

## Introduction to EES (©2000 F. Alvarado)

EES (“ease”) is a simple equations solving environment developed by Professors Klein and Alvarado to solve linear or *nonlinear* equations of all kinds. The idea is simple: just enter your equations, then click Calculate and Solve or **F2**.

- The **F1** key provides help.
- The number of equations is virtually unrestricted.
- EES has extensive plotting capabilities. First create a “table” of solutions using the Parametrics New Table command, then plot using the Plot command. To produce a plot at least one variable must be specified in the table, not in the equations.
- EES can do optimization. At least one variable must be unspecified.
- Semicolons at the end of every equation are required if more than one equation per line, optional but recommended otherwise. Trigonometric functions such as  $\text{SIN}(x)$  are supported. Squaring is done via the  $x^2$  syntax (recommended) or the  $x**2$  syntax. Comments and text are enclosed in braces. Spaces anywhere are optional. Multiplication is indicated by  $*$ .
- EES is quite unique in that it knows the Thermodynamic properties of materials.
- If EES fails in a nonlinear problem, give initial guesses (Options Variable Info **F5**).

### A simple example

Consider the equations associated with the example below. Just enter the following equations as shown and press **F2**:

```
v_g = 10;  
i_g = 5;  
(1/5)*(v_1-v_g)+(1/40)*v_1+(1/8)*(v_1-v_2)=0;  
(1/8)*(v_2-v_1)+(1/32)*v_2=i_g;
```

MatLab is a comprehensive and easily expandable program to deal with most problems involving matrices. Features of MatLab include:

- Expandability and programmability using purchased “toolboxes” or M-files.
- Ability to solve large sets of real *or complex* linear equations.
- Excellent graphics, ability to program platform-independent graphic user interfaces.
- Of great value for control courses and signal processing courses.
- Widely accepted within Universities and Industry.

### A simple example

Consider the equations associated with the Example below. The equations for this circuit are as follows:

$$\begin{aligned} V_1 &= 10 \\ -\frac{1}{5}V_1 + \left(\frac{1}{5} + \frac{1}{40} + \frac{1}{8}\right)V_2 - \frac{1}{8}V_3 &= 0 \\ \frac{1}{32}V_1 + \left(\frac{1}{8} + \frac{1}{32}\right)V_2 - \frac{1}{8}V_3 &= 5 \end{aligned}$$

To use MatLab, first start Matlab. After you get the MatLab >> prompt, enter the “left hand side” conductance matrix **G** as follows:

$$\mathbf{G} = [1 \ 0 \ 0 \ ; \ -1/5 \ (1/5+1/40+1/8) \ -1/8; \ -1/32 \ -1/8 \ (1/8+1/32)]$$

Then enter the right hand side vector **I** as follows:

$$\mathbf{I} = [10; 0; 5]$$

Solve for the unknown left hand side voltages using the “backslash” operator:

$$\mathbf{V} = \mathbf{G} \setminus \mathbf{I}$$

Quit Matlab with `quit`.

Other useful Matlab commands include: `ls`, `dir`, `diary`, `format compact` and `disp`

## Introduction to Spice and Pspice (©2000 F. Alvarado)

Spice and Pspice are widely available programs for the simulation of electric circuits using nodal analysis. There are at least two ways to use PSpice:

- By directly creating and running a circuit description file (.CIR extension) using any ordinary text editor (such as PFE), or
- By using the schematic capture program provided by OrCAD.

The first of these approaches is described below. For the second approach refer to the textbook supplement that you should have received with your book.

The following are features you need to know about spice if you will be creating your own .CIR files:

- Pspice can solve very large networks, perform DC analysis (sometimes called bias analysis by PSPICE) as well as AC analysis, and do several kinds of plots.
- Plots using spice require the use of the auxiliary program probe.
- You may prepare the data with any ASCII editor.
- The first line of the file is a comment. Other comment lines start with \*. Anything after a semicolon ; is also a comment.
- Start by numbering all your nodes sequentially (reference is zero), then name all your components. Voltage source names start with V, current sources with I, resistors with R (examples: Vs, Rin, I17). Names are followed by two node numbers and the value of the component. You may use suffixes for units. Example: R12 4 0 15K.
- Voltage controlled voltage sources start with E and voltage controlled current sources with G. The name is followed by the component node numbers and the two number identifying the controlling quantity, followed by the value of the factor. Example: E2 5 7 3 4 10. Using parenthesis for grouping for legibility is allowed: E2 5 7 (3,4) 10.
- Current controlled current source names start with F and current controlled voltage sources with H. The name is followed by the component node numbers and the name of a *voltage source* whose *current* is the controlling quantity, followed by the value of the factor. You can think of this voltage source as an ammeter. If necessary, you can put a dummy zero-volt voltage source in series with the component that you are trying to use as a controlling current. Example: H2 5 7 V1 10.
- Dependent sources can be polynomial functions of the controlling source. Example of a current source between nodes 5 and 7 that depends on the square of the voltage between nodes 3 and 4: H1 5 7 POLY(1) (3,4) 0 1. This is a ‘tricky’ way to measure power in Pspice.

- Lines starting with a dot are commands to spice. .END indicates end of data (always required). When spice runs, it produces an output (.OUT) file. .OPT nopage prevents page breaks in the output file.
- Running probe after running spice permits plotting. In order to enable probe, you must first do a run that produces some form of a sweep using the .DC, .AC, .TRAN or other such command for spice. Any time you sweep, you need to also request output using either .PRINT or .PROBE. The .PROBE command indicates that you want to follow the simulation with the use of probe and keeps everything.
- probe permits some arithmetic upon the display of output. This permits plotting, for example, powers.

## A simple example

Consider the equations associated with the same example above. A possible file to study this circuit:

```
Example 1
Vso 1 0 DC 100
Iso 5 3 DC 5
Vamp 0 5 DC 0
R1 1 2 5
R2 2 0 40
R2 2 3 8
R4 1 3 32
.END
```

Important: The order of resistor node specifications does not matter. Current source node order is in the direction of the arrow: tail end of arrow node number first, head of arrow node number second. Voltage source node order is: plus terminal node number first, minus terminal node number second.

To sweep the voltage source from a value of 10 to 30 in increments of 2 volts and prepare the output for probe insert the following lines just before .END:

```
.DC V1 10 30 2
.PROBE
```

To request that the voltage at nodes 2, the voltage across resistor R1 and the current through resistor R3 as a result of the DC sweep be tabulated, use the syntax:

```
.DC V1 10 30 2
.PRINT DC V(2) V(R1) I(R3)
```

For more information about Spice and PSpice, refer to a book by Paul Tuinenga on Spice.