

Objectives: Continued development of MatLab programming skills. Open a sequential-access text data file, read data into Matlab storage vectors, build  $[Y_{BUS}]$ .

- 1) Continue your work with MatLab to open and read in all of the system data for the IEEE standard 14-bus load flow case. You will probably need to open the file as a text file using the 't' option.
  - a) Read the columns of bus and branch data into vector arrays in MatLab.
  - b) Build  $[Y_{BUS}]$ . Display its topology using the spy function.
  - c) Verify that you have correctly constructed  $[Y_{bus}]$ .
  - d) As a test, read in the 57-bus IEEE case. Again, show its topology with the spy function.
  - e) Reflecting on the one basic thing you know about reordering, how should the 57-bus system of equations be reordered?
  
- 2) Improve your knowledge of data file formats. We need to be VERY cognizant of file/access types, use the proper options when opening a data file, and use proper strategies to read from it and write to it. What do the following signify? (Ask about these in class. Also, feel free to ask/share what you know via the class e-mail list).
  - a) Sequential-access file - what is it?
    - End-of-line characters (10=LF,13=CR) and difference for DOS-type vs. unix-type
    - End-of-file character (26)
    - Resetting or "rewinding" to start of file
    - Effect of "Append" option when writing to
  - b) Direct-access file - what is it?
    - Record Length
    - Record Number
  - c) Binary file - what is it?
    - Pointer to start of data
    - Byte number vs. offset
  - d) Data types when reading and writing
    - Interpreting text strings as formatted numbers (real, integer, complex, etc.)
    - ASCII codes used to designate characters, ISO character sets.
    - I/O using strings as intermediaries (list-directed I/O).
    - Binary
    - Binary byte sequence - least-significant byte first, or most-significant byte first?
  
- 3) Revisit last week's assignment on the function that displays a complex number in polar form. Improve it using formatting so that it actually displays it in the form  $XXX.XX/\pm AAA.AA^\circ$ . Are there Matlab functions that allow program I/O of complex numbers in polar format?

**Coming up next:** Reordering, Newton-Raphson, more MatLab programming.