

Re: Matrix Multiplication

Source: <http://sci.tech-archive.net/Archive/sci.math.num-analysis/2008-01/msg00057.html>

- *From:* Evgenii Rudnyi <usenet@xxxxxxxxx>
 - *Date:* Sun, 6 Jan 2008 13:25:15 -0800 (PST)
-

Gordon,

Once more, thank you very much for your comments. They helped me to look at my text from a different viewpoint and see how it should be modified.

I cannot still accept your suggestion to use Fortran 9*, as my plan is different. I am a practitioner and I plan to write a text from a practical viewpoint. The emphasis will be on the use of existing numerical libraries with stress on sparse matrices. So the use of different programming languages is by design. However, I will limit myself by Fortran 77, C and C++, as this is what I know. Unfortunately it is impossible to learn everything. Well, such a choice allows me to cover most numerical libraries.

An important topic that actually I plan to cover is integration. It will include interoperability between different programming languages (Fortran 77, C and C++), linking issues and then using a scripting language as an integration platform. The latter is quite appealing from many different considerations.

By the way, one library that I plan to cover is MUMPS and it is written in Fortran 90.

I wish your success. Long live Fortran.

Evgenii

On 5 Jan., 21:29, Gordon Sande <g.sa...@xxxxxxxxxxxxxxxxxxxxx> wrote:

Dear Gordon,

It is your article to write. You have picked a hard topic and asked for comments. You got them.

Re: Matrix Multiplication

I do appreciate your comments and your time. Thank you. Unfortunately, I still did not completely get your points. Hence if you allow I will ask some more questions.

They pointed out consequences of your choice to switch to C++ in 1993 and to extol its virtues. Your choice of systems to compare is certainly idiosyncratic. Python, C++ vrs C and F77 seems to be more a setup for language wars than about Matrix Programming. Insisting that comments come with "your code" is following the best newsgroup language war traditions.

Here I would disagree. The use of several programming languages in one project is a common practice nowadays, at least what I see. For example, I am using LAPACK that is written in Fortran 77 and MUMPS that is written in Fortran 90. ATLAS, TAUCS and UMPFACK are written in C. My code is in C++ and now I am going to use Python as a scripting language.

Anyway, do I have a freedom to use a programming language of my choice or not?

Your choice is for you. When you start to write expositions then you need to have well based reasons for your recommendations. "I only know F77" would not seem to make it as a reason for ignoring F90 in an exposition. It may be OK for you next programming assignment but it will look curious on your resume if you are making the point of well based recommendations on modern practice.

If you want to write about matrix programming then where is the discussion of MatLab and its clones?

Re: Matrix Multiplication

In my text

<http://matrixprogramming.com/MatrixMultiply/>

there were links to both Matlab and Mathematica.

Fortran 90 lives well in that crowd with its array operations. Matrix multiply is one of the supplied intrinsics. To ignore that reality of current practice is to either be misinformed or out of date.

The matrix multiply is also intrinsic in Matlab, Mathematica and NumPy and I guess in many other systems. The example with matrix multiply as intrinsic is at the beginning of my text in NumPy – See mm.py.

In my opinion, everything here depends on whether this intrinsic will call optimized BLAS or not. If yes, it will be efficient in any language. If not, then it will be inefficient in any language. Do you agree with this statement?

Once one has gotten the concepts sorted out with vector and matrix capabilities then one can go back to the element based operations. To dismiss F90 with the comment that you only know F77 would appear to be an admission that you do not want to follow the contemporary style of use of the array notations.

I see your point here. Thank you. Yes, I should have mentioned the contemporary style of the array notation that is available in many modern languages. Will do.

Re: Matrix Multiplication

Finally one more question about Fortran 77 and Fortran9*. LAPACK is written in Fortran 77 and many say that it is the most efficient library for dense linear algebra (of course, when one uses it with the optimised BLAS). Well, on Netlib there is Fortran 95 interface to LAPACK but this is only an interface, the computational engine is still written in Fortran 77. Could you please make your comment in this respect?

You asked for comments on your Matrix Multiply. You "forgot" to mention that F90 has that as an intrinsic which will be as efficient or not as any other system that has a run time support for such an operation. You claimed that it did not much matter as all you knew was F77. For someone who seems to be proud of using modern features that is most curious omission.

LAPACK is MUCH more than just matrix multiply. Some of those things are not particularly friendly to array notation so would not use the array notation features of F90. The software engineering enhancements of F90 are provided by the application interface offered by the F90 wrappers. One would expect to see the array operations in use in setting up the data which is then operated on by LAPACK for the more elaborate operations. The F90 type safety (which unfortunately requires that you actually use the capabilities) would help avoid programming errors. LAPACK was developed over considerable time so one would not expect to see it redone instantly. There is no native C version of it either.

In the strengths of Fortran are both its long history and its upwards compatibility. That should not be mistaken for a license to believe that it is still your grandfathers F66. The history of the long delay in producing F77 compilers, the long delay in getting to F90 and the popularity of free Unix versions used to teach system programming has lead many computer scientists to think that it is smart computer science to believe in F66. But those are the same folks who invert matrices when they solve equations etc. If you are going to discuss matrix computations then one has every right to expect better and to say so.

One of the advantages of MatLab and such is that they provide a relatively simple conceptual model. This is important for those who only program part time in otherwise technical jobs. Fortran has the same simple conceptual base. The arcane technical issues that arise due to its ability to support several differing implementation strategies make that a bit of a stretch if pushed hard. For your apparent introductory audience a simple conceptual model is a great bonus.

Re: Matrix Multiplication

Thank you again. If you find time to answer questions above, I would appreciate it.

Best wishes,

Evgenii