Computing at SIO

or

Learning to Talk Instead of Point

Duncan Agnew

MWF 3-4

but we need to reschedule MW

Doing Science (Instead of Taking Classes)

- Time scale is years, not weeks: your memory is not that good.

Doing Science (Instead of Taking Classes)

- Time scale is years, not weeks: your memory is not that good.

- You will, eventually, not know what you are doing – that's why we call it research.

Doing Science (Instead of Taking Classes)

- Time scale is years, not weeks: your memory is not that good.

- You will, eventually, not know what you are doing – that's why we call it research.

- And so, anything you do, you will probably do many times – and there are very few things you won't do more than once.

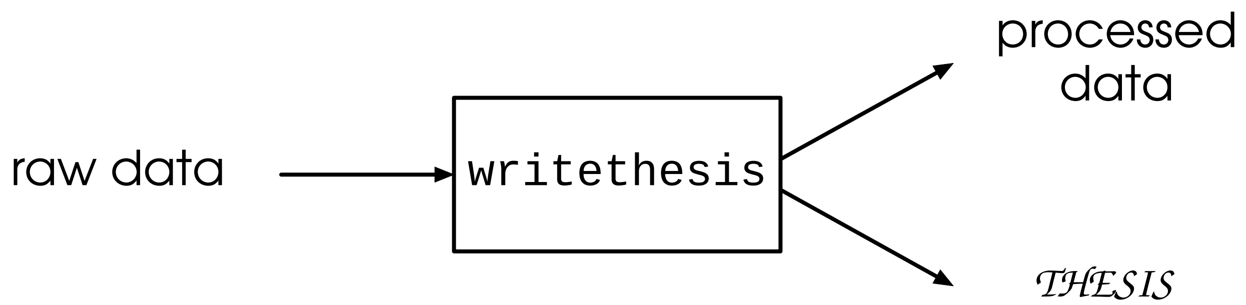
Doing Science (Instead of Taking Classes)

- Time scale is years, not weeks: your memory is not that good.
- You will, eventually, not know what you are doing – that's why we call it research.
- And so, anything you do, you will probably do many times – and there are very few things you won't do more than once.
- And part of science is being able to reproduce your results.

Doing Science (Instead of Taking Classes)

- Time scale is years, not weeks: your memory is not that good.
- You will, eventually, not know what you are doing – that's why we call it research.
- And so, anything you do, you will probably do many times – and there are very few things you won't do more than once.
- And part of science is being able to reproduce your results.
- For all these reasons you will benefit enormously from having all your computation procedures written down so that you can redo them.

A Research Ideal



What is the best computational approach to achieve this?

Options are **GUI** (Graphical User Interface) or a (actually many) **computer language**.

Advantages of GUI's (Pointing)

- Easy to learn (until they get complicated)

Advantages of GUI's (Pointing)

- Easy to learn (until they get complicated)

- Well-suited to nonverbal activities, for example, graphical editing, as in Photoshop.

Advantages of GUI's (Pointing)

- Easy to learn (until they get complicated)
- Well-suited to nonverbal activities, for example, graphical editing, as in Photoshop.

But

- Do not (usually) keep a record.

Advantages of GUI's (Pointing)

- Easy to learn (until they get complicated)
- Well-suited to nonverbal activities, for example, graphical editing, as in Photoshop.

But

- Do not (usually) keep a record.
- Become unworkable as complexity of task increases – how would you like to use one for searching in Google?

Using a Language ("Talking")

Advantages

- Language capability limited only by the skill of the developer.

Using a Language ("Talking")

Advantages

- Language capability limited only by the skill of the developer.
- Unless you use speech-recognition (don't), gives a permanent record.

Using a Language (“Talking”)

Advantages

- Language capability limited only by the skill of the developer.
- Unless you use speech-recognition (don’t), gives a permanent record.

Disadvantages

- You will need to learn an **editor** just to talk to the computer.

Using a Language ("Talking")

Advantages

- Language capability limited only by the skill of the developer.
- Unless you use speech-recognition (don't), gives a permanent record.

Disadvantages

- You will need to learn an **editor** just to talk to the computer.
- Language complexity limited only by the skill of the developer.

Using a Language (“Talking”)

Advantages

- Language capability limited only by the skill of the developer.
- Unless you use speech-recognition (don’t), gives a permanent record.

Disadvantages

- You will need to learn an **editor** just to talk to the computer.
- Language complexity limited only by the skill of the developer.

Like any new language, difficult to learn – though these are all “designed” languages, which can help.

Using a Language ("Talking")

Advantages

- Language capability limited only by the skill of the developer.
- Unless you use speech-recognition (don't), gives a permanent record.

Disadvantages

- You will need to learn an **editor** just to talk to the computer.
- Language complexity limited only by the skill of the developer.

Like any new language, difficult to learn – though these are all "designed" languages, which can help.

Using a Language ("Talking")

Advantages

- Language capability limited only by the skill of the developer.
- Unless you use speech-recognition (don't), gives a permanent record.

Disadvantages

- You will need to learn an **editor** just to talk to the computer.
- Language complexity limited only by the skill of the developer.

Like any new language, difficult to learn – though these are all "designed" languages, which can help.

Language I: Unix (aka shell)

```
cat << XXX | fgrep South | sort -n +8
1946 04 01 12 29 03  53.362 -162.854 8.6    165 South of Alaska
1950 08 15 14 09 36  28.290   96.657 8.6   3300 Eastern Xizang-India border
1952 11 04 16 58 27  52.756  160.056 9.0      - Off east coast of Kamchatka
1957 03 09 14 22 33  51.587 -175.419 8.6      - Andreanof Islands, Aleutians
1960 05 22 19 11 17 -38.294  -73.054 9.6   1260 Near coast of central Chile
1963 10 13 05 17 55  44.763  149.801 8.6      - Kuril Islands, Russia
1964 03 28 03 36 12  61.019 -147.626 9.2    236 Southern Alaska, United States
1965 02 04 05 01 21  51.210  178.498 8.7      - Rat Islands, Aleutian Islands
2004 12 26 00 58 52   3.270   95.860 9.3 227898 Off W coast of N Sumatera
2005 03 28 16 09 37   2.050   97.060 8.6   1324 Northern Sumatera, Indonesia
2007 09 12 11 10 26  -4.440  101.370 8.5     25 Southern Sumatera, Indonesia
XXX
```

We will start with this, as it is the most basic, and you can actually do quite a lot with existing Unix tools.

Language II: awk

```
echo X |awk '{pi=3.14159265
      for(i=0;i<=1000;i++) {
        t1=2*pi*i/1000
        t2=t1*n
        x=cos(t1)+a*cos(t2)
        y=sin(t1)+a*sin(t2)
        printf"%.4f %.4f0,x,y
      }
}' n=$1 a=$2 > tmp
```

This is a very simple programming language.

Language III: Matlab

```
function vector_xyz=xform_neu_to_xyz(xyz,vector_neu)
vector_neu=vector_neu(:);
vector_neu_cov=diag(vector_neu(4:6).^2);
vector_neu=vector_neu(1:3);
a=6378137;
f_inv=298.257223563;
f=1/f_inv;
e2=2*f-f^2;
p=sqrt(xyz(1)^2+xyz(2)^2);
r=sqrt(p^2+xyz(3)^2);
mu=atan(xyz(3)/p*((1-f)+e2*a/r));
long=atan2(xyz(2),xyz(1));
lat=atan2(xyz(3)*(1-f)+e2*a*sin(mu)^3,(1-f)*(p-e2*a*cos(mu)^3));
rot=[-sin(lat)*cos(long)  -sin(long)  cos(lat)*cos(long)
     -sin(lat)*sin(long)   cos(long)  cos(lat)*sin(long)
      cos(lat)             0          sin(lat)];
vector_xyz=rot*vector_neu;
vector_xyz_cov=rot*vector_neu_cov*rot';
vector_xyz_std=diag(vector_xyz_cov).^(.5);
vector_xyz=[vector_xyz;vector_xyz_std];
```

A tool of choice for mathematical programming – except for speed.

Language IV: Fortran

```
      function juldat(it)
      dimension it(*)
c
c  Julian Date from Gregorian date, Algorithm from p. 604, Explanatory
c    Supplement Amer Ephemeris & Nautical Almanac (cf Comm CACM, 11, 657 (1968)
c    and 15, 918 (1972)) Valid for all positive values of Julian Date
c
      juldat=(1461*(it(1)+4800+(it(2)-14)/12))/4
     1     + (367*(it(2)-2-12*((it(2)-14)/12)))/12
     2     - (3*((it(1)+4900+(it(2)-14)/12)/100))/4+it(3)-32075
      return
      end
```

You will need to know this for dealing with "legacy code" or writing your own.

Language V: Plotxy

```
frame grid solid
char .12
weight 12
titl A Sample Function
char .1
weight 10
xlab Time
ylab Function
file tmp
read
plot
stop
```

A very simple, but powerful, plot program, which makes much better graphics than Matlab does.

Language VI: GMT

```
gmtset GRID_CROSS_SIZE 0 ANNOT_FONT_SIZE_PRIMARY 10
gmtset PAGE_ORIENTATION portrait
pscoast -Rg -JN0/3i -Bg30 -Dc -Ggray -W -K > tmp1.ps
cat tmp2 | psxy -G255/255/255 -R -J -Sc.04i -O >> tmp1.ps
```

A very powerful, and complicated, plot program, which is excellent for geophysical and ocean0ographic data.

Language VII: Latex

```
The \textbf{magnitude} of the vector is its length,
for which the notation and definition are

\begin{equation}
\label{eq-mag}
| \bv | = \sqrt{ v_1 ^ 2 + v_2 ^ 2 + v_3 ^ 2}
\end{equation}
%
A \textbf{unit vector} is one whose magnitude is 1;
we usually designate a unit vector in the direction of $\bv$ by $\hat{\bv}$,
and designate unit vectors that are orthogonal
(at right angles) by the letter $\be$.
```

This is (many of us think) how you should write your papers. For scientific writing, MS Word is a poor substitute (though lots of people use it).