Subject: Re: What does an optimal scientific programming language/environment need?
Posted by Jason Nielsen on Fri, 19 Sep 2003 19:35:40 GMT
View Forum Message <> Reply to Message

On Fri, 19 Sep 2003, grunes wrote:

> 3. Very rapid developement and testing. Requires extreme conciseness,
> support of arrays, complex number, linear algebra, finite element and
> numerical integration of functions and differential equations, and
> little need for type and shape declarations. Can easily switch on
> automatic detection of subscript checking, memory reference checking,
> argument mismatches, fixed and floating point errors. FORTRAN style
> adjustable array bounds (e.g., a(-3:4, 5:7)).

Very efficient built-in univariate and multivariate random number generation would be a nice touch ;-)!

> 8. A compiled mode that really is as fast as FORTRAN or C, if you add
> those declarations. Compiler would produce 2nd level code for
> compilation by g77 and gcc.

If you could manage this you would definitely get peoples attention. Most modern array/matrix interpreted languages: Matlab, S-plus, R, Octave, Euler, IDL, Yorick, Ox, GAUSS etc., etc., etc. have most of your other points covered. However, all of them suffer from the fact that they are too slow for intensive simulation. I personally use a couple of these regularly and when the going gets tough re-code sections that are slowing things up in Fortran95 for dyn.loading. However writing some code in your favorite matrix language, adding some type declarations and compiling the sucker to a fast binary would be a nice touch. The only in-development project trying for something along these lines is LUSH:

http://lush.sourceforge.net/

Unfortunately they are using the dreaded lisp infix syntax.... urrrggh I can't stand that ;-)! I suppose I'll just stick with my Python, R, Matlab, and Fortran95 mix until you are finished your project ;-).

Cheers,
Jason