CS 105

Introduction to Scientific Computing

Topic #16 – Data Input and File I/O

Matt Burlick
Stevens Institute of Technology

ASSIGNMENT 11

- Read in formatted data from a text file
- Assume each line in the file is of the format
  (x1, y1)
- Use this data to
  - Compute the mean and standard deviation of the data
  - Plot the data

NECESSARY SKILLS

- Reading from text files
- Extracting parts of strings (last lecture)
- Using Matlab functions (for mean, std, sorting for plotting)
- Plotting

TOPICS

1. Files Types
2. Opening and Closing Files
3. Writing text
4. Reading lines of text

READING

- Appendices B.2, B.3, B.5

WAYS TO GET DATA

- Interactive
    - Input command
- Copy and paste into a variable
- Read in from a file

FILE TYPES

- Generally speaking, we can put file types into two categories:
  - Binary
  - Text
- Binary files contain *raw* data. Programs can read and write their own data
  - Like an actual array or matrix of stuff
  - Difficult if not impossible for humans to read
- Text files contain text, or characters defined by some character set (ASCII)

MATLAB BINARY DATA FILES

- You can save variable data into a MATLAB binary file using the command
  - save filename var1 var2 var3
- We can then read or *load* the binary file's data using
  - load filename
  - This will populate each of the variables saved in the file into your workspace

COMMANDS FOR GENERAL FILE TYPES

- Open
- Read/Write
- Close

OPENING FILES

- Fid = fopen(filename,permission)
  - Where filename is the **string** of the filename (maybe with the path)
  - The permission is a **string** that says what we plan on doing with it:
    - ‘r’ % read only (default)
    - ‘r+’ % read and write
    - ‘w’ % erase file and write
    - ‘w+’ % erase file for reading and writing
    - ‘a’ % append write-only
    - ‘a+’ % append read and write
  - If we want to write **text** data, a ‘t’ should be added to the end of the permission string

CLOSING FILES

- To close the file you just use the variable that's storing your file and call:
  - fclose(fid);
- It is important to close your files when you're done with them
  - Otherwise other programs may not be able to open them
  - Too many open files may bog down your system

WRITING TEXT

- To write text we can use the *fprintf* function.
  - fprintf(fid,format,x1,x2,…);
- This *format* string contains placeholders for the data
  - The data for the placeholders are x1, x2, etc..

FILE I/O FORMAT STRINGS

- Data types/placeholders:

• %c	char
• %d	decimal (integer)
• %e	exponent notation
• %f	fixed point (float/double)
• %s	string

- Special Characters:

• \n	Newline
• \t	Tab
• \b	Break
• \r	Return
• \\	Backslash
• \’’	Single quote
• %%	Percent sign

EXAMPLES

- Print out a vector as a comma separated list

READING TEXT

- A=fgetl(fid);   %read next line without newline
- A=fgets(fid);   %read next line with newline

- Returns -1 when there no more text (eof = end-of-file)

EXAMPLE

- Read in all lines from file and print out to screen

-
  ```
  fid=fopen('myfile.txt','r');
  line = fgets(fid);
  while(line~=-1)
          disp(line);
          line = fgets(fid);
  end
  fclose(fid);
  ```

READING FORMATTED DATA

- Now we can combine the prior example with string processing
- Once you get a string from a file and know its format, you can just extract parts of it like in Assignment 10
  - Comma Separated List
  - Ordered pairs, one per line
    - 4,3
    - 0,5
  - Some other format
    - Name: (Start Date, End Date)
    - Matt: (12/03/03, 01/01/14)
    - John: (05/03/01, 02/08/14)
- And there's other ways too
  - Check out *textscan* in the Matlab help and the textbook if you're interested