Subject: Re: generalized eigenvectors
Posted by Ralf Flicker on Mon, 15 Apr 2002 21:54:25 GMT
View Forum Message <> Reply to Message

Hey Randall - this is not really related to the original question, but since you seem to know what you're on about I figured I should ask you.

In working on large sparse arrays it has become crucial to make the SVD more efficient than O(n^3), but that seems to be easier said than done. Do you know of an efficient implementation of the Laczos bidiagonalization with selective reorthogonalization? I have not been able to accomplish it - with complete, explicit reorthogonalization it actually gets even worse than O(n^3).

More to the point, has anyone _ever_ managed to bring down the SVD significantly below O(n^3) for sparse arrays? Pointers and suggestions welcome.

aloha
ralf

Randall Skelton wrote:

```
>
>  I am not sure I understand what you mean by a "Generalized Eigenvalue
>  problem".
>
>  IDL does have routines for computing the Eigenvalues and Eigenvectors.
>  From the IDL help:
>
>   EIGENQL  - Compute eigenvectors of a real, symmetric array, given the array.
>   EIGENVEC - Compute eigenvectors of a real, nonsymmetric array, given
>              the array and its eigenvalues.
>   ELMHES   - Reduce a real, nonsymmetric array to upper-Hessenberg form.
>   HQR      - Compute the eigenvalues of an upper-Hessenberg array.
>   TRIQL    - Compute eigenvalues and eigenvectors of a real, symmetric,
>              tridiagonal array.
>   TRIRED   - Use Householder's method to reduce a real, symmetric array to
>              tridiagonal form.
>
>  Most of these methods are described in the Numerical recipies books.
>  See http://www.ulib.org/webRoot/Books/Numerical_Recipes/
>
>  In IDL the user must decide if the input matrix is symmetric or not then
>  use the appropriate tools.  The Matlab EIG function basically uses the
>  same tools (to a first order), but automatically determines the "best"
```

> method based on your input matrix... I personally find Matlab's auto-magic
> approach to be more trouble than it is worth. Moreover, it promotes the
> idea that you don't really need to understand the numerical problem you
> are trying to solve...
>
> The general approach is:
>
> - If your matrix is real and symmetric, convert to the tridiagonal form
> (TRIRED) and then use the QR procedure (TRIQL) to iteratively find the
> eigenvalues/vectors from the tridiagonal array.
>
> - If your matrix is real and not symmetric, reduce to Hessenberg form
> using the Householder's transformation method (ELMHES) and then use the QR
> procedure (HQR) to get the eigenvalues/vectors of the upper Hessenberg
> matrix.
>
> - Unfortunately, there is no built in IDL code for the QZ (Golub and Van
> Loan, 1989) algorithm which can be modified to work for complex matricies.