

CAAM420 Lecture Notes

Linh Huynh

November 13th, 2013

Some Graphics Packages

- Libraries of C/C++.
- OpenGL (Open Graphics Library), a cross-language, multi-platform application programming interface (API) for rendering 2D and 3D computer graphics.
- GUIs (Graphical User Interfaces)
- Paraview
- Gnuplot
- Matplotlib: python-based, acts like MATLAB, open source

Introduction to Python

Python is an object-oriented programming language that has many similarities to MATLAB. It is written in C and comes with Ubuntu, so you just need to go to the terminal and type “python” to open it. To get out of Python, Ctrl+D. Visit this website <http://docs.python.org/2/tutorial/> for more information. Also, install ipython (sudo apt-get install ipython) and matplotlib (sudo apt-get install python-matplotlib) by next class.

0.1 General Calculations

Now that you have had Python open, if you type a numer such as 1, then 1 will be displayed just like MATLAB. If you type 1+2, it will do the calculation for you and output 3 (just like MATLAB). However, it does not recognize sin. In order to use sine, we need the ipython package, which has all the numerical libraries. To use the ipython package, type ipython -pylab. Once you have installed the ipython package, if you type sin(3), it will output the numerical value of sin(3). The use of variables in Python is also the same as in MATLAB and C. For example, x=3 and then sin(x) will give you the same result as sin(3).

0.2 Strings

Strings in Python are just like strings in C++. You can add two strings. For example, if we have `s='string1'`, `t='string2'`, then `s+t='string1string2'`. Since Python prefers to output things inside a single quote rather than a double quote, if you want to have `'` in your string, then type `\` and then `'`. For example, `hadn\t`. In this case, your string will be in a double quote. If you want to see the string `s` without a quote, then type `"print s"`.

0.3 Parsing and Appending

Remember project 5 for which we tried to split `key=value` into `key` and `value`? No worries, we don't have to go through all the work any more. In Python, we just need to use `"split"` in a much easier way. For example, if you have a string `arg="A=B"` then just by typing `arg.split('=')` you will have `['A','B']`. Nice hah? So is it with `append`: `l=arg.split('=')` and then `l.append('3')`. Indexing with strings is just like with a vector: `l[0]='fruit'`.

0.4 Plotting and For Loop

- `x=[1,2,3,4,5]`
- `plot(x)`
- `dx=pi/10`
- `for i in range(101):`
 `x[i]=sin((i-50)*dx)/((i-50)*dx+0.00001)`
- `plot(x)`