An Eigendecomposition Updating Algorithm for Large Hermitian Matrices under Low Rank Perturbations

Martin J. Gander

June 3, 1998

1 Problem Description

In the analysis of absorption spectra for cristaline materials, one needs to solve for many values of the real parameter ω the linear system of equations

$$(H - \omega I)x = f,$$

where H is a large, dense, hermitian matrix, I is the identity and f is a complex vector. The problem is described in the frequency domain and when it is transformed into the spatial domain by an orthogonal transformation,

$$(F^H HF - \omega I)F^H x = F^H f,$$

the transformed matrix $\hat{H} := F^H HF$ displays an interesting structure. $\hat{H}$ is the sum of two matrices A and B where A is a banded matrix and B is zero except for a small block of non-zeros in the upper left corner. Both components are hermitian. Furthermore an eigenvalue decomposition of the banded matrix is known in analytic form, $AQ = Q\Lambda$.

It is therefore desirable to find an algorithm which solves the original linear system of equations efficiently for many values of ω using the known eigen decomposition of A.

2 An Updating Algorithm

Suppose we know for a given hermitian matrix A its eigen decomposition $AQ = Q\Lambda$ and we need to solve for many values of the real parameter ω the linear system of equations

$$(A + \rho bb^H - \omega I)x = f,$$

where b is a given complex vector and ρ is a real constant. Using the eigen decomposition of A, we find

$$(\Lambda + \rho Q^H bb^H Q - \omega I)Q^H x = Q^H f$$

or

$$(\Lambda + \rho z z^H - \omega I)\tilde{x} = \tilde{f}$$

where $z := Q^H b$, $\tilde{x} := Q^H x$ and $\tilde{f} := Q^H f$. Thus the system is diagonal except for the rank one perturbation $\rho z z^H$ caused by the rank one perturbation of the original problem.

There is a way of updating the eigenvalue decomposition in this situation. It consists of an easy extension of an updating method used in a divide and conquer algorithm for the tridiagonal real eigenvalue problem (Golub and Van Loan [1]) to the complex case. The following two results are needed:

Lemma 2.1 *Suppose $\Lambda = (\lambda_1, \ldots, \lambda_n)$ is a real diagonal matrix with $\lambda_1 > \ldots > \lambda_n$. Assume furthermore that ρ is a real parameter, not equal to zero and z is a complex vector of length n. If v is an eigenvector and μ the corresponding eigenvalue of the rank one perturbed problem*

$$(\Lambda + \rho z z^H)v = \mu v, \quad v \neq 0 \tag{1}$$

then $z^H v \neq 0$, and the eigenvalue μ is different from all the original eigenvalues, which implies $\Lambda - \mu I$ is nonsingular.

Proof The proof follows by contradiction as in the real case in [1]. Suppose $\mu = \lambda_i$ for some i. Then from (1) we have

$$(\Lambda - \mu I)v + \rho z^H v z = 0.$$

Looking at the $i - th$ component, we get from $\mu = \lambda_i$

$$\rho z^H v z_i = 0.$$

Since $\rho \neq 0$ and also $z_i \neq 0$ we must have $z^H v = 0$. Using this in (1) we find

$$\Lambda v = \mu v,$$

and thus v is an eigenvector of Λ. But Λ is diagonal with distinct eigenvalues which implies that the only nonzero component in v is the i-th component. But then $z^H v$ can not be zero, a contradiction. Thus μ can not equal λ_i for any i and hence $\Lambda - \mu I$ must be nonsingular and $z^H v \neq 0$. ■

Lemma 2.2 *Under the same assumptions as in the previous lemma, we find for the updated eigenvalue decomposition*

$$V^H(\Lambda + \rho z z^H)V = diag(\mu_1, \ldots, \mu_n)$$

with $\mu_1 \geq \ldots \geq \mu_n$ and $V = [v_1, \ldots, v_n]$ the relations

1. *The μ_i are the zeros of $f(\mu) = 1 + \rho z^H(\Lambda - \mu I)^{-1} z$.*
2. *If $\rho > 0$, then $\mu_1 > \lambda_1 > \mu_2 > \ldots > \lambda_n$. If $\rho < 0$, then $\lambda_1 > \mu_1 > \lambda_2 \ldots > \mu_n$.*

3. The eigenvector v_i is a multiple of $(\Lambda - \mu_i I)^{-1} z$.

Proof We proceed again as in Golub and Van Loan [1]. From equation (1) we have

$$(\Lambda - \mu I)v + \rho z^H v z = 0. \tag{2}$$

Because of the previous lemma, we know that $(\Lambda - \mu I)$ is not singular. Applying the inverse to the above equation from the left, we find that $v \in \text{span}\{(\Lambda - \mu)^{-1}z\}$ which proves 3. Furthermore applying $z^H(\Lambda - \mu I)^{-1}$ to equation (2) we find

$$z^H v(1 + \rho z^H (\Lambda - \mu I)^{-1} z) = z^H v f(\mu) = 0.$$

By the previous lemma $z^H v \neq 0$ and thus $f(\mu)$ must vanish for $\mu = \mu_i$ and thus 1 is established. To get the interlacing property 2, note that

$$f(\mu) = 1 + \rho \left(\frac{z_1 \overline{z}_1}{\lambda_1 - \mu} + \ldots + \frac{z_n \overline{z}_n}{\lambda_n - \mu} \right)$$

and its derivative is given by

$$f'(\mu) = \rho \left(\frac{z_1 \overline{z}_1}{(\lambda_1 - \mu)^2} + \ldots + \frac{z_n \overline{z}_n}{(\lambda_n - \mu)^2} \right).$$

Thus f is monotone between the poles and hence f has one root in each interval $(\lambda_i, \lambda_{i+1})$. Furthermore note that $f(\mu)$ tends to 1 as μ tends to $\pm\infty$. Thus for $\rho > 0$ the last root lies in the interval (λ_1, ∞) whereas for $\rho < 0$ the last root lies in the interval $(-\infty, \lambda_n)$. ∎

3 Using the Updating Algorithm

In the previous section we showed how the eigenvalues and eigenvectors of a given eigenproblem can be updated when the matrix is changed by a rank one update. In our application, the matrix is changed by adding a small block in the upper left corner,

$$\hat{H} = A + B = A + \begin{bmatrix} \tilde{B} & 0 \\ 0 & 0 \end{bmatrix}.$$

To use the rank one updating algorithm, we first compute an eigen decomposition of the small matrix $\tilde{B}$,

$$\tilde{B}V = VR$$

where $R = \text{diag}(\rho_1, \ldots, \rho_k)$ and $V = [v_1, \ldots, v_k]$. Denoting by $\tilde{v}_i$ the vectors v_i extended with zeros to the length n, we get

$$\hat{H} = A + B = A + \rho_1 \tilde{v}_1 \tilde{v}_1^H + \ldots + \rho_k \tilde{v}_k \tilde{v}_k^H$$

and can thus apply the updating algorithm k times to obtain the new eigen decomposition of $\hat{H}$ using the known eigen decomposition of A.

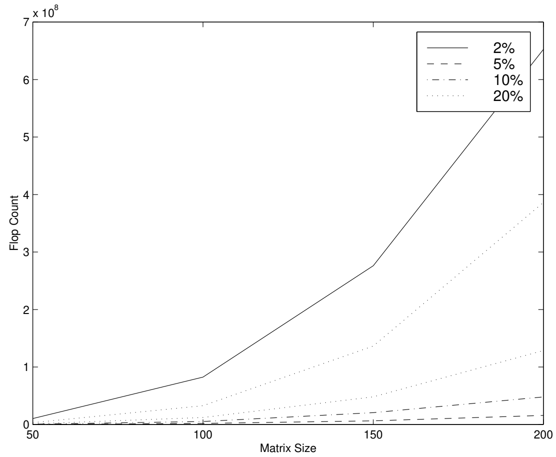

Figure 1: Comparison of the updating algorithm to the refactorization in flops for different problem sizes

4 Numerical Experiments

We performed numerical experiments on a given set of matrices $A \in C^{n \times n}$ for $n \in \{50, 100, 150, 200\}$ where the size of the small matrix $\tilde{B}$ to be inserted in the upper left corner has a size of 2%, 5%, 10%, 20% of the original matrix A. Figure 1 compares the updating algorithm to a refactorization of the new matrix. The calculations are performed in Matlab and the results are compared with a refactorization using the Matlab eig() function. The numerical results indicate that the updating algorithm brakes about even with the refactorization when the inserted matrix has a size of about 30% of the original matrix.

We are currently implementing the algorithm on a CRAY to solve a real problem from physics, where the original problem size is 16000×16000 and the small matrix to be inserted is of size 320×320 which corresponds to the case 2% above.

References

[1] Gene H. Golub and Charles F. Van Loan, Matrix Computations, Third Edition, The Johns Hopkins University Press, 1996.