Math.Comput.Sci. (2018) 12:173–181
https://doi.org/10.1007/s11786-018-0336-6

Mathematics in Computer Science

CrossMark

Mixed Precision Bisection

Rui Ralha

Received: 31 May 2017 / Revised: 29 January 2018 / Accepted: 31 January 2018 / Published online: 13 March 2018

Abstract We discuss the implementation of the bisection algorithm for the computation of the eigenvalues of symmetric tridiagonal matrices in a context of mixed precision arithmetic. This approach is motivated by the emergence of processors which carry out floating-point operations much faster in single precision than they do in double precision. Perturbation theory results are used to decide when to switch from single to double precision. Numerical examples are presented.

Keywords Eigenvalues · Bisection algorithm · Mixed precision

Mathematics Subject Classification 65F15

1 Introduction

As observed in [11], *We are now seeing growing use of mixed precision, in which different floating point precisions are combined in order to deliver a result of the required accuracy at minimal cost*. Single precision arithmetic (32 bits) is an attractive alternative to double precision because it halves the costs of storing and transferring data and there are processors around that carry out floating point operations faster in single precision. For instance, on Intel chips the SSE extensions allow single precision arithmetic to run twice as fast as double.

This new technological paradigm is likely to have a significant impact in the design of fast algorithms, namely in the area of numerical linear algebra. Even when one wishes to produce highly accurate results, the opportunity to exploit the fast single precision mode is not to be discarded. Iterative algorithms adapt well to this paradigm of mixed-precision arithmetic: single precision may be used to get close enough to the solutions, double precision will be used in the last iterations when convergence is usually faster. There is already a significant amount of work following this line of research for the solution of linear systems; for dense matrices, the authors in [3,12,13] use Gaussian elimination in single precision with iterative refinement of the solution to the full double precision accuracy; for sparse systems and mixed precision Krylov subspace methods see [4,9]. A mixed-precision QR decomposition on GPUs is presented in [19]. For the symmetric tridiagonal eigenvalue problem, a mixed precision MRRR algorithm may be found in [15].

R. Ralha (✉)
Center of Mathematics, School of Sciences, University of Minho, Braga, Portugal
e-mail: r_ralha@math.uminho.pt

We take this approach in the context of a bisection-like algorithm for computing the eigenvalues of symmetric tridiagonal matrices. The reader is supposed to be familiar with this method. Single precision will be used to produce real intervals, as narrow as possible, containing one or more eigenvalues, that will be finally refined with double precision arithmetic.

This simple picture is flawed for a subtle, often overlooked, reason: some eigenvalues may not be defined to full accuracy by the data (i.e., the entries of the tridiagonal). Single precision rounding errors can perturb some eigenvalues to more than single precision variation at each bisection step. On top of that, it is not at all rare for a single precision interval, passed on to DP arithmetic, to be empty.

There is an easy remedy that is justifiable but not fully satisfactory. For symmetric matrices all eigenvalues are perfectly conditioned with respect to the norm of the matrix. Thus, a stopping criterion of the form

$$\texttt{interval length} \leq O(\epsilon) \cdot \|T\| \tag{1}$$

where ϵ denotes the rounding error unit (adjusted for single and double precision), would be a safe stopping criterion. The blemish here is that $\|T\| / |\lambda|$ may be huge and λ may sometimes be defined to high relative accuracy. In this case more steps of bisection could have been performed in single precision. However, one should not carry out single precision bisection steps (at least, not too many) on intervals that are not guaranteed to contain a desired eigenvalue. Therefore, the main purpose of this paper is to use perturbation theory results to approach, as close as possible, the optimal point for switching from single to double precision arithmetic. In practical terms, we wish to replace, whenever possible, the right hand side of the criterion (1) with some quantity significantly smaller.

The outline of the remaining of the paper is as follows. In the next section we justify our choice of the mixed precision bisection algorithm and review the basic facts of the method; in Sect. 3 we present two numerical examples to support our claim that from all LAPACK codes to compute the eigenvalues of a symmetric tridiagonal matrix, DSTEBZ, which implements bisection, is the most accurate one; in Sect. 4 we describe the mixed precision implementation of the method; in Sects. 5 and 6 we discuss some criteria to reduce the number of double precision steps. We end up with some conclusions. Throughout the paper we use ϵ_s and ϵ_D to denote rounding error units in single and double precision arithmetic, respectively.

2 Why Mixed Precision Bisection

For the computation of the eigenvalues of symmetric tridiagonal matrices, there are algorithms of different flavors and the most popular are implemented in LAPACK [1]. For brevity, we organize them in 3 sets: (1) algorithms based upon similarity transformations (QR, dqds), (2) divide-and-conquer and (3) iterative refinement (this includes bisection). For the methods in (1) it is clear that there is no point in carrying out double precision similarity transformations of matrices which have been derived from the original matrix through single precision transformations. Similarly, for divide-and-conquer methods, the errors introduced at early stages will inevitably affect the accuracy of the computed eigenvalues. On the other hand, one may consider the use of those methods for computing single precision approximations to be improved with some iterative refinement carried out in double precision. If successful, such approach will give rise to codes that mix not only different precisions for the arithmetic but also different methods. Along these lines, there are many possible hybrid algorithms that may prove to be efficient. For instance, one may consider a two-stage dqds algorithm: in the first stage, approximations are computed in single precision and serve only as shifts to accelerate convergence in the second stage, in which similarity transformations are re-started with the initial matrix. However, there are non trivial issues to be addressed in this approach, namely the question of knowing whether this will reduce significantly the total number of transformations to be carried out in double precision as compared to the standard approach.

In our experience with LAPACK's routines, we found that bisection, as implemented in routine DSTEBZ, is indeed the unique method that consistently delivers approximations as accurate as possible (see Sect. 3). Furthermore, it is a versatile method that may be used to compute only a part of the spectrum. For these reasons, and also because the gain in speed from using a mixed precision bisection algorithm is unquestionable (on architectures whose arithmetic is faster in single precision than it is in double precision) and case independent, we consider a mixed precision bisection algorithm. A detailed description of the algorithm can be found in [8,10] or [14]. Let

$$T = \begin{bmatrix} d_1 & e_1 & & \\ e_1 & d_2 & \ddots & \\ & \ddots & \ddots & e_{n-1} \\ & & e_{n-1} & d_n \end{bmatrix} \tag{2}$$

be an $n \times n$ symmetric tridiagonal matrix. For any given real number x, if

$$T - xI = LDL^T, \tag{3}$$

where L is unit lower triangular and $D = diag(q_1(x), \dots, q_n(x))$ is diagonal, then

$$q_1(x) = d_1 - x \tag{4}$$

$$q_k(x) = d_k - x - e_{k-1}^2 / q_{k-1}(x), \quad k = 2, \dots, n. \tag{5}$$

According to Sylvester's law of inertia, the inertia of D equals the inertia of $T - xI$ so that the number of negative $q_k(x)$ gives the number of eigenvalues of T which are smaller than x. Following [7], we will use $count(x)$ to denote this number. The bisection method is able to compute the eigenvalues of a symmetric matrix which is very close to the exact one. In fact (see Lemma 5.3 in [8], p. 230), the values $q_k(x)$ computed in floating point arithmetic, using (4)–(5), have the same signs (and so compute the same inertia) as the $\widehat{q}_k(x)$ that would be obtained if exact arithmetic was carried out with the matrix $\widehat{T}$ such that

$$\widehat{d}_k = d_k \tag{6}$$

$$\widehat{e}_k = e_k (1 + \delta_k), \quad \text{with } |\delta_k| \leq 2.5\epsilon + O\left(\epsilon^2\right) \tag{7}$$

where, in accordance with our notation, we should replace ϵ with ϵ_s or ϵ_D, depending upon the precision of the arithmetic, single or double, respectively.

Therefore, the bisection method, correctly implemented, is able to deliver eigenvalues to high relative accuracy, even if they are much smaller than $\|T\|_2$, provided that T defines them well, i.e., small relative perturbations in its entries induce correspondingly small perturbations in its eigenvalues. Using the exception handling facilities of IEEE arithmetic, the computation produces a correct $count(x)$ even when some $q_{k-1}(x)$ in (5) is exactly zero. In this case, $q_k(x) = -\infty$, $q_{k+1}(x) = d_{k+1} - x$ and the computation continues unexceptionally [6,7]. A numerically robust, vectorized implementation of the algorithm is available in LAPACK's SSTEBZ (single precision) and DSTEBZ (double precision) routines. For parallel processing, care must be taken to ensure the correctness of the results. The logic of the bisection algorithm depends on $count(x)$ being a monotonic increasing function of x. However, depending upon the features of the arithmetic, monotonicity can fail and incorrect eigenvalues may be computed, because of rounding or as a result of using networks of heterogeneous parallel processors. In [7], several parallel algorithms are proposed and detailed analysis are carried out to ensure the correctness of the codes even when the arithmetic is non-monotonic. One of such algorithms has been implemented in ScaLAPACK [2]. For an implementation of the bisection algorithm on GPUs (Graphical Processing Units) see [17].

3 Bisection Versus Competitors (Accuracy of Eigenvalues)

In LAPACK, the routine DSTERF uses the Pal–Walker–Kahan variant (square-root free) of the QR algorithm for computing eigenvalues only, DSTEQR and DSTEDC use the implicitly shifted QR algorithm and divide and conquer algorithm, respectively, to compute eigenvalues and also eigenvectors. The routine DSTEMR uses dqds and bisection to compute eigenvalues and numerically orthogonal eigenvectors (optional) are computed with the use of various suitable LDL^T factorizations near clusters of close eigenvalues (Multiple Relatively Robust Representations). Finally, DSTEBZ uses bisection to compute some or all of the eigenvalues with prescribed accuracy. As already observed, DSTEBZ, with the appropriate stopping criterion, is the only routine that consistently delivers eigenvalues as accurate as the data warrant. To support our claim, we now present two examples.

Example 1 The matrix

$$T = \begin{bmatrix} 1 & 10^{100} & & & & & \\ 10^{100} & 1 & 10^{20} & & & & \\ & 10^{20} & 1 & 10 & & & \\ & & 10 & 1 & 10 & & \\ & & & 10 & 1 & 10^{20} & \\ & & & & 10^{20} & 1 & 10^{100} \\ & & & & & 10^{100} & 1 \end{bmatrix}$$

has eigenvalues (with fifteen correct digits) $\lambda_1 = -10^{100}$, $\lambda_2 = -10^{100}$, $\lambda_3 = -13.1421356237310$, $\lambda_4 = 1$, $\lambda_5 = 15.1421356237310$, $\lambda_6 = 10^{100}$, $\lambda_7 = 10^{100}$. We know that T defines well its eigenvalues (in [16] we have shown that a symmetric tridiagonal matrix defines to high relative accuracy all the eigenvalues of size not smaller than the second largest absolute value in the main diagonal, this is better explained in Sect. 5). Only DSTEQR and DSTEBZ (with abstol = 2SFMIN) compute λ_3, λ_4 and λ_5 correctly. For instance, DSTEMR delivers $\widetilde{\lambda}_3 = \widetilde{\lambda}_4 = \widetilde{\lambda}_5 = 1.0$.

Example 2 The symmetric positive definite matrix

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

defines well its eigenvalues[1] which, to fifteen digits of accuracy, are $\lambda_1 = 0.955 \times 10^{-32}$ and $\lambda_2 = \lambda_3 = 1$. This time, DSTERF and DSTEDC, together with DSTEBZ, are able to compute the smallest eigenvalue accurately but DSTEQR and DSTEMR compute $\widetilde{\lambda}_1 = -2.25 \times 10^{-34}$ and $\widetilde{\lambda}_1 = 1.0 \times 10^{-32}$, respectively.

4 From Single to Double Precision

In the following, we will use $count_S(x)$ and $count_D(x)$ to denote the computed values of $count(x)$ in single and double precision arithmetic, respectively. Suppose that a certain number of bisection steps have been carried out in single precision, producing an interval $[y, z]$ such that $p = count_S(y) < q = count_S(z)$. Now, we switch to double precision arithmetic to compute better approximations for the eigenvalues $\lambda_{p+1}, \ldots, \lambda_q$. Of course, our goal is to guarantee that the final approximations will be as accurate as those that would be produced if double precision was used from the very beginning. To this end, we must correct possible errors introduced in the single precision steps. Correction of the bound y will be necessary only if $count_D(y) > count_S(y)$; the bound z needs to be corrected only when $count_D(z) < count_S(z)$. For instance, if we have $count_S(y) = 3$ and $count_D(y) = 4$, we will trust that the four leftmost eigenvalues satisfy

$$\lambda_1 \le \lambda_2 \le \lambda_3 \le \lambda_4 < y$$

and to compute λ_4 we must therefore find another value y such that $count_D(y) = 3$. To simplify the presentation, we assume that we are looking for a specific eigenvalue, the kth one, say. What we need, to start double precision bisection steps, is an interval $[y, z]$ such that

$$count_D(y) < k, \; count_D(z) \ge k \tag{8}$$

[1] This matrix is scaled diagonally dominant, see [5].

and one of these conditions may fail in practice.[2] When this happens, we use an iterative procedure that updates the values of y and z in a way that doubles the width of the interval at each step. Let $[y_0, z_0]$ be the interval produced in single precision and $h = z_0 - y_0$. If $count_D(y_0) \geq k$, we set $z_1 = y_0$ and take tentatively $y_1 = z_1 - 2h$; if $count_D(y_1) < k$, we have produced an interval of width $2h$ which satisfies (8), otherwise, if $count_D(y_1) \geq k$, we set $z_2 = y_1$ and take $y_2 = z_2 - 4h$. The procedure stops when we find y_j such that $count_D(y_j) < k$ (at this point, the width of the interval $[y_j, z_j]$ is $2^j h$). The case $count_D(z_0) < k$ is treated in a similar manner. We will refer to this scheme as the "doubling procedure".

For the sake of the efficiency of the mixed precision bisection algorithm, the criteria for stopping the single precision phase should be adjusted to each case (matrix and eigenvalue). On one hand, if we stop too soon, we will carry out bisection steps in double precision that could have been done in faster single precision; on the other hand, too many bisection steps in single precision are a waste in two ways: firstly, because bisection steps of wrong intervals do not produce any useful information; secondly, because for each iteration that delivers a wrong $count_S$, a step of the doubling procedure will be needed in the process of retrieving an interval that satisfies (8). In the following we will denote by p the maximum number of steps with successive points $x_1, x_2, \ldots$, produced from an initial interval $[y, z]$ such that $count_S(x_i) = count_D(x_i)$ for $i = 1, \ldots, p$ but $count_S(x_{p+1}) \neq count_D(x_{p+1})$. Note that in the single precision phase we are not computing $count_D(x_i)$ but we would like to predict the optimal value p and carry out single precision bisection iterations till this point exactly. If one assumes that the eigenvalue being looked for may be computed to full relative accuracy, then we carry out single precision bisection steps while the following condition is verified

$$z - y > \epsilon_S \cdot (|y| + |z|) \tag{9}$$

(this is used in [10], p. 439, for the usual bisection algorithm without mixed precision arithmetic). This optimistic criterion should not be used unless we know that the eigenvalue is defined by the matrix entries to full relative accuracy. If this rule is violated then it is likely that we carry out a number of single bisection steps which is significantly larger than the optimal number p. We illustrate this with the following

Example 3 The eigenvalues of a matrix T, of order n, with diagonal entries $d_i = 2$ and off-diagonal entries $e_i = 1$ are

$$\lambda_r = 4\left(\sin\frac{\pi r}{2\,(n+1)}\right)^2, \quad r = 1, \ldots, n. \tag{10}$$

We start with the initial interval $[0, 4]$ to compute the smallest eigenvalue of such a tridiagonal matrix T of size $n = 100$. With sixteen digits correct, the eigenvalue is

$$\lambda_1 = 4\left(\sin\frac{\pi}{202}\right)^2 \approx 9.674354160238702e - 004. \tag{11}$$

If we carry out single precision bisection and use the stopping criterion given in (9), with $\epsilon_S = 2^{-24}$, 36 iterations will be done; in Table 1 we list the bisection points x_k, for $k = 26, \ldots, 36$, along with the corresponding values of $count_S(x_k)$. For each x_k, we also give, for the sake of comparison, the value $count_D(x_k)$. Note, however, that the sequence of points x_k produced with double precision arithmetic would not be the same.

Since $count_D(x_k) = count_S(x_k)$ for $k \leq 28$ and $count_D(x_{29}) \neq count_S(x_{29})$, the optimal number of single precision bisection steps is $p = 28$, that is, we should have switched to double precision after computing $[x_{28}, x_{26}]$, if we could guess it, of course. Single precision has produced the interval

$[x_{36}, x_{26}] = [\mathbf{9.674429}311417043\text{e}{-004}, \mathbf{9.674429}893493652\text{e}{-004}]$

[2] If the arithmetic is not monotonic, both conditions may actually fail; see [7] for details.

Table 1 The optimal number of single precision bisection steps here is $p = 28$

k	x_k	$count_S(x_k)$	$count_D(x_k)$
...	...	...	...
26	9.674429893493652e−004	1	1
27	9.674131870269775e−004	0	0
28	9.674280881881714e−004	0	0
29	9.674355387687683e−004	0	1
30	9.674392640590668e−004	0	1
31	9.674411267042160e−004	0	1
32	9.674420580267906e−004	0	1
33	9.674425236880779e−004	0	1
34	9.674427565187216e−004	0	1
35	9.674428729340434e−004	0	1
36	9.674429311417043e−004	0	1

as a candidate to contain the desired eigenvalue λ_1. If we now switch to double precision arithmetic and test these bounds, we get

$$dcount(x_{26}) = 1, \quad dcount(x_{36}) = 1$$

and conclude that $[x_{36}, x_{26}]$ does not, in fact, contain λ_1. Because we carried out 8 steps too many, in single precision, we need an equal number of steps of the ”doubling procedure”, in double precision, to get an interval $[y, z]$ that satisfies (8), plus the number of iterations required to refine this interval to a prescribed accuracy.

5 A Better Switching Criterion

In alternative to the criterion (9) which, as illustrated in the previous example, may allow a number of single precision bisection steps significantly larger than the optimal p, one may use the following

$$z - y > \epsilon_S \cdot \max\{|a|, |b|\} \tag{12}$$

where $[a, b]$ is the initial interval containing all eigenvalues. This is a natural criterion when absolute errors of the order of magnitude of $\epsilon_S \cdot \|T\|_2$ are to be expected in the computed eigenvalues. For matrices with entries and eigenvalues of very different magnitudes, we may improve significantly the criteria. In [16] we have shown (see Theorem 3.1) that the bounds for the relative errors (in the eigenvalues) due to relative perturbations in the entries of T are proportional to the size of the diagonal entries; more precisely, if the relative perturbation in each entry is bounded by some quantity denoted by τ, we have, for each λ_k and the corresponding perturbed eigenvalue $\widetilde{\lambda}_k$,

$$\left|\lambda_k - \widetilde{\lambda}_k\right| < 2.02n\tau\left(M + \left|\widetilde{\lambda}_k\right|\right) \tag{13}$$

where M denotes the second largest absolute value of the diagonal entries. Therefore, for eigenvalues of magnitude not smaller than M the criterion (9) is quite adequate. For those which have magnitude smaller than M we should keep bisecting, in single precision, while $z - y > M \cdot \epsilon_S$. Therefore, we now replace condition (9) with the following

$$z - y > \epsilon_S \cdot (|y| + |z| + M) \tag{14}$$

Table 2 Positive (negative) integers account for too many (few) single precision steps

Deviation from p (optimal)	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6
Criterion (9)	2	2	19	0	0	0
Criterion (12)	– 7	– 7	– 19	– 19	– 9	– 9
Criterion (14)	2	2	1	0	0	0

where, for simplicity, we neglect the multiplicative constant proportional to n (for large matrices, the final condition should reflect the growth of the error with the size n). We illustrate the usefulness of criterion (14) with the following

Example 4 The matrix

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

has eigenvalues (the sixteen figures displayed are correct, see Example 1 in [16])

$$\lambda_1 = \lambda_2 = -999999.000\,000\,500\,0$$
$$\lambda_3 = 0.000\,000\,000\,0010000$$
$$\lambda_4 = 1.999\,999\,999\,999000$$
$$\lambda_5 = \lambda_6 = 1000001.000\,000\,500.$$

The question is how accurately can we compute λ_3 and λ_4 with bisection. If the absolute errors are of the order of magnitude of $\epsilon_S \cdot \|T\|_2 \approx 6 \times 10^{-2}$, then λ_4 will be approximated with only two or three correct places and for λ_3 we do not even get close to approximate its order of magnitude. This is not the case because we have $M = 1$, therefore λ_3 and λ_4 will exhibit errors of magnitude ϵ_S. The gain obtained with the new stopping criterion can be appreciated in Table 2 where we list the deviation from the optimal number of bisection steps for each one of the three stopping criteria

Note that both switching criteria (9) and (12) perform very poorly for the computation of λ_3. Criterion (9) is too optimistic since it assumes that λ_3 can be computed with full relative accuracy and this is not the case; for this reason, it allows 19 too many single precision steps. Criterion (12) is too pessimist and switches too soon to double precision. Finally, criterion (14) misses the optimal value by only one step.

6 More on Perturbations of Symmetric Tridiagonals

To show that the bisection method is not able, in general, to compute every eigenvalue with low relative error, Wilkinson [18, p. 307] used the matrix which we have also used in our Example 3 and made clear that for x such that $|x|$ is much smaller than the diagonal entries $d_k = 2$, many digits of x will be inevitably lost in the subtraction $d_k - x$.[3] If x_1 and x_2 are two numbers such that the computed values of the differences $d_k - x_1$ and $d_k - x_2$ are the same, for every $k = 1, \ldots, n$, then the computed sequence of the principal minors for $T - x_1 I$ and $T - x_2 I$ will be the same. Note that this may happen even when x_1 and x_2 are quite far apart in the relative sense. Wilkinson's argument has to be analyzed carefully. In fact, the argument is quite right for a matrix with constant main diagonal

[3] The method, as presented in [18], uses the Sturm sequence property of the leading principal minors of $(T - xI)$ but the discussion of the errors in $d_k - x$ applies to the formulae (4)–(5) in the same way.

(see also Proposition 2.3 in [16]) but there are tridiagonal symmetric matrices with diagonal entries of varying size for which the loss of digits of x in the subtractions $d_k - x$, for some d_k such that $|d_k| \gg |x|$, is quite harmless. This is the case of the matrix

$$T = \begin{bmatrix} 1 & e_1 & 0 \\ e_1 & d_2 & e_2 \\ 0 & e_2 & 1 \end{bmatrix} \tag{15}$$

whose eigenvalue $\lambda = 1$ does not depend upon the entries d_2, e_1 and e_2. Therefore, no matter how large $|d_2|$ is, the loss of digits of $x \approx 1$ in the difference $d_2 - x$ does not affect the accuracy of the computation of $\lambda = 1$. More generally, such errors will be always harmless when computing an eigenvalue that is well defined by the matrix entries and this does not imply that the eigenvalue can not have size that is smaller than $\max\{|d_k|\}$. In [16] we have proposed a technique to derive a new bound for the errors due to the relative perturbations in the entries of bigger size. For matrices with entries and eigenvalues of varying size, our bound may be much sharper than the classical ones. The idea of the technique is to express as many as possible of the relative perturbations in the larger entries in terms of a diagonal congruence (we say these entries to be "cleaned"). We recall here the technique. Referring to the previous matrix T, suppose that $|d_2|$, $|e_1|$ and $|e_2|$ are much bigger than unity. Then, with relative perturbations η_i and δ_i, we may write

$$\begin{aligned} \widetilde{T} &= \begin{bmatrix} 1\,(1+\eta_1) & e_1\,(1+\delta_1) & 0 \\ e_1\,(1+\delta_1) & d_2\,(1+\eta_2) & e_2\,(1+\delta_2) \\ 0 & e_2\,(1+\delta_2) & 1\,(1+\eta_3) \end{bmatrix} \\ &= X \underbrace{\begin{bmatrix} 1\left(1+\eta_1'\right) & e_1 & 0 \\ e_1 & d_2 & e_2 \\ 0 & e_2 & 1\left(1+\eta_3'\right) \end{bmatrix}}_{\widehat{T}} X \end{aligned}$$

where

$$X = diag\left((1+\delta_1)\,(1+\eta_2)^{-1/2} \quad (1+\eta_2)^{1/2} \quad (1+\delta_2)\,(1+\eta_2)^{-1/2}\right)$$
$$(1+\eta_1') = (1+\eta_1)\,(1+\delta_1)^{-2}\,(1+\eta_2)$$
$$(1+\eta_3') = (1+\eta_3)\,(1+\delta_2)^{-2}\,(1+\eta_2)\,.$$

On one hand, since $\left\|X^T X - I\right\|$ is very close to one, each eigenvalue $\widehat{\lambda}_k$ of $\widehat{T}$ is very close (in the relative sense) to the corresponding eigenvalue $\widetilde{\lambda}_k$ of $\widetilde{T}$; on the other hand, the absolute errors $\left|\widehat{\lambda}_k - \lambda_k\right|$ are all bounded by the norm of an additive perturbation matrix which is smaller than the norm of the additive perturbation in T. Therefore, in this case, and following our proposal in [16], we would take $M = 1$ (the size of the largest entry not cleaned in the congruence transformation) to use in the stopping criterion given in (14).

Finally, we note that if, in the matrix as given in (15), $|d_2|$, $|e_1|$ and $|e_2|$ are much smaller than unity, then it is possible to choose the congruence transformation such that the entries equal to one are the ones to be cleaned. By using the general procedure, fully described in [16], one may find M in each case (the largest absolute of the entries not cleaned) to be used in criterion (14).

7 Conclusions and Future Work

We have presented the bisection algorithm for the computation of the eigenvalues of symmetric tridiagonal matrices in a context of mixed precision arithmetic. It guarantees double precision results even if a significant number of the initial steps is carried out in single precision. The key issue is the determination of the number of steps to carry out in single precision before switching to double precision. In this direction, we used new error bounds for the eigenvalues of symmetric tridiagonal matrices and illustrated the usefulness of our bounds with numerical examples carried with our Matlab codes.

Sharper error bounds for the eigenvalues will improve the computational efficiency of a mixed precision bisection algorithm so that they deserve further investigation. Another line of research that was not explored in this paper is the use of mixed precision in the computation of the pivots $q_k(x)$ for the same point x. Based upon the bounds for the errors induced by perturbations on each entry, it is possible to use double precision only for those $q_k(x)$ that involve entries whose perturbations cause more damage to the eigenvalues accuracy.

References

1. Anderson, E., et al.: LAPACK Users' Guide. SIAM, Philadelphia (1999)
2. Blackford, L., et al.: ScaLAPACK Users' Guide. SIAM, Philadelphia (1997)
3. Buttari, A., et al.: Mixed precision iterative refinement techniques for the solution of dense linear systems. Int. J. High Perform. Comput. Appl. **21**, 457–466 (2007)
4. Buttari, A., et al.: Using mixed precision for sparse matrix computations to enhance the performance while achieving 64-bit accuracy. ACM Trans. Math. Softw. TOMS **34**(4), 17:1–17:22 (2008)
5. Demmel, J.W.: The inherent inaccuracy of implicit tridiagonal QR. LAPACK working note 15 (1992)
6. Demmel, J.W., Li, X.: Faster numerical algorithms via exception handling. IEEE Trans. Comput. **43**, 983–992 (1992)
7. Demmel, J.W., Dhillon, I., Ren, H.: On the correctness of some bisection-like parallel eigenvalue algorithms in floating point arithmetic. Electr. Trans. Numer. Anal. **3**, 116–149 (1995)
8. Demmel, J.W.: Applied Numerical Linear Algebra. SIAM, Philadelphia (1997)
9. Giraud, l, Haidar, A., Watson, L.: Mixed-precision preconditioners in parallel domain decomposition solvers. Lect. Notes Comput. Sci. Eng **60**, 357–364 (2008)
10. Golub, G., Van Loan, C.: Matrix Computations, 2nd edn. The Johns Hopkins University Press, Baltimore (1989)
11. Higham, N.: The rise of mixed precision arithmetic. https://nickhigham.wordpress.com/2015/10/20/the-rise-of-mixed-precision-arithmetic/. Accessed Nov 2015
12. Kurzak, J., Dongarra, J.: Implementation of mixed precision in solving systems of linear equations on the CELL processor. Concurr. Comput. Pract. Exp. **19**, 1371–1385 (2007)
13. Langou, J., et al.: Exploiting the performance of 32 bit floating point arithmetic in obtaining 64 bit accuracy (revisiting iterative refinement for linear systems. In: Proceedings of the ACM/IEEE 2006 Conference. IEEE Computer Society Press (2006)
14. Parlett, B.N.: The Symmetric Eigenvalue Problem. Prentice-Hall, Englewood Cliffs (1998)
15. Petschow, M., Quintana-Orti, E.S., Bientinesi, P.: Improved accuracy and parallelism for MRRR-based eigensolvers—a mixed precision approach. SIAM J. Sci. Comput. **36**(2), C240–C263 (2014)
16. Ralha, R.: Perturbation splitting for more accurate eigenvalues. SIAM J. Matrix Anal. Appl. **31**, 75–91 (2009)
17. Volkov, V., Demmel, J.W.: Using GPUs to accelerate the bisection algorithm for finding eigenvalues of symmetric tridiagonal matrices. LAPACK working note 197 (2008)
18. Wilkinson, J.H.: The Algebraic Eigenvalue Problem. Oxford University Press, London (1965)
19. Yamazaki, I., Tomov, S., Dongarra, J.: Mixed-precision Choleski QR factorization and its case studies on multicore CPU with multiple GPUs. SIAM J. Sci. Comput. **37**(3), C307–C330 (2015)