Thanks Paulo, I totally agree with your analysis. Having tried out
Python myself for a new project, I will add a few things:

1) awkwardness of Numpy

I totally agree with you and dislike the mix between numpy's ndarray's
and python lists. Specially the syntax overhead is not exactly aesthetic
(eg a = np.array([1,2,3]) instead of a simple a = [1, 2, 3]). However,
I've come to really appreciate some of Numpy's broadcasting syntax,
which simplifies some operations, for example:

```
>>> a = np.zeros((3, 4))
>>> a += [1, 2, 3, 4]
>>> a
array([[ 1.,  2.,  3.,  4.],
       [ 1.,  2.,  3.,  4.],
       [ 1.,  2.,  3.,  4.]])
```

To be compared to IDL:

```
IDL> a = FLTARR(4, 3)
IDL> a += [1, 2, 3, 4] # (FLTARR(3)+1)
IDL> a
       1.0000000      2.0000000      3.0000000      4.0000000
       1.0000000      2.0000000      3.0000000      4.0000000
       1.0000000      2.0000000      3.0000000      4.0000000
```

And there are some other advantages (e.g the ellipsis syntax:
a[1:3, ...] which helps you not to bother about variable dimensions any
more)

2) Debugger

I agree with you again. One thing that surprised me is how slow Python
debuggers are, and that they require a separate running mechanism. IDL
in turn has no formal "debug mode" (just put a breakpoint and wait) and
is running at the exact same speed. Does anyone know why the two
debuggers are so different?

3) Python third party libraries

The major reason for choosing python for my new project (aside of the
license issue) is simply that I spared a HUGE amount of work, simply

because the things we badly needed were already there. I wont give a full list here but basically, after a year using libraries such as Pandas, Shapely, or Scikit-Learn, going back to IDL for serious data crunching is quite frustrating...

I think that the recent efforts of the IDL devs to make the syntax more flexible are the right ones, and the idea to allow python - IDL bindings is really great. But IMO the next steps should be a revisiting of some unsatisfying basic functionalities:
- the handling of time: IDL's floating julian days are not adapted to today's standards, and plotting timeseries still is a pain.
- it would be great to have indexed and labeled arrays such as in Pandas and xray...
- it would be much nicer to be able to do:

IDL> a = list([1, 2, 3], DICTIONARY('a', 1, 'b', 2))
IDL> a[0][1]
        2
IDL> a[1].b
        2

instead of the current syntax:

IDL> (a[0])[1]
        2
IDL> (a[1])['b']
        2

Cheers,

Fabien

On 09/02/2015 10:22 PM, Paulo Penteado wrote:
> snip