounded voltage sources, do not bother with voltage-controlled current sources. Be sure to test your modified simulator, you will need a working version for problem (3).

Some Helpful Notes.

It will be necessary to make contributions to the right-hand side (RHS) vector for each resistor that is connected to a voltage source, and the matrix will require modifications as well. In calculating these modifications, you will find the array, *sourcenodes* generated by *readckt* helpful.

To make the simulator easier to debug, the node numbers in input files correspond to the row numbers in the generated conductance matrix. If the input file node numbers are not contiguous, or if voltage source nodes are not last, there will be rows in the resulting G matrix with only a single one on the diagonal.

3) In this problem we will examine the heat conducting bar basic example, but will consider the case of a “leaky” bar to give you practice developing a numerical technique for a new physical problem.

With an appropriate input file, the simulator you developed in problem 2 can be used to solve numerically the one-dimensional Poisson equation with arbitrary boundary conditions. The Poisson equation can be used to determine steady-state temperature distribution in a heat-conducting bar, as in

$$\frac{\partial^2 T(x)}{\partial x^2} = \frac{\kappa_a}{\kappa_m} (T(x) - T_0) - \frac{H(x)}{\kappa_m} \quad (1)$$

where $T(x)$ is the temperature at a point in space x , $H(x)$ is the heat generated at x , κ_m is the thermal conductivity along the metal bar, and κ_a is the thermal conductivity from the bar to the surrounding air. The temperature T_0 is the surrounding air temperature. The ratio $\frac{\kappa_a}{\kappa_m}$ will be small as heat moves much more easily along the bar than dissipates from the bar into the surrounding air.

Now suppose one is trying to decide if it is necessary to have heat sink (a heat sink is usually just a large chunk of metal which dissipates heat rapidly enough to stay close to room temperature) connections at the ends of an electronic package. You can use simulation to help you make this decision.

a) Use your Matlab simulator to numerically solve the above Poisson equation for $T(x)$, $x \in [0, 1]$, given $H(x) = 50$ for $x \in [0, 1]$, $\kappa_a = 0.001$, and $\kappa_m = 0.1$. In addition, assume the ambient air temperature is $T_0 = 350$, and $T(0) = 300$ and $T(1) = 300$. The boundary conditions at $x = 0$ and $x = 1$ models heat sink connections to a cool metal cabinet at both ends of the package. That is, it is assumed that the heat sink connections will insure both ends of the package are fixed at near room temperature. In your numerical calculation, how did you pick Δx ? How do you know your solution is accurate.

b) Now use your simulator to numerically solve the above equation for $T(x)$, $x \in [0, 1]$, given $H(x) = 50$ for $x \in [0, 1]$, $\kappa_a = 0.001$, and $\kappa_m = 0.1$. In addition, assume the ambient air temperature is $T_0 = 350$, and that $T(0)$ and $T(1)$ are unknown but $\frac{\partial T(0)}{\partial x} = 0$ and $\frac{\partial T(1)}{\partial x} = 0$. The zero heat flow

boundary conditions at $x = 0$ and $x = 1$ implies that there are no heat sinks at either end of the package. Compare your results to part (a).

c) For the case examined in part (b) above, what will happen if κ_a is identically zero. Can you solve this problem? Can you find a reasonable solution by examining what happens as κ_a approaches zero?

4) This last problem uses the simulator you developed in the previous question to remind you about some of the properties of eigenvalues and eigenvectors before we use them in lecture. Be sure to familiarize yourself with MATLAB's *eig* command before you start.

a) Consider the “leaky bar” system in question 3, assume $T_0 = T(0) = T(1) = 0$ and use thermal conductivity values in part 3a). For this system, find a heat distribution, $H(x)$, such that $\max_x H(x) = 1$ and when you solve for the temperature,

$$\lambda T(x) = H(x)$$

where λ is a real number. You need only plot $H(x)$ and $T(x)$, you do not need an analytical formula.

b) For the problem in part 4a, how many different $H(x)$ and λ 's are there?

c) Suppose $\kappa_a = 0$, but all the other settings in part 3a) hold. How do the $H(x)$'s and λ 's which satisfy

$$\lambda T(x) = H(x)$$

change? Please explain your results.