Toward a Stable, Efficient, Nonsymmetric Eigensolver

Jason Slemons
University of Washington
Department of Applied Mathematics
17 July 2005

Overview

- Definitions and Decompositions
- Eigenvalue Problem
  - Symmetric
  - Unsymmetric
- Thesis proposal
  - dqds in the nonsymmetric case
  - formulations and stability
  - other topics

Definitions

- λ is an eigenvalue and v is an eigenvector for a matrix A: $Av = \lambda v$
- A is a symmetric matrix: $A = A^T$
- U is a unitary matrix: $U^{-1} = U^T$.
- T is a tridiagonal matrix:

$$T = \begin{pmatrix} b_1 & a_1 & & & & \\ c_1 & b_2 & \ddots & & & \\ & \ddots & \ddots & a_i & & \\ & & c_i & b_{i+1} & \ddots & \\ & & & \ddots & \ddots & a_{n-1} \\ & & & & c_{n-1} & b_n \end{pmatrix} \tag{1}$$

Definitions

- A is an upper hessenberg matrix:

$$A = \begin{pmatrix} b_1 & a_1 & * & * & * & * \\ c_1 & b_2 & \ddots & * & * & * \\ & \ddots & \ddots & a_i & * & * \\ & & c_i & b_{i+1} & \ddots & * \\ & & & \ddots & \ddots & a_{n-1} \\ & & & & c_{n-1} & b_n \end{pmatrix} \quad (2)$$

- $A \in \mathbb{R}$ is positive definite if: $x^T Ax > 0$ whenever $x \neq 0$.(Matrix Computations, Golub and Van Loan)

Decompositions

- A a general matrix, A has a similarity transform: $A = SBS^{-1}$(same eigenvalues)
- A a symmetric matrix, A has a conjugate transfrom: $A = SBS^T$(same inertia)
- A a symmetric matrix, U a unitary matrix, T a tridiagonal matrix, A has a reduction to tridiagonal form: $A = UTU^T$
- A a nonsymmetric matrix, U a unitary matrix, H an upper hessenberg matrix, A has a reduction to upper hessenberg form: $A = UHU^T$
- A a nonsymmetric matrix, T a tridiagonal matrix, A has a reduction to tridiagonal form: $A = STS^{-1}$

The General Eigenvalue Problem

Many different ways to skin this cat

- Symmetric case: Well mined
  - Divide and Conquer [5]
  - Bisection [2]
  - QR [7], [4], [2]
  - The Grail [3], [11]
- Nonsymmetric case: Not as well mined
  - Divide and Conquer [9]
  - QR
  - XHR [10], [8]

Typical Solution to the General Eigenvalue Problem

- Similarity reduction of A to tridiagonal system T(upper hessenberg matrix H if nonsymmetric)

$$A = QTQ^T \quad (3)$$
$$A = QHQ^T \quad (4)$$

- Find eigenvectors and eigenvalues of the tridiagonal matrix T(upper hessenberg matrix H if nonsymmetric) by reduction to diagonal matrix D or upper triangular matrix R, by unitary matrices Q_i:

$$T = Q_1 Q_2 ... Q_k D Q_k^T ... Q_2^T Q_1^T \quad (5)$$
$$H = Q_1 Q_2 ... Q_k R Q_k^T ... Q_2^T Q_1^T \quad (6)$$

- back transform eigenvectors of T or H to get those of the matrix A.

Eigenvalues of a Symmetric Matrix A

- State problem $Av = \lambda v$
- absolute accuracy $|\lambda_i(A) - \lambda_i(A + \delta A)| \leq ||f(A)||$
- an example:

$$A = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \tag{7}$$

$\lambda(A) = 1, 3$ exactly. Let $\delta A = 1000 * \epsilon_{mach} * A$. Then we solve $A + \delta A$.

$$A + \delta A = \begin{pmatrix} 1 + 2.2e - 13 & 2 + 4.4e - 13 \\ 2 + 4.4e - 13 & 1 + 2.e - 13 \end{pmatrix} \tag{8}$$

the larger eigenvalue $\lambda(A + \delta A) = 3(1 + 1000 * \epsilon_{mach})$, the smaller is similar.

Eigenvalues of a Symmetric Matrix A

- in this case $f(A)$ was $||\delta A||$.

$$
\begin{aligned}
|\lambda_i(A) - \lambda_i(A + \delta A)| &= 3 - 3(1 + 1000 * \epsilon_{mach}) && (9) \\
&= 3000 * \epsilon_{mach} && (10) \\
&= ||\delta A|| && (11)
\end{aligned}
$$

- Weyl's Theorem states that $|\lambda_i(A) - \lambda_i(A + \delta A)| \leq ||\delta A||$ for Symmetric A.
- condition of the problem: $sup_{\delta A}||\lambda(A) - \lambda(A + \delta A)||/||\delta A|| = 1$
- Relative accuracy would be $\frac{|\lambda_i(A) - \lambda_i(A+\delta A)|}{\lambda_i(A)} \leq ||\delta A||$
- not always possible to achieve for small λ_i.

Eigenvalues of a General Matrix A

- $Av = \lambda v$ where $A \neq A^T$
- Accuracy results are not as good;(Bauer-Fike)if $A = SDS^{-1}$ with D diagonal and $A + E$ is a perturbation then:

$$|\lambda(A) - \lambda(A+E)| \leq ||S|| * ||S^{-1}|| * ||E|| \qquad (12)$$

- Condition of an eigenvalue is $\frac{1}{w^T v}$ where w is a right eigenvector: $w^T A = \lambda w^T$
- Lack of fast algorithms as well(unlike $O(n^2)$ algorithms for eigenvalues in the symmetric case). Due to upper hessenberg form.(QR, orthogonal iteration)
- Algorithms that could use tridiagonalization may be faster, dqds?
- Stability can be an issue.

Eigenvalues of a General Matrix A

Consider the following matrix:

$$A : \; = \begin{pmatrix} \lambda_1 & b \\ 0 & \lambda_2 \end{pmatrix}$$

and the different perturbations

$$E_1 := \begin{pmatrix} 0 & 0 \\ \varepsilon & 0 \end{pmatrix} \qquad E_2 := \begin{pmatrix} \varepsilon & 0 \\ 0 & 0 \end{pmatrix} \qquad E_3 := \begin{pmatrix} \varepsilon & 0 \\ 0 & \varepsilon \end{pmatrix}$$

We have $\|E_i\|_\infty = \varepsilon$. The eigenvalues λ_j^i of $A + E_i$ are

$$\lambda_{1,2}^1 \;=\; \frac{1}{2}\left(\lambda_1 + \lambda_2 \pm \sqrt{(\lambda_1 - \lambda_2)^2 + 4b\varepsilon}\right)$$

$$\lambda_1^2 = \lambda_1 + \varepsilon, \qquad \lambda_2^2 = \lambda_2$$

$$\lambda_1^3 = \lambda_1 + \varepsilon, \qquad \lambda_2^3 = \lambda_2 + \varepsilon$$

Motivation for the unsymmetric tridiagonal

- Lanczos algorithm on an unsymmetric matrix (ABLE code).(LBNL, Osni Marques)
- Right now industry reformulates unsymmetic problem as symmetric because of speed and robustness.(MSC Software, Dr. Louis Komzsik)
- Fast, reliable method would be welcome.(MSC Software, and LBNL)
- Lessons can be learned from the symmetric case.
- Hessenberg form is slower($O(n^2)$ for each step) than tridiagonal form.
- There is almost no work being done on tridiagonal matrix solvers.
- There is no tridiagonal matrix iteration routine without serious stability issues.

A Lesson Learned from the Symmetric Case

- T is a symmetric positive definite tridiagonal
- Or B is a bidiagonal matrix and $BB^T = T$:

$i = 0$
repeat
  Choose shift τ_i smaller than the smallest eigenvalue of T_i.
  Compute Cholesky factorization of $T_i - \tau_i I = B_i^T B_i$
  $T_{i+1} = B_i B_i^T + \tau_i I$
  $i = i + 1$
until convergence

- T_i converges to an upper triangular matrix and $T_i = B_i T_i B_i^{-1}$.

A Lesson Learned from the Symmetric Case

- Start with Tridiagonal T, get the new $\hat{T}$, by rewriting the above as:

$T - \tau I = LU$
$UL - \tau I = \hat{L}\hat{U} = \hat{T}$
We can assume without loss that U has a diagonal of $u_1, ...u_n$ and a superdiagonal of 1's. L has subdiagonal $l_1, ...l_n$ and a diagonal of 1's.

$$L = \begin{pmatrix} 1 & & & & & \\ l_1 & \ddots & & & & \\ & \ddots & \ddots & & & \\ & & l_i & 1 & & \\ & & & \ddots & \ddots & \\ & & & & l_{n-1} & 1 \end{pmatrix} \tag{13}$$

A Lesson Learned from the Symmetric Case

$$U = \begin{pmatrix} u_1 & 1 & & & & \\ & u_2 & \ddots & & & \\ & & \ddots & 1 & & \\ & & & u_i & \ddots & \\ & & & & \ddots & 1 \\ & & & & & u_n \end{pmatrix} \tag{14}$$

$$LU = \begin{pmatrix} u_1 & 1 & & & & \\ l_1 u_1 & u_2 + l_1 & \ddots & & & \\ & \ddots & \ddots & 1 & & \\ & & l_i u_i & u_i + l_{i-1} & \ddots & \\ & & & \ddots & \ddots & 1 \\ & & & & l_{n-1} u_{n-1} & u_n + l_{n-1} \end{pmatrix} \tag{15}$$

A Lesson Learned from the Symmetric Case

$UL - \tau I = \hat{L}\hat{U}$ Written out element by element this is:

$$\hat{u_1} = u_1 + l_1 - \tau \tag{16}$$

$$\textbf{for } i = 1, n-1 \tag{17}$$

$$\hat{l_i} = \frac{l_i u_{i+1}}{\hat{u_i}} \tag{18}$$

$$\widehat{u_{i+1}} = l_{i+1} + u_{i+1} - \tau - \hat{l_i} \tag{19}$$

$$\textbf{endfor} \tag{20}$$

dqds, [12], [6], [1], [13]

rewriting $u_{i+1} - \hat{l}_i - \tau_i$ as a new variable d_i we get dqds:

$$\begin{aligned}
d_1 &= u_1 - \tau_1 && (21) \\
\textbf{for } i &= 1, n-1 && (22) \\
\hat{u}_i &= l_i + d_i && (23) \\
\hat{l}_i &= \frac{l_i u_{i+1}}{\hat{u}_i} && (24) \\
d_i &= d_i(\frac{u_{i+1}}{\hat{u}_i}) - \tau_i && (25) \\
\textbf{endfor} & && (26)
\end{aligned}$$

dqds

- d_i gets rid of subtractions in dqd.
- The d_i give a very good approximation of the smallest eigenvalue of T.
- Amazing relative accuracy: $6n * \epsilon$. Normally its just O($K(T) * \epsilon$) [2]
- Can find eigenvalues down to 10^{-309} when QR iteration or D & C just gives 10^{-16}. [6]
- Very fast because a good shift strategy is available using the d_i. Tridiagonal iteration is only order n for each step as well.
- Can use this solver in the non symmetric case. It preserves the tridiagonal form for non symmetric matrices.

Thesis Proposal

- Develop a nonsymmetric eigensolver based on dqds, with shifts.
- Formulate dqds in real arithmetic, including a shift by a complex numbers that keeps aritmatic real. [14]
- Formulate a stable dqds using blocks if possible. Given T factor $T = LDU$, with D block diagonal, to avoid blow up in L,U.
- Test each on different matrices, against XHR and QR.
- Test different shifts on these algorithms and explain how the 'd' relates to this shift.

Thesis Proposal: dqds real and complex arithmatic

- T nonsymmetric, $T = LU$ with L,U as before, so the dqds algorithm will work. Trouble is the accuracy of the eigenvalues as $||L||$,$||U||$ gets large.

- For the complex arithmetic algorithm test accuracy and stability against the algorithm that uses all real arithmatic and the blocked algorithm.

- For the algorithm with real arithmatic, use the shift by a complex number, developed in [14] and analyze backward stability.

- What should the shift strategy be?

- To improve accuracy, look at the shift that gives us a deflation and store it. Then re-run the algorithm again on those shifts. Restart algorithm if it breaks down, avoiding previous break down inducing shifts.

Thesis Proposal: dqds blocked formulation

- For $T = LDU$ factorizations find $\hat{T} = ULD$ or $\hat{T} = DUL$.

- Formulate where the blocks should go in D, in order to avoid element growth at that step. Research stability of that strategy.(T becomes block tridiagonal)

- Make sense of shifts, shift by 2×2 blocks? Make sense of the d_i from dqds shift strategy.

Thesis Proposal: High accuracy

- Algorithms exist that find high relative accuracy SVD of a matrix A given an RRD.(DGESVD)
  - These exist already in many cases
  - Propose to find remaining RRD's
- High relative accuracy algorithms exist, using the SVD algorithm above, that find eigenvalues and eigenvectors for symmetric A(Dopico, Molera)
  - Propose to find algorithm for nonsymmetric matrices
  - Start with skew hermitian case.(work in progress with Dopico and Molera)
- possible theoretical connection between XHR and dqds. like 2 LR step $=$ 1 QR step.

References

[1] Osni Marques Beresford Parlett. An implementation of the dqds algorithm(positive case). Technical Report LAWN 155, U. Tennessee, Computer Science Division, August 2002.

[2] J. Demmel. *Applied Numerical Linear Algebra*. SIAM, Philadelphia, 1997.

[3] I.J. Dhillon. *A new $O(n^2)$ Algorithm for computing the symmetric tridiagonal eigenvalue/eigenvector problem*. PhD thesis, U.C. Berkeley, 1997.

[4] J. G. F. Francis. The qr transformation parts 1, 2. *The Computer Journal*, 4:265–271 and 332–345, 1961.

[5] D. C. Sorensen, J. J. Dongarra. A fully parallel algorithm for the

symmetric eigenvalue problem. *SIAM J. Sci. Stat. Comp.*, 8(2):139–153, 1987.

[6] Beresford Parlett K. Vince Fernando. Accurate singular values and differential qd algorithms. *Numer. Math.*, 67:191–229, 1994.

[7] V. N. Kublanovskaya. On some algorithms for the solution of the complete eigenvalue problem. *USSR Computational Mathematics and Physics*, 3:637–657, 1961.

[8] Zhi-Shun Liu. On the extended hr algorithm. Technical report, Center for Pure and Applied Mathematics, U.C. Berkeley, 1992.

[9] P. Arbenz Loyce Adams. Towards a divide and conquer algorithm for the real nonsymmetric eigenvalue problem. *SIAM J. Matrix Anal. Appl.*, 15:1333–1353, 1994.

[10] Zhi-Shun Liu Osni Marques, Beresford Parlett. Numerical experiments with xhr. From personal communication with the authors.

[11] I. S. Dhillon R. A. van de Geijn P. Bientinesi. A parallel eigensolver for dense symmetric matrices based on multiple relatively robust representations. *SIAM Journal on Scientific Computing*, 27(1):43–66, 2005.

[12] Beresford Parlett. The new qd algorithm. *Acta Numerica*, pages 459–491, 1995.

[13] H. Rutishauser. *Lectures On Numerical Mathematics*. Birkhäuser, Boston, 1990.

[14] Zhuang Wu. *Triple dqds Algorithm for Complex Eigenvalues*. PhD thesis, U.C. Berkeley, 1996.

Backward error

- Backwards stability means that computed $\hat{L}$ and $\hat{U}$ solve a nearby problem$(L * U = A + \epsilon A)$
- theoretically LU has backward error:

$$\hat{L}\hat{U} = A + \delta A \quad (27)$$

$$\frac{||\delta A||}{||L||||U||} \leq O(e_{mach}) \quad (28)$$

- LU with pivoting has backward error:

$$\hat{L}\hat{U} = A + \delta A \quad (29)$$

$$\frac{||\delta A||}{||A||} \leq O(\rho e_{mach}) \quad (30)$$

where ρ is the growth factor that is bounded by 2^{n-1}. (matlab time)

Forward error

- Forward error is the error between the exact solution L,U and the computed solution, $\hat{L}$, $\hat{U}$.
- go through the accumulation of errors and hope you get an answer.
- its hard to calculate because it depends on the condition of the problem, usually a global thing.
- what about Gaussian Elimination?
- matlab time again