Subject: Re: speed comparison of IDL, numPy, Matlab
Posted by Paul van Delst on Mon, 05 Feb 2001 20:08:54 GMT
View Forum Message <> Reply to Message

Benyang Tang wrote:
>
> 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.

O.k. :o)

I've never used Python, and Matlab only once or twice, but what's with the Python syntax? Seems a tad wordy. Looks like Nick Bower's IDL-like package in Python (http://nickbower.com/computer/pydl) is worth another look though......

paulv

P.S. BTW, in Matlab, how do you simply multiply the corresponding matrix elements? (i.e. not a matrix multiply).

--
Paul van Delst A little learning is a dangerous thing;
CIMSS @ NOAA/NCEP Drink deep, or taste not the Pierian spring;
Ph: (301) 763-8000 x7274 There shallow draughts intoxicate the brain,
Fax: (301) 763-8545 And drinking largely sobers us again.
Email: pvandelst@ncep.noaa.gov Alexander Pope.

Subject: Re: speed comparison of IDL, numPy, Matlab
Posted by Omur Bas on Mon, 05 Feb 2001 20:43:29 GMT
View Forum Message <> Reply to Message

Try

>> A.*B

for element-by-element multiplication.

Omur
>
> P.S. BTW, in Matlab, how do you simply multiply the corresponding matrix
elements? (i.e.
> not a matrix multiply).
>
>
> --

> Paul van Delst A little learning is a dangerous thing;
> CIMSS @ NOAA/NCEP Drink deep, or taste not the Pierian
spring;
> Ph: (301) 763-8000 x7274 There shallow draughts intoxicate the
brain,
> Fax: (301) 763-8545 And drinking largely sobers us again.
> Email: pvandelst@ncep.noaa.gov Alexander Pope.

Subject: Re: speed comparison of IDL, numPy, Matlab
Posted by David Lees on Tue, 06 Feb 2001 05:44:14 GMT
View Forum Message <> Reply to Message

Paul,

Thanks for the URL to pydl. It looks quite interesting.

I am a Python newbie, but your comments Python syntax and being "wordy"
seem a bit misplaced. The 'for' statement could have been written with
a 'range' so that it looks similar to the IDL for statement. The
wordiness of Python relative to IDL or Matlab is just due to the built
in functions used, not something inherent in Python. A function could
easily be defined for generating the uniform square matrices used in the
benchmark. I really like the balance that Python achieves between
clarity and program length. Certainly there have been languages like
APL that were far more compact than IDL or Matlab, but they were a bit
cryptic for my taste.

david lees

Paul van Delst wrote:
>
> Benyang Tang wrote:
>>
>> 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.
>
> O.k. :o)
>
> I've never used Python, and Matlab only once or twice, but what's with the Python syntax?
> Seems a tad wordy. Looks like Nick Bower's IDL-like package in Python
> (http://nickbower.com/computer/pydl) is worth another look though......
>
> paulv
>
<snip>

Subject: Re: speed comparison of IDL, numPy, Matlab
Posted by Nathaniel Gray on Tue, 06 Feb 2001 07:58:50 GMT
View Forum Message <> Reply to Message

Paul van Delst wrote:

> I've never used Python, and Matlab only once or twice, but what's with the
> Python syntax? Seems a tad wordy. Looks like Nick Bower's IDL-like package
> in Python (http://nickbower.com/computer/pydl) is worth another look
> though......

It's true that Numpy code can end up more verbose than Matlab, although this example doesn't really hilight that issue. This is to be expected from a language that wasn't designed from the ground up as a numerical computing language. There's a faint glimmer of hope that we may get a new Python operator for matrix multiplication, though. (See PEP 211)

On the flipside, have you ever tried to write user interfaces in Matlab? Have you ever tried to implement a dictionary in Matlab? Have you ever tried to understand when Matlab is making a copy of a giant array and when it's using a reference? Have you ever had your Matlab license expire the day of a meeting, _before_ you've made your plots?

I have. <shudder>

Having used IDL and Matlab quite extensively, I've realized that in any moderately complex numerical program 75% of the code is *programming* and only 25% of the code is *numerical*. I'd take a great general-purpose language with decent numerical extensions over a decent language with great numerical capabilities any day of the week, and twice on Sunday. :^)

> paulv
>
> P.S. BTW, in Matlab, how do you simply multiply the corresponding matrix
> elements? (i.e. not a matrix multiply).

mat1 .* mat2;

-n8

--
_.~'^`~._.~'^`~._.~'^`~._.~'^`~._.~'^`~._
Nathaniel Gray
California Institute of Technology
Computation and Neural Systems
n8gray <at> caltech <dot> edu
_.~'^`~._.~'^`~._.~'^`~._.~'^`~._.~'^`~._

Subject: Re: speed comparison of IDL, numPy, Matlab
Posted by Nathaniel Gray on Tue, 06 Feb 2001 07:58:50 GMT
View Forum Message <> Reply to Message

Paul van Delst wrote:

> I've never used Python, and Matlab only once or twice, but what's with the
> Python syntax? Seems a tad wordy. Looks like Nick Bower's IDL-like package
> in Python (http://nickbower.com/computer/pydl) is worth another look
> though......

It's true that Numpy code can end up more verbose than Matlab, although this example doesn't really hilight that issue. This is to be expected from a language that wasn't designed from the ground up as a numerical computing language. There's a faint glimmer of hope that we may get a new Python operator for matrix multiplication, though. (See PEP 211)

On the flipside, have you ever tried to write user interfaces in Matlab? Have you ever tried to implement a dictionary in Matlab? Have you ever tried to understand when Matlab is making a copy of a giant array and when it's using a reference? Have you ever had your Matlab license expire the day of a meeting, _before_ you've made your plots?

I have. <shudder>

Having used IDL and Matlab quite extensively, I've realized that in any moderately complex numerical program 75% of the code is *programming* and only 25% of the code is *numerical*. I'd take a great general-purpose language with decent numerical extensions over a decent language with great numerical capabilities any day of the week, and twice on Sunday. :^)

> paulv
>
> P.S. BTW, in Matlab, how do you simply multiply the corresponding matrix
> elements? (i.e. not a matrix multiply).

mat1 .* mat2;

-n8

```
--
_.~'^`~._.~'^`~._.~'^`~._.~'^`~._.~'^`~._
            Nathaniel Gray
   California Institute of Technology
     Computation and Neural Systems
     n8gray <at> caltech <dot> edu
_.~'^`~._.~'^`~._.~'^`~._.~'^`~._.~'^`~._
```