Outline

- Announcements
  - HWI due today, 5PM
  - HWII available later today
  - Sign-up for CTC account
- MATLAB
- Bringing MATLAB into C

MATLAB

- "MATrix LABoratory"
- Started out as a front end for LINPACK, a linear algebra library that was the antecedent of LAPACK
- MATLAB is an interactive system for scientific computing
  - lots of built-in functions
  - programmable
  - excellent graphics
- But what *is* MATLAB?

MATLAB basics

- Interact with MATLAB through a command line
- Commands are evaluated one-by-one
- MATLAB stores variables in the "workspace"
  - Can get info on variables by typing "whos"

MATLAB variables

- MATLAB variables are 2D arrays of doubles (e.g. matrices)
  - Through V4, this was strictly true
  - V5 added ND-arrays, structs, and cell-arrays

MATLAB arrays

- Can create arrays with
  - brackets
    - A=[1 3 5;2 4 6];
  - commands like zeros, ones, rand
    - A=zeros(2,3);
  - colon operator
    - A=[1:2:5; 2:2:6];
  - or by loading a file
    - A=load('Cfinal.txt');

MATLAB operations

- Arithmetic and logical operations in MATLAB are vectorized:
  - A=B+C is equivalent to

```
for(j=0;j<nrows;j++){
    for(k=0;k<ncols;k++){
        A(j,k)=B(j,k)+C(j,k);
    }}
```

    - B and C must be same size or scalar
  - A=B*C is matrix multiplication
    - B: m-by-p, C: p-by-n, then A: m-by-n

MATLAB operations

- MATLAB's "\" solves linear systems
  - x=A\b; %solves Ax=b
  - "\" is very smart

MATLAB functions

- MATLAB provides lots of built-in functions
  - Math: sin, cos, sqrt
  - Lin. Alg: eigs, svd, lu
  - Solve ODE's
  - These are all vectorized
- There are many toolboxes which contain additional functions

MATLAB programming

- Can create new functions (m-files)
  - text files ending with .m & containing MATLAB commands
  - first line of file has form:
    - function [o1, o2,… oM]=name(i1,i2,… iN)

Outputs
-value of o1, o2,etc at end will be returned

Inputs
-call-by-value

MATLAB Programming

```
function [C]=SolveA(A,RHS);
    [m,n]=size(A);
    [p,q]=size(RHS);
    if(p!=m)
        error('RHS &A must have same number of rows');
    end
    C=A\RHS;
```

- MATLAB programs are simple and compact
- Excellent language for prototyping

Calling MATLAB

- C/C++ programs can make use of many of MATLAB's nice features:
  - vectorized operations
  - complex types
  - linear algebra and math functions

Calling MATLAB

- Bring these capabilities to C was not easy
  - need to manage memory in a MATLAB-like way
  - need to call functions in a MATLAB-like way

Using MATLAB C Library

- #include "matlab.h"--Matlab C-functions and types
- Declare variables used with MATLAB functions to be
  - mxArray *volatile name=NULL;
- Start MATLAB memory management
  - mlfEnterNewContext(nout,nin,outvars,invars);
    - nout,nin--number of MATLAB output/input variables
    - outvars--list of output variables
    - invars--list of input variables
    - for main, this is just (0,0)

Using MATLAB C Library

- Write your code
- Destroy mxArray variables:
  - mlfDestroyArray(name);
- Stop MATLAB memory management
  - mlfRestorePreviousContext(nout,nin,outvars,invars)

Using MATLAB C Library

- Create with mbuild:
  - mbuild matccode.c
  - Passes matccode.c to C-compiler, and links to appropriate libraries
- MATLAB Libraries
  - MATLAB C functions are included in several "Shared-Object" (.so) libraries
  - Will talk more about them on Wed.
  - Must add <matlab>/extern/lib/<arch> to LD_LIBRARY_PATH
    - setenv LD_LIBRARY_PATH /usr/local/matlab/extern/lib/sgi:/usr/local/matlab/extern/lib/sgi/bin/sgi:$LD_LIBRARY_PATH

Using MATLAB C Library

- A=<Matlab expression>
  - mlfAssign(&A,mlf expression)
  - Ex: A=B*C;
    - mlfAssign(&A,mlfMtimes(B,C))

Using MATLAB C Library

- Ex: [L,U]=lu(A);
- Matlab function can take 1-2 inputs and produce up to 3 outputs
- C-version provides this too
  - mlfAssign(&L,mlfLu(&U,NULL,A,NULL));

Mixing Simple C and MATLAB C

- Converting C-arrays to mxArrays:
  - mlfAssign(&MXarray, mlfDoubleMatrix(m,n,Carray,NULL))
  - Copies contents of Carray (or first m*n elements) into Mxarray
  - For 1D arrays, this is straight-forward, but:

Multidimensional Arrays

A=

1	2	3
4	5	6

C: Row-major

1
2
3
4
5
6

MATLAB: Column-major

1
4
2
5
3
6

Multidimensional Arrays

mlfAssign(&Marray,
mlfDoubleMatrix(2,3,Carray,NULL))

Carray

1
2
3
4
5
6

Marray

1	3	5
2	4	6

NOT:

1	2	3
4	5	6

Mixing Simple C and MATLAB

- Converting from mxArrays to C
  - Get pointer to data in mxArray
    - double ptr;
    - ptr=mlfGetPt(A);
  - Use C function "memcpy"
    - memcpy(C,ptr,m*sizeof(double));
      - m is the amount of data in A to copy
      - m=prod(size(A));

Working with .mat files

- MATLAB archives data in an efficient file type called .mat
- We can read and write to .mat files from C
  - mlfSave(mxCreateString("name.mat"),"w",
    - "o1",o1,"o2",o2,…,NULL);
  - mlfLoad(mxCreateString("name.mat"),
    - "i1",i1,"i2",i2,…,NULL);