Python as a research toolkit for scientists and engineers

David Ascher
Brown University

Outline

- Part I: The Numeric Extension
- Part II: Python as a Research Tool
  - Requirements Analysis
  - Case Study: Visual Surface Perception
  - Let me count the ways...
  - Where to now?

Numeric Extension

- Jim Hugunin & Matrix-SIG
- Stable, efficient, flexible
- Two new types and a few functions (!)
  - Multidimensional Array
  - Universal Functions
- Easy to interface to existing C and Fortran libraries

Numeric Examples

```
>>> a = arange(9)
>>> a
0 1 2 3 4 5 6 7 8
>>> a.shape = (3,3)
>>> a
0 1 2
3 4 5
6 7 8
>>> a[0]
0 1 2
>>> a[:,1]
1 4 7
>>> a * 2
 0  2  4
 6  8 10
12 14 16
```

```
>>> a * (a + 1)
 0  2  6
12 20 30
42 56 72
>>> greater(a,3)
0 0 0
0 1 1
1 1 1
>>> choose(greater(a,3), (-a, a))
 0 -1 -2
-3  4  5
 6  7  8
>>> add.reduce(a)
9 12 15
>>> add.reduce(a.flat)
36
```

Extensions to Numeric

- LAPACK/LinearAlgebra
- FFTPACK/FFT
- RANLIB/RandomArray
- NetCDF
- Konrad Hinsen's modules
- OpenGL/GLOOP
- GIST
- PLPLOT
- etc.

Numeric Documentation

- Draft of the tutorial available on the web
- Tagged with TIM *(Janssen/Xerox)*
  - HTML, TeX, PS, Text
  - SGML someday?
  - How to do math?
- Standard needed for documentation of extensions

Python for Scientists

- Control freaks by necessity
- Fickle by nature
- Non-programmers:
  - Too many closed tools with assumptions
  - Not enough API's to GUI's
- Programmers:
  - Easy to learn yet flexible and "real"
  - Quicker development cycle than C++
  - Big Fast Math
  - Nicer than Matlab, S+, etc.

Visual Surface Perception

How do *orientation, depth, shading, transparency* etc. affect surface perception?

- 3D modeling software:
  - Too "rigid"
  - Not programmable
- OpenGL w/ C, C++:
  - Hard to be fickle
  - Just plain hard

OpenGL Module

- w/ Mike Harsthorn, Jim Hugunin, Tom Schwaller
- Open standard for 3-D rendering
  - SGI, IBM, Sun, HP, etc.
  - Hardware acceleration
  - Mesa is free "lookalike"
- Polygons, light sources, textures

OpenGL con't

- Features:
  - OpenGL 1.0, some 1.1 extensions
  - GLU, GLUT
  - Tk widget (Brian Paul's Togl)
  - Tight integration with Numeric:

    ```
    glVertex(really_big_array)
    ```
- Remaining Issues:
  - Cross-platform: Togl, fonts
  - Keeping up with OpenGL Extensions
  - GLU Incomplete
  - GLOOP: Object Oriented Framework

Examples

@ Brown: Behavioral Sciences

- Modeling
  - Neural Networks (in a course)
  - Computational Vision (in a PhD?!?)
- Data Analysis
  - Simple Statistics
  - FFT, Image Analysis (PIL)
  - FIDL + IMSL = “Real” Statistics
- Complicated Simple Munging
  - Slurp: Dedicated to tabular data formats

@ Brown: Computer Vision

Ben Kimia, Zach Roadhouse, Dale Bertrand, Perry Stoll

- Goals:
  - IDE for research
  - Separation of algorithms and display
- Raphael:
  - Modular, dynamic framework for image analysis and processing
  - ImageMagick & PIL backends
- PTUI:
  - Tkinter-based editor w/ builtin interpreter, object browser

Perspective

- Framework in place
  - Solid Core Functionality
  - Good First Extensions:
    - LAPACK/LinearAlgebra, FFTPACK/FFT
    - RandomArray, Konrad's modules, NetCDF
    - OpenGL
- Still Needed: Tools and more Extensions
  - Statistics (vs. S+, IMSL, SPSS, etc.)
  - Symbolic Math (vs. Mathematica et al.)
  - Better Graphing (vs. Matlab & Mathematica)
  - Interactive/Exploratory Data Analysis (vs. S+ and GUI's)

References

- Numeric:

http://www.sls.lcs.mit.edu/~jjh/numpy

- Numeric Tutorial:

http://maigret.cog.brown.edu/Python/Extensions/Numeric/

- OpenGL Extension:

ftp://ftp.python.org/pub/python/contrib/Graphics/PyOpenGL.tar.gz

- Other extensions to be advertised on the Numeric home pages.