

## Topics for Today

- Overview of next four projects
- Plotting Libraries
  - Matplotlib
  - VPython
- Digital Audio

### Reading

- Matplotlib Tutorial: <http://matplotlib.sourceforge.net/tutorial.html>
- VPython plotting <http://www.vpython.org/webdoc/visual/graph.html>

## Projects

- **Digital Audio**
  - Handed out February 1
  - Submitted in two parts: writing basic functions and creating sounds
  - Arrays and loops; complete functions from specifications; create your own sounds
- **Percolation**
  - Handed out February 15
  - submitted in two parts: writing functions and running experiments
  - Reusable functions, recursion, evaluating computational experiments

## Projects

- **Simulating Physical Systems**
  - Handed out February 29
  - Submitted in two parts: ideal gas simulation and Ising spin simulation
  - Monte Carlo simulations (i.e., use random sampling to compute results) and visualizations of system behavior
- **Analyzing Biological Processes**
  - Handed out March 21
  - Use a data set generated by an experiment to identify protein-protein interactions
  - <http://www.nature.com/nature/journal/v415/n6868/full/415141a.html>
- **Project 5 tbd**

1/28/2008

## Reminder!

### EXAM 1

- Wednesday, February 13, 7-8pm
- Location: MJIS 1001 (Biomedical Engineering Bld.)
- MJIS is across State Street in Discovery Park

1/28/2008

## Matplotlib (MPL)

- <http://matplotlib.sourceforge.net/>
- **from pylab import \***
- matplotlib is a powerful Python 2D plotting library generating plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc, with just a few lines of code
- Plots can be saved in many file formats
  - useful in other courses and projects )
- Plot is drawn using show()
  - this needs to be the last statement of your program

5

1/28/2008

## Clicker Question 1 (participation only)

What plotting software have you used before?

- A. None**
- B. MS Excel**
- C. MatLab / Mathematica / Maple**
- D. VPython**
- E. Other**

6

1/28/2008

## Matplotlib versus VPython

- In many ways, Matplotlib is a more complete plotting library than VPython
- VPython has 3D features and interactive features missing in MPL

## Matplotlib versus Matlab

- Plotting in MPL is very similar to Matlab
- MPL integrates with Python which can be an advantage
- Matlab is not a free software

## MPL use in course

- Simple plots
- Bar graphs
- Histograms
  
- Most control of visuals is optional
- Add basic text and labels
- Computed data will generally be in an array
- Use when no dynamic updates should be seen and plot is generated at end of computation

## Future value program (Zelle p 44)

```
# futval.py
# A program to compute the value of an investment
# carried 10 years into the future

def main():
    print "This program calculates the future value"
    print "of a 10 year investment."

    principal = input("Enter the initial principal: ")
    apr = input("Enter the annualized interest rate: ")

    for i in range(10):
        principal = principal * (1 + apr)

    print "The value in 10 years is:", principal

main()
```

## Clicker Question 2

```
>>> my_list = ['a', 'b', [4, 5], 'x']  
>>> print my_list[2:]
```

- A. `[[4, 5], 'x']`
- B. `[4, 5, 'x']`
- C. `['b', [4, 5], 'x']`
- D. `[4, 5]`

11

1/28/2008

## Clicker Question 3

```
>>> p  
[[1, 1, 3], [4, 6, 0], [2, 5, -2], [2, 2, 2]]  
>>> print len(p[0]), len(p)
```

- A. 1 4
- B. [3, 4]
- C. 3 4
- D. 3, 3

12

1/28/2008

## Graphing the yearly value

- Show as plot
  - X-axis: value
  - Y-axis years
- Show as bar graph
  - One year as one bar (assuming not too many years)

13

1/28/2008

```
from pylab import *  
from numpy import *
```

MPL\_interest\_plot.py

```
principal = 10000  
apr = 5  
years = 10  
values = zeros(years)  
apr = (100+apr)/100.
```

```
for i in range(years):  
    values[i] = principal  
    principal = apr*principa
```

```
# show a MPL plot; use default values  
plot(range(years),values)  
show()
```

14

1/28/2008

## More on graphing future values

- Bar graph
  - replace `plot(range(years),values)` with `bar(range(years),values)`
  - file `MPL_interest_bar1.py`
- More customization
  - file `MPL_interest_bar2.py`
  - Give graph a name
  - Label x and y coordinate, make a title

15

1/28/2008

## Another MPL plotting example

```
from pylab import *

def plot_three():
    t = arange (0.0, 20.1, 0.1)
    s = range(21)
    # plot three functions
    plot (s,s, t,2*t, "rs", t,t* t, 'go')

    xlabel("input size")
    ylabel("time")
    ti = title("Plotting three functions")
    ti.set_color("r")
    ti.set_fontsize(14)
    ti.set_fontweight("bold")

if __name__ == "__main__":
    plot_three()
    show()
```

`MPLplots.py`

16

1/28/2008

## Making subplots in MPL

MPLsubplots.py

```
def mysubplots():
    t = arange(0.0, 20.1, .1)
    s = range(21)

    # define first figure
    subplot(211) # final figure has 2 rows, 1 column
                # use 121 for two columns and one row
    plot(s,s, t,2*t,"r+", t,t*t, 'go')
    xlabel("input size")
    ylabel("time")
    title("Plotting three runtime functions")

    # define second figure
    subplot(212)
    plot(t, sin(2*pi*t))
```

17

1/28/2008

## Histograms

```
hist(x, bins=10, ...)
```

- computes the histogram of x
- bins is either an integer number or a sequence giving the bi
- returns (n, bins, patches)

Look up commands at

[http://matplotlib.sourceforge.net/pylab\\_commands.html](http://matplotlib.sourceforge.net/pylab_commands.html)

18

1/28/2008

## MPL Histogram example

MPLhistogram1.py

```
v=[1,1,1,2,3,4,4,5,7,7,7,7,10,12,12,12,12,13,13,15,22,22,25]
```

```
# the histogram of the data in v
n, bins, patches = hist(v, 10)
# n is an array with the frequencies
```

```
setp(patches, 'facecolor', 'g')
xlabel('v')
ylabel('Counts')
title('Data set v')
```

19

1/28/2008

## Plotting in VPython

VPplot1.py

```
from numpy import *
from visual.graph import * # import graphing features

funct1 = gcurve(color=color.cyan)

for x in arange(0., 8.1, 0.1): # x goes from 0 to 8
    funct1.plot(pos=(x, 5.*cos(2.*x)*exp(-0.2*x))) # plot
```

*pos=(x,y) adds points to the plot shown  
in the display*

20

1/28/2008

## Future value plotting with VPython

- **VPinterest-2d.py**: sort of boring
- **VPinterest-3d.py**: placing text is tedious
- **VPinteractive\_interest2d.py**: example of a slider interface
- **VPinteractive\_interest3d.py**: sort of overkill