
Subject: speed comparison of IDL, numPy, Matlab
Posted by [Benyang Tang](#) on Mon, 05 Feb 2001 18:18:04 GMT
[View Forum Message](#) <> [Reply to Message](#)

Out of curiosity, I did a quick benchmark test of IDL, NumPy and Matlab on my desktop machine. I know benchmarking is a complicated issue; don't take my naive test too serious.

=====
Conditions of the test:
=====

* Machine: a dual Intel Xeon 550 MHz box with 1GB ram, running RedHat Linux 6.2. The machine was not doing any serious service, so the test code should have had close to 100% of the resources.

* IDL, version 5.3

* Python version 1.5.2; NumPy release 15.3-1

* Matlab version 5.3

* For each code, I ran it several times so the timing became somewhat stable. I just took the last reading; for the size of 600X600, timing fluctuation of less than 2% was observed.

* All tests were done in a period of 20 minutes; within this period the following data were collected. About 2 hours later, I redid the tests and obtained similar results.

=====
Timing (in seconds) of matrix multiplication of 2 m-by-m matrixes
=====

m	100	200	300	400	500	600

single precision						
IDL	0.02	0.12	0.54	1.83	4.06	6.90
NumPy	0.01	0.12	0.46	1.37	3.05	5.49
Matlab						(1)

double precision						
IDL	0.04	0.41	1.89	5.14	10.72	18.61
NumPy	0.02	0.16	0.97	2.50	5.05	8.75
Matlab	0.01	0.15	0.94	2.50	4.90	8.69

m	100	200	300	400	500	600

(1) Matlab does not do single-precision calculation.

=====

What the tests tell:

=====

- 1) NumPy is about as fast as Matlab;
- 2) NumPy is about 25% faster (in single precision), or more than 100% faster (in double precision) than IDL.

=====

Here are the codes (for the double precision test):

=====

IDL:

===

```
for m = 100,600,100 do begin
  a = double(randomn(0,m,m))
  b = double(randomn(0,m,m))

  time0 = systime(1)
  c = a##b
  dtime = systime(1)-time0

  print, string(m,m,dtime, format=('"multiplication of " ,i3, "X", i3, " matrixes takes ", f5.2)')
endfor
```

NumPy:

=====

```
from Numeric import *
from RandomArray import *
import time

for m in [100,200,300,400,500,600]:
  a= uniform(-1,1,(m,m,))
  b= uniform(-1,1,(m,m,))
  a=a.astype(Float64)
  b=b.astype(Float64)

  time0 = time.time()
  c = matrixmultiply(a,b)
  dtime = time.time() - time0

  print 'multiplication of %dX%d matrixes takes %5.2f' %(m,m,dtime)
```

Matlab:

=====

```
for m = [100 200 300 400 500 600];
```

```
a = randn(m);
b = randn(m);

time0 = clock;
c = a*b;
dtime = etime(clock,time0);

fprintf('multiplication of %dX%d matrixes takes %5.2f\n', m,m,dtime)
end

--
<> Benyang Tang
<> 300-323, JPL
<> 4800 Oak Grove Drive
<> Pasadena, CA 91109, USA
```
