Performance
Coursework
Coursework

- hw10 due 5/3
- hw11 due 5/3
- Grade check period: May 1–5. All posted Compass grades will be considered final after this. Double-check if the grade says “upcoming”. Exception: hw02 may be listed wrong.
Question

Are you present today?

A Yes.
B Yes.
C Yes.
D Yes.
E Yes.
MATLAB provides the `interp1` function:

```matlab
x = 0:0.5:1;
y = rand(3,1);
xstar = 0.33;
ystar = interp1( x,y,xstar );

figure();
plot( x,y,'ro',xstar,ystar,'rx' );
ylim( [ 0 1 ] );
```
MATLAB provides the \texttt{interp1} function:

\begin{verbatim}
x = 0:0.5:1;
y = rand( 3,1 );
xstar = 0.33;
ystar_l = interp1( x,y,xstar, 'linear' ); %default
ystar_s = interp1( x,y,xstar, 'spline' );

figure();
plot( x,y,'ro',xstar,ystar_l,'rx',xstar,ystar_s,'gx' );
ylim( [ 0 1 ] );
\end{verbatim}
Interpolation

- In 2D, use `interp2` to get a single point, or `griddata` to reconstruct a whole grid:

```matlab
x = 0:0.05:1;
y = 0:0.05:1;
z = rand( 21 );
xstar = 0.33;
ystar = 0.33;
zstar = interp2( x, y, z, xstar, ystar );

figure();
hold on;
mesh( x, y, z );
plot3( xstar, ystar, zstar, 'o' )
```
Interpolation

\[ f(\mathbf{r}) = x \cdot (1-x) \cdot \cos(4\pi x) \cdot \sin(2\pi \sqrt{y}); \]

% Define the basic grid coordinates.
grid_x = 0:0.01:1;
grid_y = grid_x( : );

% Define a random subset of the grid for which % we will generate data.
pts = rand( 2,800 );  % x,y pairs
vals = f( pts( 1,: ),pts( 2,: ) );

% Generate a grid.  'nearest','linear','cubic'
grid_z=griddata(pts(1,:),pts(2,:),vals,grid_x,grid_y,'nearest');
figure();
hold on;
mesh( grid_x,grid_y,grid_z );
plot( pts( 1,:),pts( 2,:),’k.’ )
Performance
MATLAB provides a timing interface in `tic` and `toc`. Use one to start the stopwatch, the other to finish it.

```matlab
tic;

toc
```
Application: measure time required to solve a linear system.

t = zeros( 1,100 );
for n = 1:100
    A = rand(n,n);
    b = rand(n,1);
    tic;
    x = A \ b;
    t( n ) = toc;
end
plot( t )
Application: compare time required to solve a linear system.

t = zeros( 1,100 );
n = 100;
A = rand( n,n );
b = rand( n,1 );

tic;
x = A \ b;
t_leftdiv = toc;

tic;
x = inv( A ) * b;
t_inv = toc;
Performance

- Application: measure average and minimum time

REPS = 1000;    minTime = Inf;    nsum = 10;
tic; % TIC, pair 1

for i=1:REPS
    tStart = tic; % TIC, pair 2
    total = 0;
    for j=1:nsum
        total = total + besselj(j,REPS);
    end

    tElapsed = toc(tStart); % TOC, pair 2
    minTime = min(tElapsed, minTime);
end

averageTime = toc/REPS; % TOC, pair 1
Parallel for
Your machine has many processors, so it can solve certain kinds of problems \textit{in parallel}.

The simplest way is to use \texttt{parfor}—parallel for.

\begin{verbatim}
parfor i = 1:12
  disp( i );
end
\end{verbatim}
Notice that the processes don’t always execute in the same order—this is known as race conditions. To keep this from being a problem, parallel algorithms must be constructed carefully.
Performance

- Application: a computationally intensive task
- Serial version (one process):

```matlab
 tic
 n = 200;
 A = 500;
 a = zeros( n );
 for i = 1:n
   a( i ) = max( abs( eig( rand( A ) ) ) );
 end
 toc
```
Performance

- Application: a computationally intensive task
- Parallel version (many processes):

```matlab
tic
n = 200;
A = 500;
a = zeros( n );
parfor i = 1:n
    a( i ) = max( abs( eig( rand( A ) ) ) );
end
toc```
The first time you run a parallel process in MATLAB, it is slower because the system must be started up in parallel. Later runs are faster.

Not all problems are suitable for parallel execution because of this setup time. This one is too small:

```matlab
tic
n = 2 .^ 10;
A = zeros( n );
for i = 1:n  % try parfor also
    A( i,:) = 1:n .* sin( i*2*pi/2.^10 );
end
toc```
Practice Question
Compose a MATLAB script which prints out each number from 1 to 100. For each number, the following set of rules should be applied to the output:

- If divisible by 3, print number followed by ”fizz”.
- If divisible by 5, print number followed by ”buzz”.
- If divisible by both 3 and 5, print number followed by both ”fizz” and ”buzz”.
- Otherwise, just print the number.
For instance, the expected output from 9 to 15 will be:

9fizz
10buzz
11
12fizz
13
14
15fizzbuzz
for i = (1:100)
    if mod(i,15) == 0
        str = sprintf( '%dfizzbuzz',i );
    elseif mod(i,3) == 0
        str = sprintf( '%dfizz',i );
    elseif mod(i,5) == 0
        str = sprintf( '%dbuzz',i );
    else
        str = sprintf( '%d',i );
    end
    disp( str );
end
What Else is There?
Python concepts:
- data types: *set*, objects (classes)
- SciPy (scientific functions, equation solvers)
- Pandas (data analysis)
- SymPy (symbolic algebra)
- functional programming: *map*, *filter*, *lambda*
Other things to know

MATLAB concepts:
- toolboxes: Image Processing, Statistics, PDEs
- differential equation solvers (ode45)
- parallel programming (parfor)
- cells (like Python lists)
- linear algebra, symbolic algebra
Other things to know

Miscellaneous concepts:
- LaTeX markup for equations
- Mathematica, Maple, MathCAD (symbolic math)
- Campus Cluster (supercomputing)
Curriculum

- CS 205 Data-Driven Discovery
- TAM 470 Computational Mechanics
- CS 357 Numerical Methods
- STAT 440 Statistical Data Management
- GEOG 489 Programming for GIS

- CS minor (cs.illinois.edu)
- CSE minor (cse.illinois.edu)
- Software Carpentry workshops
Course Feedback (ICES)
That’s all Folks!