Subject: Re: Cholesky factorisation
Posted by davidf on Mon, 21 Feb 2000 08:00:00 GMT
View Forum Message <> Reply to Message

Peter Scarth (p.scarth@mailbox.uq.edu.au) writes:

> I'm trying to determine if a symmetric matrix is positive-definite. If this
> sounds like gobbdlygook you might like to stop reading here. If the
> idl2matlab translate-o-matic existed, it would probably need to know about
> this to translate the handy \ (mldivide) operator.

This is an automatic response from David Fanning's news reader:

Sorry, but all articles with the words "idl2matlab translate-o-matic" have been targeted for immediate destruction. Thank you for your interest in IDL.

--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Cholesky factorisation
Posted by Peter Scarth on Tue, 22 Feb 2000 08:00:00 GMT
View Forum Message <> Reply to Message

Howdy all,
I'm trying to determine if a symmetric matrix is positive-definite. If this sounds like gobbdlygook you might like to stop reading here. If the idl2matlab translate-o-matic existed, it would probably need to know about this to translate the handy \ (mldivide) operator. There a number of tests for positive-definiteness and the one I'm looking at is to attempt cholesky factorisation of the matrix. IDL's CHOLDC procedure halts and returns CHOLDC: choldc failed (note that matlab can return a value to flag a failure in the decomposition).
The metho that I an using to get the decomposition looks something like this:

```
----------
pro choldc2,a,p
sa=size(a)
nd=sa[0] & m=sa[1] & n=sa[2]

; Some initial tests.....
; .....
```

```
p=fltarr(n)
for i=0,n-1 do begin
 for j=i,n-1 do begin
 sum=a(j,i)
 if i gt 0 then sum=sum-total(a(0:i-1,i)*a(0:i-1,j))
 if (i eq j) then begin
  if (sum le 0) then begin
  p=-1
  return
  endif else p(i)=sqrt(sum)
 endif else a(i,j)=sum/p(i)
 endfor
endfor
return
end
```
---

it works ok, returns -1 in p if the method fails but only runs at 1/10 the speed of the built in version.
Is it possible to vectorise this further, or has someone out there in cyberland found a more elegant solution to this problem already?

Thanks,

Peter Scarth
Biophysical Remote Sensing Group
The University of Queensland.

Subject: Re: Cholesky factorisation
Posted by Craig Markwardt on Tue, 22 Feb 2000 08:00:00 GMT
View Forum Message <> Reply to Message

Pavel Romashkin <pavel@netsrv1.cmdl.noaa.gov> writes:
>
> I can't help hearing some sarcasm regarding IDL in David McClain's
> postings. Why is that? After all, accepting computer's numerical
> precision is a matter of faith, too, especially in life-critical
> applications Raytheon is developing. I find using upper-level
> language like IDL very convenient, and availability of "lower-level"
> tools like pointers is nice. C is great, but so is IDL I think. If
> we don't trust functions written by others, we might as well wright
> everything ourselves in C...
>

I think McClain's point is that the implementation of built-in

functions in IDL is strictly black-box. But in IDL's case, it's more like black magic... :-)

Seriously though, many of the routines *are* implemented in IDL itself, and thus inspectable. That's good.

For the built-in routines, however, one has to take it on faith that RSI did a robust implementation, and Numerical Recipes is not always the right choice. Floating point precision on a computer is well-defined these days by the IEEE, and thus I have more faith in it.

Craig

--

---------- ----------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
---------- ----------