
Subject: Re: speed comparison of IDL, numPy, Matlab
Posted by [Christopher Lee](#) on Mon, 12 Feb 2001 15:14:04 GMT
[View Forum Message](#) <> [Reply to Message](#)

>>>> > "Gavrie" == Gavrie <gavrie@my-deja.com> writes:

Gavrie> In article <3A7EEE5C.6419B707@pacific.jpl.nasa.gov>, Benyang
Gavrie> Tang <btang@pacific.jpl.nasa.gov> wrote:

Gavrie> So, this difference seems a bit strange, doesn't it? I'm using
Gavrie> MATLAB 6 (which is supposed to be *slower* than 5.3), and Python
Gavrie> 1.5.2.

I believe there's an explanation for this:

While I've heard that Matlab 6's footprint is much larger, especially the interface, I have heard that Matlab 6 does (finally) use LAPACK and BLAS which are optimized for modern machines w/ hierarchical memory. I would therefore expect that it would do substantially better than the out-of-the-box numpy build for matrix multiplication. By default, numpy uses a naive algorithm for matrix multiplication which is quite cache unfriendly. When numpy is linked to ATLAS's BLAS routines and LAPACK, it's more cache-friendly---and much faster.

I did some benchmarks myself: For matrix inversion of a 1000x1000 matrix, numpy-atlas is 7 times faster than matlab 5.3 (no lapack). The difference is greater if you have a dual processor machine because ATLAS now has options for multi-threaded operation. And for matrix multiplication you can compare the naive numpy multiplication with the optimized-dgemm based version on the same matrix:

```
# 1000 x 1000 matrix
time elapsed for matrixmultiply (sec) 24.142714
time elapsed for dgemm-matrixmultiply (sec) 5.997188
```

-chris

[BTW: I'm working on putting together an optimized-numpy distribution that uses ATLAS by default.]
