

Re: Who uses clapack?

Source: <http://sci.tech-archive.net/Archive/sci.math.num-analysis/2004-12/0312.html>

From: Michael Hosea (*mhosea_at_mathworks.com*)

Date: 12/10/04

Date: Fri, 10 Dec 2004 15:21:05 -0500

I can't say whether my experience is rare or not, but there has not always been a FORTRAN compiler available for the target platforms that I was working on, namely the embedded ones. The C versions based on F2C weren't very inviting, either. I can run F2C myself, and I have. I've even hand-translated LAPACK code to efficient C. Even when a FORTRAN compiler is available, it can be an expensive option when its only purpose is to compile a FORTRAN library or two on a very occasional basis, and we might be talking about more than one FORTRAN compiler if the C source were cross-compiled on multiple platforms. In some cases it makes perfect economic sense to build and use a FORTRAN library, and in others it doesn't seem as appealing. Unfortunately for me, I've usually found myself in positions where it seemed quite unappealing for one reason or another. On one system there was even a custom C compiler that used a different floating point model. To my knowledge, no compatible FORTRAN compiler ever even existed for that platform!

What I have often wished for was a C implementation that represented the same quality and attention to detail as the FORTRAN version.

I have no idea what you could do about any of that as long as FORTRAN is the highest priority language to support, the "mother language", if you will. Obviously FORTRAN is still heavily used in some niches. I'm not questioning this, nor passing any kind of value judgement on it, but I would definitely look very hard at the question of what the priority *should* be, basing this on what you are trying to accomplish with a new version of LAPACK. You might get one answer if, for example, you were primarily interested in supercomputing or are only planning a relatively minor incremental update, and you might reach a different conclusion if you wanted use of the library to be easy and convenient to incorporate for the widest possible audience.

--

Mike

"Victor Eijkhout" <see.sig@for.addy> wrote in message
news:lgokq2z.1qtze21fbcwaoN%see.sig@for.addy...

> The authors of lapack/scalapack are starting to work on a new release of
> these packages. One of the things we want to address is the silliness
> that C users, instead of linking to the binaries of the Fortran version,
> use an automatically translated C version.

>

> Therefore we'd like to know precisely what the reasons for this are.

sci.math.num-analysis: Re: Who uses clapack?

>
> Naively one would say, compile the Fortran version, append an underscore
> (when appropriate) to routine names, and just link those libraries to
> your object files.
>
> What are the gotchas, and what are the real stumbling blocks here?
> Please spell it out in as much detail as you can muster. This situation
> really needs to improve.
>
> V.
> --
> email: lastname at cs utk edu
> homepage: www cs utk edu tilde lastname