De we need yet another code for symmetric tridiagonals?

Rui Ralha (Univ. do Minho, Portugal)
Carlos Campos (ESTG, Portugal)

IWASEP7, Dubrovnik, June 9-12, 2008

A special indefinite symmetric tridiagonal matrix

$$GK = \begin{bmatrix} 0 & 10^{100} & & & & & \\ 10^{100} & 0 & 10^{20} & & & & \\ & 10^{20} & 0 & 10 & & & \\ & & 10 & 0 & 10 & & \\ & & & 10 & 0 & 10^{20} & \\ & & & & 10^{20} & 0 & 10^{100} \\ & & & & & 10^{100} & 0 \end{bmatrix}$$

$$\lambda(GK) = \pm\sigma(B)$$

$$B = \begin{bmatrix} 10^{100} & 10^{20} & & \\ & 10 & 10 & \\ & & 10^{20} & 10^{100} \\ & & & 0 \end{bmatrix}$$

Singular values of B

B defines well its singular values
LAPACK is able to compute them with high relative accuracy

LAPACK DBDSQR	MATLAB svd
1.000000000000000e+100	=
1.000000000000000e+100	=
1.414213562373095e+001	=
0	=

DBDSQR: dqds (for singular values only), QR iteration (when singular vectors are also wanted)

◇ J. W. Demmel and W. Kahan, *Accurate singular values of bidiagonal matrices, SIAM J. Sci. Stat. Comput.*, 11 (1990), pp. 873-912.

◇ K. Fernando and B. Parlett, *Accurate singular values and differential qd algorithms*, Num. Math.,67(1994), pp. 191-229.

Eigenvalues of GK with LAPACK 3.1.1 (Fev. 2007)

DSTERF	DSTEQR	DSTEDC
...	...	...
-5.7...E+045	-2.0...E-082	-5.7...E+045
-4.1...E-076	1.3...E-029	-5.5...E+083
3.5...E-044	2.4...E+084	3.5...E-044
...	...	...

- DSTERF uses the Pal-Walker-Kahan variant (square-root free) of the QR algorithm (eigenvalues only).
- DSTEQR uses the implicitly shifted QR algorithm (also eigenvectors).
- Both switch between QR and QL variants to handle graded matrices.
- DSTEDC uses the "Divide-and-Conquer" algorithm (eigenvalues and eigenvectors).

Eigenvalues of GK with LAPACK 3.1.1 (Fev. 2007)

DSTEMR	DSTEBZ (abstol=2*SFMIN)
...	...
0.000000	-14.14213562373 10
0.000000	-2.9 ... E-108
0.000000	14.14213562373 10
...	...

- DSTEMR uses dqds and bisection to compute eigenvalues. Numerically orthogonal eigenvectors (optional) are computed by the use of various suitable LDL^t factorizations near clusters of close eigenvalues (RRRs, Relatively Robust representations).
- DSTEMR fails to deliver good approximations for the eigenvalues $\pm 14.14\ldots$
- DSTEBZ uses bisection to compute some or all of the eigenvalues with prescribed accuracy.

Another indefinite symmetric tridiagonal matrix

$$GK + I = \begin{bmatrix} 1 & 10^{100} & & & & & \\ 10^{100} & 1 & 10^{20} & & & & \\ & 10^{20} & 1 & 10 & & & \\ & & 10 & 1 & 10 & & \\ & & & 10 & 1 & 10^{20} & \\ & & & & 10^{20} & 1 & 10^{100} \\ & & & & & 10^{100} & 1 \end{bmatrix}$$

- Eigenvalues (to 15 digits of accuracy): -10^{100}, -10^{100}, -13.1421356237310, 1.0, 15.1421356237310, 10^{100}, 10^{100}
- T defines well its eigenvalues (Ralha, 2008):

$$\frac{\left|\lambda_k - \widetilde{\lambda}_k\right|}{\left|\widetilde{\lambda}_k\right|} < 2.02n\varepsilon \left(1 + \frac{M}{\left|\widetilde{\lambda}_k\right|}\right)$$

  - M = second largest absolute value in main diag;
  - ε = bound for every entry-wise relative perturbation

Eigenvalues of GK+I with LAPACK 3.1.1 (Fev. 2007)

DSTERF	DSTEQR	DSTEDC
...	...	...
1.0	-13.14213562373 10	1.0
1.0	1.0	1.0
2.0E+062	15.14213562373 10	2.0E+062
...	...	...

DSTEMR	DSTEBZ (abstol=2*SFMIN)
...	...
1.0	-13.14213562373 10
1.0	1.0
1.0	15.14213562373 09
...	...

A symmetric positive definite sdd matrix

(J. Demmel, LAPACK Working Note #45)

$$T = \begin{bmatrix} 1 & 0.15 \cdot 10^{-16} & \\ 0.15 \cdot 10^{-16} & 10^{-32} & 0.15 \cdot 10^{-16} \\ & 0.15 \cdot 10^{-16} & 1 \end{bmatrix}$$

- It defines well its eigenvalues since $T = DAD$:

$$D = \begin{bmatrix} 1 & & \\ & 10^{-16} & \\ & & 1 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 0.15 & \\ 0.15 & 1 & 0.15 \\ & 0.15 & 1 \end{bmatrix} \text{ well cond.}$$

- Eigenvalues of T (to 16 digits of accuracy): 1, 1, $0.955 \cdot 10^{-32}$

The smallest eigenvalue of T with LAPACK 3.1.1 (Fev. 2007)

$\lambda = 0.955 \cdot 10^{-32}$

DSTEBZ	9.549999999999999E-033
DSTERF	9.550000000000001E-033
DSTEQR	-2.250000000000000E-034
DSTEDC	9.550000000000001E-033
DSTEMR	1.000000000000000E-032

Arithmetic bisection

- With stop. criteria

$$b_k - a_k < 2 * eps * \max(|a_k|, |b_k|)$$

and initial interval (Gershgorin)

$$[a_0, b_0] = [-0.3 \cdot 10^{-16},\ 1]$$

usual bisection takes 158 iterations to compute the interval

$[9.549999999999998e-033,\ 9.550000000000001e-033]$

- Takes 107 bisection steps to produce

$[9.016178841186530e-033,\ 1.517915466322569e-032]$

(endpoints a and b of the same magnitude).

Geometric vs arithmetic bisection (1)

$[a_0, b_0], (0 < a_0 < b_0) \quad \longrightarrow [a_k, b_k], b_k - a_k < tol$

- For $j = 1, \cdots, k$:
  - Arithmetic bisection (AB): $m_j = A(a_j, b_j) = (a_j + b_j)/2$
  - Geometric bisection (GB): $m'_j = G(a_j, b_j) = (a_j \cdot b_j)^{1/2}$ (bisects the interval of exponents);

$$0 < a_j < b_j \Rightarrow m'_j < m_j$$

- For endpoints of equal magnitude, the same convergence rate

$$\frac{b_0}{a_0} < 2 \Rightarrow \ |k(\text{AB}) - k(\text{GB})| \leq 1 \qquad \text{(for any } tol\text{)}$$

- "Interlacing property" (extreme cases):

$$\begin{aligned} a_0 \quad &\leftarrow \quad \cdots < m_{j+1} < m'_j < m_j < \cdots \\ \cdots \quad &< \quad m_{j-1} < m'_j < m_j < \cdots \rightarrow b_0 \end{aligned}$$

Geometric vs arithmetic bisection (2)

IF $a_0 \ll b_0$, to produce $[a_j, b_j]$ with $\frac{b_j}{a_j} < 2$:

- case 1: λ close to a_0, GB is faster:

$$\begin{aligned} j(AB) &> \log_2 \left(\frac{b_0 - a_0}{a_0} \right) \approx \log_2 \left(\frac{b_0}{a_0} \right) \\ j(GB) &> \log_2 \log_2 \left(\frac{b_0}{a_0} \right) \end{aligned}$$

- case 2: λ close to b_0, AB is faster:

$$\begin{aligned} j(AB) &= 1 \\ j(GB) &> \log_2 \log_2 \left(\frac{b_0}{a_0} \right) \end{aligned}$$

Geometric bisection in practice

CASE	NEW ITERATE
$a_0 < b_0 < 0$	$m_1' = -G(\lvert a_0 \rvert, \lvert b_0 \rvert) = -(a_0 \cdot b_0)^{1/2}$
$a_0 < 0 < b_0$	$m_1' = 0$
$a_j = 0$ ($j = 0$ or $j = 1$)	$m_{j+1}' = G(realmin, b_j)$
$b_j = 0$ ($j = 0$ or $j = 1$)	$m_{j+1}' = -G(\lvert a_j \rvert, realmin)$

- For $[a_0, b_0] = [-0.3 \cdot 10^{-16}, 1]$, $\lambda_1 = 0.955 \times 10^{-32}$, $\lambda_2 = \lambda_3 = 1$, GB takes 11 iterations to locate λ_1 in

$$[7.1\cdots \times 10^{-33}, 1.4\cdots \times 10^{-32}]$$

and an extra 5 iterations to locate λ_2 in

$$[5.\cdots \times 10^{-1}, 1]$$

(AB takes 107 and 0 iterations, resp.)

Newton-Raphson

$$T = \begin{bmatrix} d_1 & e_1 & & \\ e_1 & \ddots & \ddots & \\ & \ddots & \ddots & e_{n-1} \\ & & e_{n-1} & d_n \end{bmatrix}$$

The computed pivots in the LDL^t dec. of $T - xI$ are exact for

$$\begin{aligned} \widetilde{d}_k &= d_k (1 + O(\epsilon)) - x \cdot O(\epsilon) \\ \widetilde{e}_k &= e_k (1 + O(\epsilon)) \end{aligned}$$

Accurate computation if $|x|$ not much larger than $|\lambda|$.

- When to switch bisection$\rightarrow$ *NR*?
- 2 bisection steps + 3 NR iterations: $\lambda_1 = 9.550000000000058 \times 10^{-33}$.

Harmonic bisection

For $0 < a_j < b_j$,

$$\begin{aligned} H(a_j,b_j) &= \frac{2}{\frac{1}{a_j}+\frac{1}{b_j}} = \frac{2a_jb_j}{a_j+b_j} \\ &= \frac{1}{A(\frac{1}{a_j},\frac{1}{b_j})} \end{aligned}$$

- $a_0 \ll b_0 \Rightarrow H(a_0,b_0) \approx 2a_0$
- Harmonic bisection for $[a_j, b_j]$ $\Leftrightarrow$Arithmetic bisection for $[1/b_j, 1/a_j]$
- Good for eigenvalues close to a_0

Mixed precision algorithms

- Single precision arithmetic (SP) faster than double (DP) : Intel's Pentium IV (2 times faster), IBM's Cell Broad Engine processor (10 times faster),...
- Iterative algorithms adapt well to the mixed-precision paradigm:
  - SP to get "close enough",
  - DP in the last iterations (when convergence is usually faster)
- Work in iterative refinement for linear systems: Dongarra et al. (LAWN's 175, 177, 180)
- BISECTION: SP may be used to deliver intervals to be refined in DP. When to switch SP$\rightarrow$DP?

Interval arithmetic implementation of bisection (1)

$\lambda \in [a_j, b_j], x \in int\left([a_j, b_j]\right)$
For each $k = 1, ..., n$, compute $q_k^-(x)$ and $q_k^+(x)$ of equal sign:

$$q_k^-(x) \leq q_k(x) \leq q_k^+(x)$$

Wtith the rounding mode set to $+Inf$:

$$q_1^-(x) = -(x - d_1) \; ; q_1^+(\lambda) = d_1 - x$$

For $k = 2, ..., n$ [while sign $\left(q_k^-(x)\right) = \text{sign}(q_k^+(x))$]:

$$\begin{cases} q_k^-(x) = -\left[x + \left(e_{k-1}^2 / q_{k-1}^-\right) - d_k\right] \\ q_k^+(x) = d_k + \left[\left(-e_{k-1}^2 / q_{k-1}^+\right) - x\right] \end{cases}$$

If $k = n$, all the signs are correct: $[a_j, b_j] \rightarrow [a_{j+1}, b_{j+1}]$;
if $k < n$, then switch to double precision with $[a_j, b_j]$ (standard, not interval)

Interval arithmetic implementation of bisection (2)

EXAMPLE: For $\lambda = 9.55 \times 10^{-33} \in [a_0, b_0] = [-0.3 \cdot 10^{-16}, 1]$, GB in SP produces guaranteed intervals

$$
\begin{aligned}
[a_1, b_1] &= [0, 1] \\
[a_2, b_2] &= [\sqrt{realmin}, 1] \\
&\dots \\
[a_{11}, b_{11}] &= [7.1\cdots \times 10^{-33}, 1.4\cdots \times 10^{-32}] \quad (b_{11} < 2 \cdot a_{11}) \\
&\dots \\
[a_{32}, b_{32}] &= [9.5499967\cdots \times 10^{-33},\ 9.5500033\cdots \times 10^{-33}]
\end{aligned}
$$

and fails with $x = \sqrt{a_{32} \cdot b_{32}}$ since it produces

$$
\begin{aligned}
q_3^-(x) &= -1.6\cdots \times 10^{-6} \\
q_3^+(x) &= 1.8\cdots \times 10^{-6}
\end{aligned}
$$

Interval arithmetic implementation of bisection (3)

ADVANTAGES OF THE ALGORITHM:

- Uses faster single-precision arithmetic to deliver a correct $[a_j, b_j]$;
- Single-precision enough if low accuracy required for eigs;
- It is possible to deal with $q_k^- (x) = 0$ or $q_k^+ (x) = 0$ (if *Inf* is allowed)

LIMITATIONS OF THE ALGORITHM:

- In the SP phase, requires the computation of two sequences $q_k^- (x)$ and $\leq q_k^+ (x)$ for each point and comparison of signs;
- Requires rounding mode set to $+\infty$ (OR $-\infty$). Not available on some processors (Cell Processor,...)
- SP interval computations mail fail for points close to eigs of leading submatrices (not necessarily close to eigs of matrix); Rare in practice.

An alternative algorithm

IN SINGLE PRECISION:
$\{(a_i, b_i, scount(a_i), scount(b_i)), i = 1, \cdots, p\}$

IN DOUBLE PRECISION: for each i, compute eigs $\lambda_{scount(a_i)+1}$, ...,$\lambda_{scount(b_i)}$

- compute $dcount(a_i)$; if $dcount(a_i) > scount(a_i)$, correct a_i

$$\begin{aligned} \text{while } dcount(a_i) &> scount(a_i) \\ h_i &\leftarrow b_i - a_i \\ a_i &\leftarrow a_i - h_i \end{aligned}$$

  - doubles size of $[a_i, b_i]$ in each iteration
- compute $dcount(b_i)$: if $dcount(b_i) < scount(b_i)$, correct b_i in similar way

When to switch from SP to DP? (1)

Use SP while:

- $b_i - a_i > \epsilon_S \cdot (|a_i| + |b_i|)$ ("optimistic criteria")
  - relative errors $O(\epsilon_S)$ for well-defined eigs (good)
  - too many SP iterations for eigs not well defined $\Rightarrow$DP iterations required to enlarge the interval (bad)
- $b_i - a_i > \epsilon_S \cdot (|a| + |b|)$ ("pessimistic criteria")
  $[a, b]$ is Gershgorin's interval
  - SP always produces a correct interval (good)
  - switches prematurely to DP if eig is well defined (bad)

Optimization of the criteria requires us to know sharp bounds for the relative perturbations of the eigs

When to switch from SP to DP? (2)

The error depends upon the size of the diagonals entries, not $\|T\|$.

Use SP while

$$b_i - a_i > \epsilon_S \cdot \max\left\{\left(|a_i| + |b_i|\right),\ M\right\}$$

M = second largest absolute value in main diag

JUSTIFICATION: (Ralha, 2008)

$$\frac{\left|\lambda_k - \widetilde{\lambda}_k\right|}{\left|\widetilde{\lambda}_k\right|} < 2.02n\varepsilon\left(1 + \frac{M}{\left|\widetilde{\lambda}_k\right|}\right)$$

When to switch from SP to DP? (3)

Example

$$T = \begin{bmatrix} 1 & 10^6 & & & & & \\ 10^6 & 1 & 1 & & & & \\ & 1 & 1 & 1 & & & \\ & & 1 & 1 & 1 & & \\ & & & 1 & 1 & 10^6 & \\ & & & & 10^6 & 1 & \end{bmatrix}$$

$$\begin{aligned} \lambda_1 &= -9999\ 99.000\,000\,500\,000\,5 \\ \lambda_2 &= -999\ 999.000\,000\,499\,999\,5 \\ \lambda_3 &= 0.000\,000\,000\,001\,000\,000\,0 \\ \lambda_4 &= 1.999\,999\,999\,999 \\ \lambda_5 &= 100\ 000\ 1.000\,000\,499\,999\,5 \\ \lambda_6 &= 100\ 000\ 1.000\,000\,500\,000\,5. \end{aligned}$$

When to switch from SP to DP? (4)

Example (cont.)

$x_1, x_2, \cdots$ bisection points
"Optimal number" of SP iterations is p :

$$\begin{cases} scount\left(x_i\right) = dcount\left(x_i\right), \text{for } i \leq p \\ scount\left(x_{p+1}\right) \neq dcount\left(x_{p+1}\right) \end{cases}$$

$i - p$ (deviation from optimal number of SP iterations):

USE SP, while $b_i - a_i$	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6
$> \epsilon_S \cdot (\lvert a_i \rvert + \lvert b_i \rvert)$	+2	+2	+19	0	0	0
$> \epsilon_S \cdot (\lvert a \rvert + \lvert b \rvert)$	-7	-7	-19	-19	-9	-9
$> \epsilon_S \cdot \max\{(\lvert a_i \rvert + \lvert b_i \rvert), M\}$	+2	+2	+1	0	0	0

When to switch from SP to DP? (5)

SPLITTING PERTURBATIONS (Ralha08)

$$\widetilde{T} = \begin{bmatrix} 1+\eta_1 & 10^5(1+\delta_1) & \\ 10^5(1+\delta_1) & 10^5(1+\eta_2) & 10^5(1+\delta_2) \\ & 10^5(1+\delta_2) & 1+\eta_3 \end{bmatrix} = T + E$$

$$X = diag\left[(1+\delta_1)^{-1}(1+\eta_2)^{1/2},\ (1+\eta_2)^{-1/2},\ (1+\delta_2)^{-1}(1+\eta_2)^{1/2}\right]$$

$$X\widetilde{T}X = \begin{bmatrix} 1+\theta_1 & 10^5 & \\ 10^5 & 10^5 & 10^5 \\ & 10^5 & 1+\theta_3 \end{bmatrix} = T + D,\ \|D\| \ll \|E\|$$

ALGORITHM:

- sort entries of T in decreasing absolute value
- "clean" as many entries as possible in this sequence; $M = |T_{pq}|$, where T_{pq} is the largest entry not cleaned.

When to switch from SP to DP? (6)

POSITIVE-DEFINITE CASE

$M = \max_{1 \le k \le n} \left| T_{kk} \cdot \left(T^{-1}\right)_{kk} \right|$

- If $M \le 1$, use SP while $b_i - a_i > \epsilon_S \cdot (|a_i| + |b_i|)$
- If $M > 1$, use SP while $b_i - a_i > \epsilon_S \cdot M \cdot (|a_i| + |b_i|)$

JUSTIFICATION:

- $X\widetilde{T}X = T + D$, with X and D diagonal such that $\left\|X^T X\right\|$ small and D is $diag(T)$ with small relative perturbations (Ralha08)
- A Hermitian positive definite. Perturbation of a diagonal entry $\widetilde{A}_{jj} = A_{jj}\,(1 + \varepsilon) \Rightarrow$

$$\max_{1 \le k \le n} \frac{\left|\lambda_k - \widetilde{\lambda}_k\right|}{\left|\widetilde{\lambda}_k\right|} \le \varepsilon \left| A_{jj} \cdot \left(A^{-1}\right)_{jj} \right|$$

  (Mathias97).
- $\left(T^{-1}\right)_{kk}$ may be computed in about $6n$ flops (Dhillon98).

Bibliography

Demmel90 J. W. Demmel and W. Kahan, *Accurate singular values of bidiagonal matrices, SIAM J. Sci. Stat. Comput.*, 11 (1990), pp. 873-912.

Dhillon98 I. S. Dhillon, *Reliable Computation of the Condition Number of a Tridiagonal Matrix in $O(n)$ Time*, SIAM J. Matrix Anal. Appl., vol. 19, n. 3 (1998), pp.776-796.

Fernando94 K. Fernando and B. Parlett, *Accurate singular values and differential qd algorithms*, Num. Math.,67(1994), pp. 191-229.

Mathias97 Roy Mathias, *Spectral Perturbation Bounds For Positive Definite Matrices,* SIAM J. Matrix Anal. Appl., vol. 18, n. 4 (1997), pp.959-980.

Ralha08 Rui Ralha, *Perturbation Splitting For More Accurate Eigenvalues*, to appear in SIAM J. Matrix Anal. Appl.