

Re: help about ARPACK solver

Source: <http://sci.tech-archive.net/Archive/sci.math.num-analysis/2007-05/msg00148.html>

- *From:* andy2O <andy2O@xxxxxxxxxxxxx>
 - *Date:* 19 May 2007 15:50:33 -0700
-

On 19 May, 09:55, lorin <wwang...@xxxxxxxxxx> wrote:

On May 17, 3:44 am, andy2O <and...@xxxxxxxxxxxxx> wrote:

On 16 May, 10:54, lorin <wwang...@xxxxxxxxxx> wrote:

Hi,

I am now involved in a computational electromagnetics project and have to obtain a few eigenvalues and corresponding eigenvectors of a large non-symmetric complex matrix (2K-by-2K).

Hello,

I used the "parallel" vector to verify if the result satisfies $A * x = \lambda * x$. For the eigenvector from Matlab, $(A * x - \lambda * x)$ could be as precise as (E-14), however, when I used the eigenvector from my fortran program (yes, there is a constant factor between the two eigenvectors), $(A * x - \lambda * x)$ can only be (E-04), sometimes even (E-03), I think such precision can not be satisfied in our application.

Thanks all the same,

With my best wishes,

lorin

Re: help about ARPACK solver

Hi lorin,

I'm glad to hear my explanation was useful and I'm sorry to hear your Fortran code isn't working right yet. Two simple thoughts come to mind (I apologise if they are too basic and you have done this already! No offence meant!):

1) When you compute the value of $(A*x - \lambda*x)$ for your Fortran code, are you doing this within the Fortran program? I ask because if you try and save the values to a file and load them into Matlab to do the check you will lose precision.... So, make sure you compute $(A*x - \lambda*x)$ using the Fortran code to make an accurate comparison (You probably already do this, but I thought I should check!).

2) Matlab's `eigs()` function uses ARPACK, so logically your Fortran code *should* be able to solve the problem as accurately as Matlab *if* you get everything set up right... I don't know ARPACK well enough to give detailed help to you, but:

– Matlab uses 'double precision' variables throughout by default. Fortran uses single precision variables by default. If you want accurate results from your Fortran code you *must* use double precision. So:

(a) make sure you declare all your Fortran variables to be double precision – note that if you rely of Fortran's implicit typing mechanism this won't happen.

(b) remember that if you type a number such as 1.234 or 5.67e8 in a Fortran program, that is *single* precision. If you want double precision you need to use 1.234D0 or 5.67D8. This is a common cause of loss of precision in Fortran, particularly in test case code.

If you identify any detailed Fortran problems, try asking in `comp.lang.fortran` – they're very helpful.

– Have you set the options and tolerances correctly? You can see roughly what options Matlab uses when it calls ARPACK by looking at the Matlab documentation and default parameter. Start at:

(<http://www.mathworks.com/access/helpdesk/help/techdoc/index.html?/access/helpdesk/help/techdoc/ref/eigs.html>)

I hope someone else can give more advice.

Good luck,
andy

.