2.18: Introduction to Programming for Geoscientists

Lecture 4 - Array computing and curve plotting

19 November 2015

Vectors and Arrays

- Vector: a one-dimensional data structure containing a sequence of elements. Often represented as a list.

$$\begin{bmatrix} 5 & 10 & -1 \end{bmatrix}$$

- Array: also contains a sequence of elements, but is a generalisation of vectors to n dimensions.

$$\begin{bmatrix} 0 & 12 & -1 \\ -1 & -1 & -1 \\ 11 & 5 & 5 \end{bmatrix}$$

- Fixed size, can only contain one data type.
- Generally faster than lists.

Arrays: linspace and zeros

- Two useful functions for creating arrays:
  - linspace(start, end, n): array of n uniformly distributed points in [start, end].
  - zeros(n): array of n elements all initialised to zero.
- Or...define as a list of lists and cast/convert to an array:

$$\begin{bmatrix} 0 & 12 & -1 \\ -1 & -1 & -1 \\ 11 & 5 & 5 \end{bmatrix}$$

```
a = [ [0, 12, -1],
[-1, -1, -1],
[11, 5, 5] ]
a = array(a)
```

- Remember `from numpy import *`

Arrays: Referencing/accessing elements

- Same as referencing list elements.
- `a[i][j]` accesses the element at row `i` and column `j`.
- Row fiRst, Column seCond.

Arrays: References vs Copies

- `a = x` will make a reference to, not a copy of, array `x`.
- Any changes to `a` will also occur in `x`.
- Use `a = x.copy()` to make a copy of array `x`.

Arrays: Vectorised functions

- A vectorised function accepts an array as its input...
- ...and for each element of that array, compute the result...
- ...and output all results in a new array.

```
from numpy import *
a = linspace(0, 1, 10)
result = sin(a) # Result is an array here.
```

- The loop over elements is implicit.