

---

Subject: Idea for an IDL numerical toolbox

Posted by [Liam Gumley](#) on Mon, 12 Apr 1999 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

In light of some recent discussions about calling FORTRAN routines from within IDL, I started thinking about how one might go about putting together an IDL toolbox based on a well-known freely available FORTRAN library such as LAPACK (<http://www.netlib.org/lapack>).

One reason to do this would be to extend the numeric capabilities of IDL by making use of a very well tested FORTRAN library that is available in optimized form for many architectures, e.g.

SGI: <http://www.sgi.com/software/scsl.html>

Sun:

<http://www.sun.com/workshop/performance/wp-perflib/capabilities.html>  
and is also freely available in source form (see LAPACK link above).

Another reason would be to help combat a view I've heard from some Matlab users, which is that IDL is not well suited to intensive numerical computations.

Some items of information that I think could be relevant:

(1) It seems like there would be a fair amount of work involved in writing the C wrappers and IDL interface routines for a large FORTRAN library. Could this be automated somehow? NAG markets something called the NAGWare Gateway Generator, which is described at <http://www.nag.co.uk/nagware/NN.html>. Here's a quote:

"The NAGWare Gateway Generator (NGG) produces an interface that seamlessly integrates Fortran subprograms with MATLAB. It analyses the source code of a Fortran routine and automatically generates a MATLAB gateway. This allows the user to access the Fortran subprogram as if it were any other MATLAB command. Both forward and reverse gateways can be generated. Reverse gateways allow the Fortran program to access MATLAB commands and scripts."

There is a very nice freely available FORTRAN analyzer named ftnchek, available at <http://www.dsm.fordham.edu/~ftnchek> which will automatically extract information about the arguments of a FORTRAN subroutine. Using this tool it shouldn't be too difficult to build a script that automatically creates a C wrapper and IDL interface routine for a FORTRAN subroutine.

(2) There are a number of well documented and tested freely available FORTRAN numerical libraries built around LAPACK, one of which is "The Control and Systems Library SLICOT", available at <http://www.win.tue.nl/niconet/NIC2/slicot.html>

(3) Some work would be needed to make the compiling options work seamlessly on all IDL platforms (Unix at least). Right now it seems like the first time someone tries to link IDL with FORTRAN, there's a steep learning curve. I'm thinking of a package that can be downloaded, installed once, and then used directly from IDL without the user having to worry about how it was compiled or linked. The package would be smart enough to check (at installation time?) for the presence of optimized LAPACK on the host architecture, and if it was not present, LAPACK would be compiled from source.

Sounds like a nice cooperative project for the readers of comp.lang.idl-pvwave.

Any comments? Volunteers?

Cheers,  
Liam.

---

Liam E. Gumley  
Space Science and Engineering Center, UW-Madison  
<http://cimss.ssec.wisc.edu/~gumley>

---