Computer Science 102

Into to Computational Modeling

Special Topics: Programming in Matlab

Matlab

- An integrated programming and graphical environment
- *Interpreted* : interactive; get answer immediately
- Also supports saving and loading large programs
- Student version available for $100 from mathworks.com
- Free, open-source “equivalent” : Octave

`http://www.gnu.org/software/octave`

- Losing ground to Python “scipy”: numpy / matplotlib

- Questions
  1) Why Matlab vs. lower-level (C++ / Java) language?
  2) Why Matlab vs. a similar package (Mathematica, Maple) ?
  3) Why Matlab vs. Python / scipy ?

Matlab vs. C++/Java

- Abstraction: We should be able to ignore low-level details, such as memory management (required in C++)
- Expressiveness:
  - Simple (primitive) operations like add, subtract, multiply, divide, should work on entire matrices (tables) at once, without having to consider individual elements
  - Common operations like Σ (sum over elements) should be built into the language, so we don't have to code them ourselves.

Expressiveness

Generate two matrices of uniformly distributed random numbers and add them:

```
>> a = rand(3)

a =

    0.7577    0.6555    0.0318
    0.7431    0.1712    0.2769
    0.3922    0.7060    0.0462

>> b = rand(3)

b =

    0.0971    0.3171    0.4387
    0.8235    0.9502    0.3816
    0.6948    0.0344    0.7655

>> c = a + b

c =

    0.8549    0.9726    0.4706
    1.5666    1.1214    0.6585
    1.0871    0.7405    0.8117
```

Expressiveness

Where one matrix is greater than another, set the first matrix to zero:

```
>> a(a>b) = 0

a =

         0         0    0.0318
    0.7431    0.1712    0.2769
    0.3922         0    0.0462
```

Matlab vs. Maple/Mathematica

Features

Matlab: Numerical (linear algebra)

Maple, Mathematica: Symbolic (calculus)

from

`http://www.physicsforums.com/archive/topic/`

`t45208_Matlab_vs_Maple_vs_Mathematica.html:`

... **Matlab** is excellent for **manipulating data**, or dealing with any sort of **matrix calculations**..

From my experience **Mathematica** is ideal when it comes to **symbolics** such as **differentiation** and **integration**. I've seen a few cases where Mathematica blows Maple out of the water when one compares the types of integrals they can evaluate.

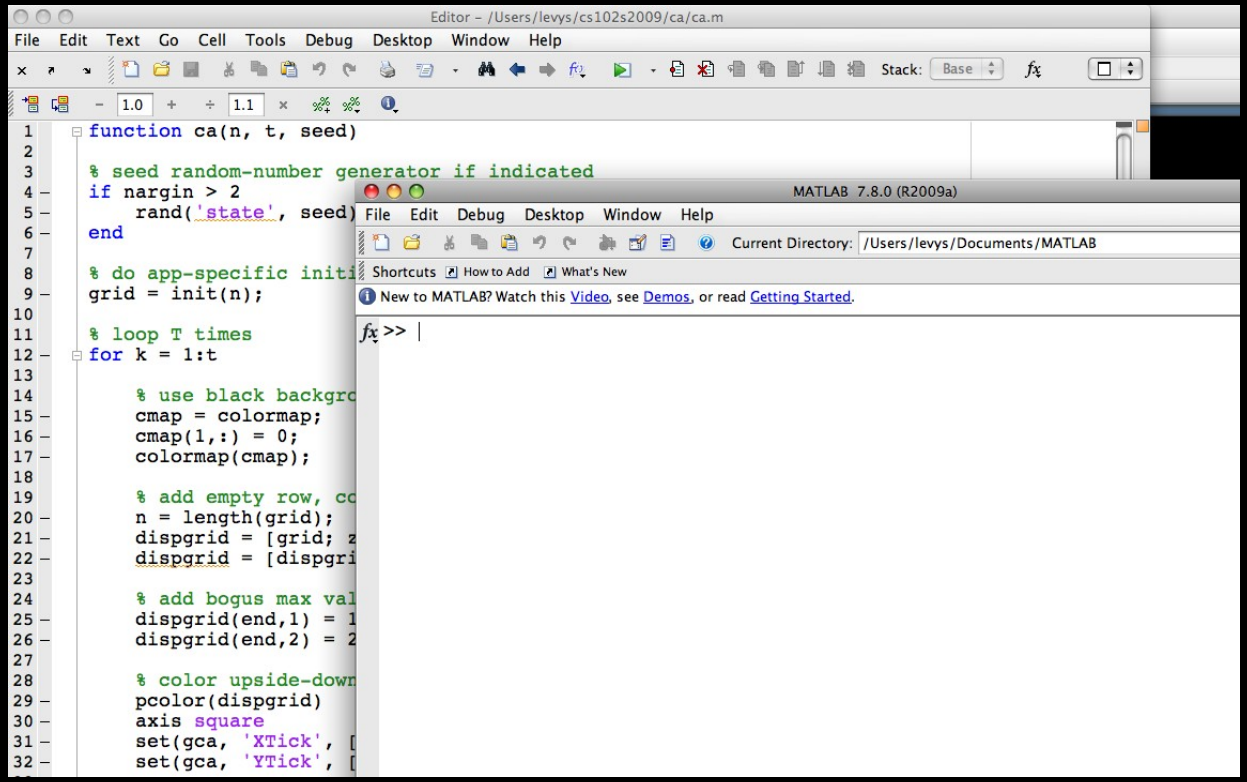
Maple and Matlab have the best **graphics** in my opinion. In both of them you are allowed to rotate 3D graphics and zoom in on the spot (2D) in realtime. In Mathematica things are little more complicated, which often [elicits] frustration. With Mathematica, in order to zoom, you must change the window that you are plotting with.

Matlab vs. Python (scipy)

- Matlab advantages:
  - Beautiful 3D graphics with minimum effort
  - Dominates in industry, many labs
- Python advantages:
  - Free, open-source
  - Hundreds of useful libraries (robotics, DNA sequencing)

The Matlab Environment

- Integrated Development Environment
- Editor + Command Window (Interpreter)



Expressions and Commands

- Anything you type into the Matlab interpreter (>> prompt) is a ***statement***.
- An ***expression*** is a statement that provides all the information needed to invoke (perform) a computation.

Expressions and Commands

- The interpreter responds by evaluating the expression
- Simplest example is a *constant* value:
  - `>> 4`
  - `ans = 4`

Expressions and Commands

- An expression typically contains
  - an operator: +, *, -, /, ^, sin, cos, sqrt
  - one or more inputs (a.k.a. *arguments*; a.k.a. *operands*)
- Two different styles (same result)
  - Functional: `plus(3, 2)`
  - Infix: `3 + 2`

Expressions and Commands

- As in arithmetic notation,
  - build compound expressions from simple expressions: `3 + 2 * 5 / sin(pi/3)`
  - use Order of Operations (PEMDAS) and parentheses to disambiguate:
  - `(3 + 2) * 5 / sin(pi/3)`

Expressions and Commands

- Good programmers don't
  - assume that others know order of operations
  - want to waste time using o.o.o. to disambiguate code they're reading
- So use parens if there's a reasonable chance of ambiguity

```
1 + 2 * 3 / 4
1 + ((2 * 3) / 4)
```

Programs should be written primarily to be read and understood by people – D. Knuth.

Changing State: Assignment

- Just as a system (health, airplane, particle) can have a state, so can a computation.
- State of computation is contents of computer's memory (RAM).
- We change this state (and hence state of system we're modeling) by ***assigning*** values to ***variables***:

```
>> altitude = 13000
>> temperature = 98.7
>> area = pi * radius^2
```

Changing State: Assignment

More generally:

>> *name* = *expression*

Variable name must start with a letter or underscore (_), and can then contain letters, numbers, and underscores:

```
a
radius
buffalo66
cost_of_living
```

Changing State: Assignment

- This notation differs from arithmetic, where it makes no sense to say, for example, $x = x + 1$
- In addition to state, variables help us to
  - write general formulas:
  - `>> c = sqrt(a^2 + b^2)`
  - avoid repeating computation of useful intermediate values.
- We should avoid repeating the same computation (code), because if there's a bugg in it, we have to fix it multiple times.
- We should avoid repeating the same computation (code), because if there's a bugg in it, we have to fix it multiple times.

Changing State: Assignment

E.g., gravitational force *F* of moon at location (*x*, *y*, *z*) w.r.t. center of earth:

$$Fx = \frac{-GmMx}{\left(x^2+y^2+z^2\right)^{\frac{3}{2}}}$$

$$Fy = \frac{-GmMy}{\left(x^2+y^2+z^2\right)^{\frac{3}{2}}}$$

$$Fz = \frac{-GmMz}{\left(x^2+y^2+z^2\right)^{\frac{3}{2}}}$$

all contain

$$\frac{-GmM}{\left(x^2+y^2+z^2\right)^{\frac{3}{2}}}$$

Changing State: Assignment

all contain $\frac{-GmM}{(x^2+y^2+z^2)^{\frac{3}{2}}}$

So...

```
>> pos_to_force = -G*m*M/((x^2 + y^2 + z^2)^(3/2));
>> Fx = x*pos_to_force;
>> Fy = y*pos_to_force;
>> Fz = z*pos_to_force;
```

Help!

All Matlab operators have built-in help:

```
>> help cos
 COS    Cosine.
    COS(X) is the cosine of the elements of X.

    See also ACOS, COSD.
```

Unbroken block of comments at top of your own code gets reported as help:

```
>> help simon_lab1
  Simon Levy, CSCI 121, Jan. 04,2005
  Lab 1: Polynomial Curve Fitting
```

Functions: The Basis of Computing

- Most of what we do in Matlab (or any language) involves writing and using functions.
- Like a function in math, a Matlab function has one or more inputs (arguments) and an output (return value).
- Each function is stored in its own document (.m file)
- We can call functions directly from the command prompt, and functions can call each other.

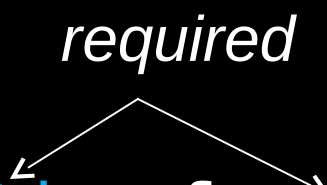
Example: Hypotenuse of Right Triangle

```
1   function c = pythag(a, b)
2   % PYTHAG implements Pythagorean theorem to compute hypoteneuse.
3   %
4   %    PYTHAG(A, B) returns the hypoteneuse of right triangle whose legs have
5   %    lengths A and B, using the Pythagorean Theorem.
6
7   c = sqrt(a^2 + b^2);
```

```
>> pythag(3, 4)

ans =

     5
```

```
>> help pythag
  PYTHAG implements Pythagorean theorem to compute hypoteneuse.

     PYTHAG(A, B) returns the hypoteneuse of right triangle whose legs have
     lengths A and B, using the Pythagorean Theorem.
```

Variable Names Are Arbitrary But Should Make Sense




```
function foo = bar(baz, moo)
foo = sqrt(baz^2 + moo^2)
```

A Function's Variables Are "Private" (Hidden)

```
>> h = pythag(5, 7)

h =

    8.6023

>> a
??? Undefined function or variable 'a'.

>> b
??? Undefined function or variable 'b'.

>> c
??? Undefined function or variable 'c'.
```

Programming to an API

- It is rare to build an entire software project from scratch.

- Instead, we reuse our own or other's code.
- So we have to know how to invoke (call) existing code.

- Application Programming Interface specifies how to do this

- Simplest form is what we get when call **`help`** on a function

Requirement Specification via Stub Functions

- Stub function is like an API for something not yet written:

```
1   function grid = init(n)
2   % INIT Cellular Automaton initialization
3   %
4   %  INIT(N) returns an NxN grid of values initalized for a particular CA.
5
6   % XXX currently returns an empty grid
7   grid = [];
```