Example Slide Show with Lasting Impact

Paul van Eikeren, Ph.D.

Presentation Outline

Word as a Slide Show Editor

Word provides headings with styles

Python is a dynamic object-oriented programming language used in data analysis

Word provides bullets with styles

- **Matplotlib**: a python library for 2-D technical plotting
- **Numpy and Scipy**: python libraries to support technical computing

Word provides equations editing

$$x_0^2 - 81(x_1 + 0.1)^2 + \sin(x_2) + b = 0$$

Inference provides dynamic execution

```
print "Hello world"  # in python
Hello world
```

Demo Python List Operations

Executes First

```
# Create a list
resize=[6.45,'SOFIA',3,8.2E6,15,14]
print(resize); print(len(resize))
[6.4500000000000002, 'SOFIA', 3,
8200000.0, 15, 14]
6

# Shrink a list
resize[1:4]=[55]
print(resize); print(len(resize))
[6.4500000000000002, 55, 15, 14]
4
resize[3]=['all','for','one']
print(resize); print(len(resize))
[6.4500000000000002, 55, 15, ['all',
'for', 'one']]
4
```

Executes Second

```
# add to list
temp=resize[:3]
resize=temp+resize[3]
print(resize)
print(len(resize))
[6.4500000000000002, 55, 15, 'all',
'for', 'one']
6

# delete an element
del resize[3]
print(resize)
print(len(resize))
[6.4500000000000002, 55, 15, 'for',
'one']
5
```

Solve Nonlinear Equations

Problem to solve:

$$3x_0 - \cos(x_1 x_2) + a = 0$$

$$x_0^2 - 81(x_1 + 0.1)^2 + \sin(x_2) + b = 0$$

$$e^{-x_0 x_1} + 20x_2 + c = 0$$

```
def nonlin(x,a,b,c):
   x0,x1,x2 = x
   return [3*x0-cos(x1*x2)+ a, x0*x0-81*(x1+0.1)**2 + sin(x2)+b,
   exp(-x0*x1)+20*x2+c]
```

Assign and call: assign values to a, b and c; assign starting search values; call solver

```
import scipy; from scipy import *; from numpy import pi
a,b,c = -0.5,1.06,(10*pi-3.0)/3
startValues = [0.1,0.1,-0.1]
root = scipy.optimize.fsolve(nonlin, startValues, args=(a,b,c))
residuals = nonlin(root,a,b,c)
```

Results:

Equation Roots (x0, x1, x2)	Equation Residuals (eq1,eq2,eq3)
5.000e-001 | 1.381e-013 | -5.236e-001	0.000e+000 | -2.231e-012 | 7.461e-014

Illustrate Simple Function

How to illustrate the function

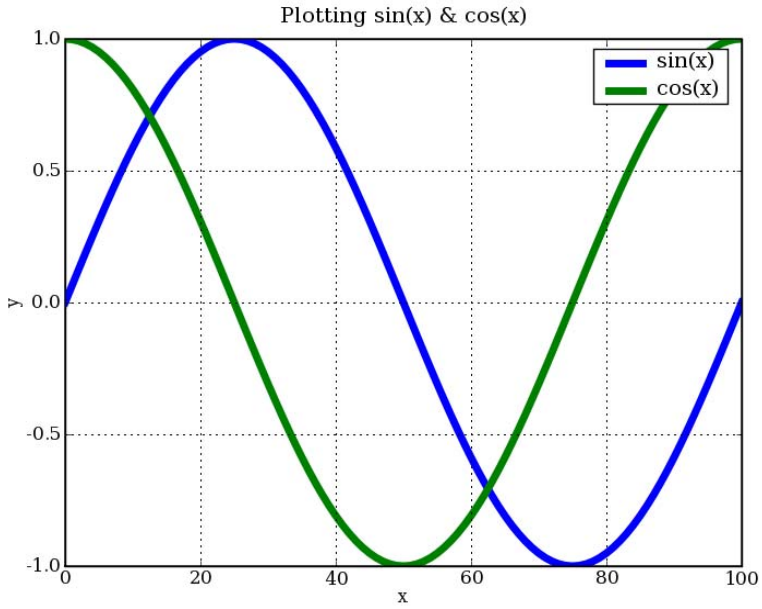
```
x = r_[0:101]
y01 = sin(2*pi*x/100)
y02 = cos(2*pi*x/100)

import pylab
from pylab import *

clf()
plot(x, y01, linewidth=5.0)
hold(True)
plot(x, y02, linewidth=5.0)
xlabel('x'); ylabel('y')
title('Plotting sin(x) & cos(x)')
legend(('sin(x)','cos(x)'))
grid(True)
```

What the function looks like

```
savefig(getInferenceImage())
```



Illustrate Complex Function

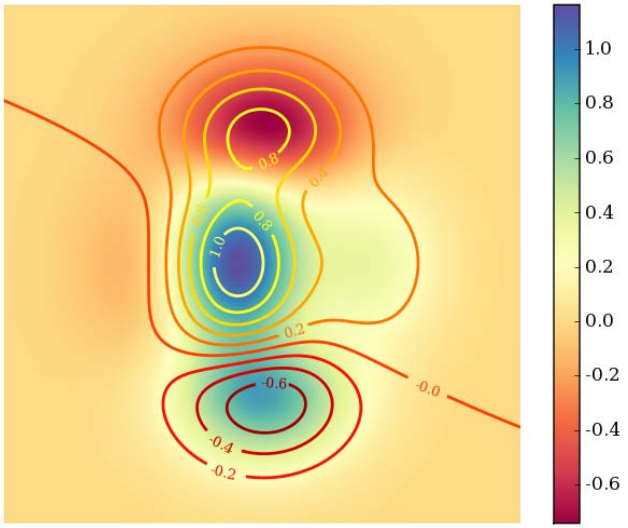
How to illustrate the function

```
import numpy as N; import pylab as pl
def z_func(x,y):
    return (1-x+x**3+y**5)*N.exp(-(x**2+y**2))

x = N.arange(-3.0,3.0,0.025)
y = N.arange(-3.0,3.0,0.025)
X,Y = pl.meshgrid(x, y)
Z = z_func(X, Y)
clf()
im=pl.imshow(Z,interpolation='bilinear',
    cmap=pl.cm.Spectral)
cset=pl.contour(Z,N.arange(-1.2,1.6,0.2),
    linewidths=2,cmap=pl.cm.hot)
pl.clabel(cset,inline=True,fmt='%1.1f',
    fontsize=10)
pl.colorbar(im)
pl.axis('off')
```

What the function looks like

savefig(getInferenceImage())



Using Code Objects

Presentation Issues / Solutions

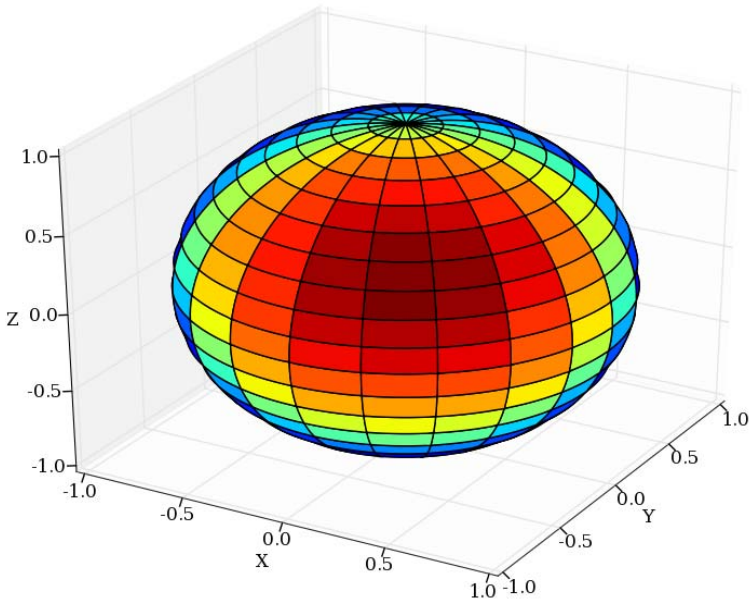
- ☐ Lack of space for analysis code
- ☐ Don't want to display analysis code
- ☐ Don't want to distract the audience with complex analysis code
- ☐ Want an accurate record of solution for posterity
- ☐ Want to be able to share solution method with others

Embed the code solution as a code object

Dynamically execute the code object

Illustration of Solution

`plotGlobe()`



Using Binary Data Objects

Presentation Issues / Solutions

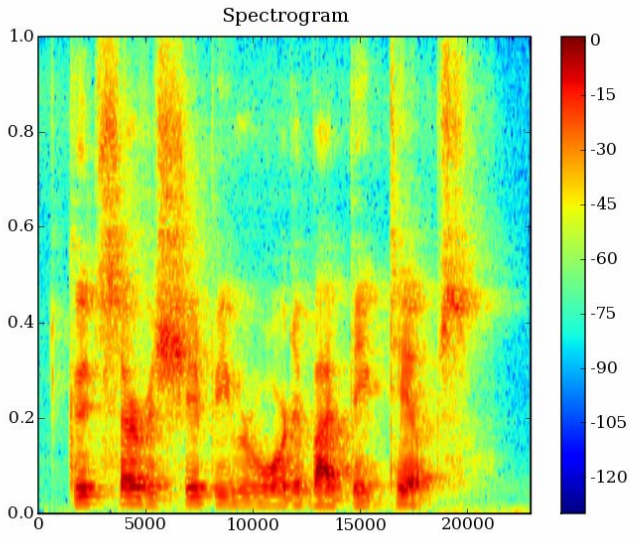
- Data subject to analysis can come from two sources: generated programatically or imported from storage
- Large data collections are not effectively stored as ascii text (eg., 46,000 element vector)
- Want to keep context, data and solution together

Embed the data as a binary data object

Dynamically access the data object

Illustration of Solution

```
from pylab import *
clf(); specgram(wavedata);colorbar ();
title('Spectrogram')
savefig(getInferenceImage())
```



Using Excel-based Data Set

Presentation Issues / Solns

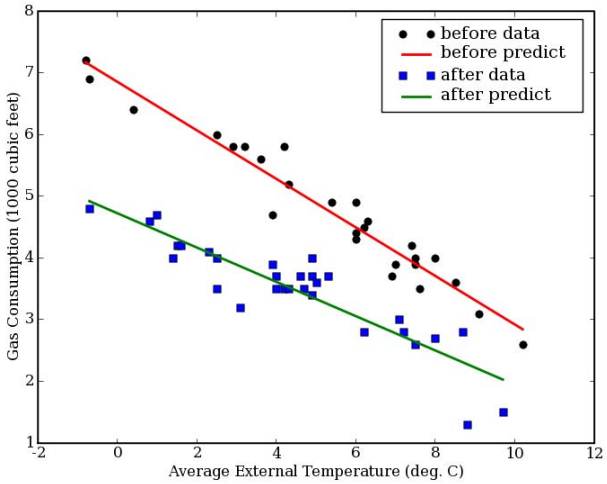
- ☐ Need to assemble, structure and inspect data prior to analysis
- ☐ Need a familiar and easy-to-use data preparation tool for collaboration with others
- ☐ Need to fully annotate data
- ☐ Need to access data and annotations programmatically
- ☐ Need to keep contex, data and solution together

*** Prepare data with Excel**

*** Embed the data as Excel doc**

Illustration of Solution

`runAnalysis()`



Case	slope	intercept	residual std error
before	-0.393	6.854	0.270
after	-0.278	4.724	0.343

More Information

For more information, please contact us at support@inference.us or visit us on the web at www.inference.us.