Solving the 3 × 3 Real Symmetric Eigenproblem

Carlos F. Borges†

June 19, 2018

Abstract

We develop an algorithm solving the 3×3 real symmetric eigenproblem. This is a common problem and in certain applications it must be solved many thousands of times, see for example [3] where each element in a finite element grid generates one. Because of this it is useful to have a tailored method that is easily coded and compact. Furthermore, the method described is fully compatible with development as a GPU based code that would allow the simultaneous solution of a large number of these small eigenproblems.

1 Reduction to arrow form

The traditional first step in solving any real symmetric eigenproblem is to use unitary similarity transformations to reduce the matrix to tridiagonal (Hessenberg) form. For a 3 × 3 this would typically involve a Givens rotation that eliminates the (1, 3) and (3, 1) elements. We will stray from this approach and instead rely on a unitary similarity that eliminates the (1, 2) and (2, 1) elements and transforms our original matrix into an ordered 3 × 3 real symmetric arrow matrix. Specifically, a matrix of the form

\[ A = \begin{bmatrix} \alpha_1 & 0 & \beta_1 \\ 0 & \alpha_2 & \beta_2 \\ \beta_1 & \beta_2 & \gamma \end{bmatrix} \]

with \( \alpha_1 \geq \alpha_2 \).

Although such a similarity can be constructed in several ways, we will use a Jacobi rotation (in effect, the eigenvectors of the principal 2 × 2) operating in the (1, 2)-plane. The rotation can always be constructed so that the resulting arrow will have the elements of the shaft properly ordered. It is important to construct the Jacobi rotation carefully and we do so using the algorithm presented in [1] which is demonstrably superior to the standard approach.

†Authors address: Department of Applied Mathematics, Naval Postgraduate School, Monterey, CA 93943. Email: borges@nps.edu

* I use an arrow structure rather than a tridiagonal for pedagogical reasons as I believe the derivations are easier to follow in this form. The two are equivalent under a simple permutation similarity.
1.1 Deflating the Arrow

We note that if any $\beta_j = 0$ then it is possible to set $\lambda_j = \alpha_j$ and deflate the matrix since $e_j$ is clearly an eigenvector [6]. This is known as $\beta$-deflation. A second type of deflation occurs if $\alpha_1 = \alpha_2$, in that case we can apply a $2 \times 2$ rotation similarity transformation in the $(1, 2)$-plane that takes $\beta_1$ to zero and creates a $\beta$-deflation. This is known as a combo-deflation.

Exact deflations are theoretically easy to handle, however, before proceeding we must address the issue of numerical deflation which happens whenever tiny changes in the matrix can lead to deflation. Because the matrix is only $3 \times 3$ we can deal with the problem in a very direct manner. We begin by constructing a Givens rotation in the $(1, 2)$-plane that takes $\beta_1$ to zero. The first step is to compute $h = \sqrt{\beta_1^2 + \beta_2^2}$. If $h = 0$ then the matrix is diagonal and we are finished. If not, then we let

$$G = \begin{bmatrix} \beta_2 / h & -\beta_1 / h & 0 \\ \beta_1 / h & \beta_2 / h & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$  

Then

$$GAG^T = \begin{bmatrix} \frac{\alpha_1 \beta_2 + \alpha_2 \beta_1^2}{h^2} & \frac{\alpha_1 \beta_1 + \alpha_2 \beta_1 \beta_2}{h^2} & 0 \\ \frac{\alpha_1 \beta_1 + \alpha_2 \beta_2^2}{h^2} & \frac{\alpha_1 \beta_1^2 + \alpha_2 \beta_2^2}{h^2} & h \\ \frac{\alpha_1 \beta_1 + \alpha_2 \beta_2}{h} & \frac{\alpha_1 \beta_1 + \alpha_2 \beta_2}{h} & \gamma \end{bmatrix}$$

where $\alpha = \alpha_1 - \alpha_2$.

As the matrix is now tridiagonal we can adapt the deflation condition that is used in EISPACK (see [4] pp.352-353) to our cause which would have us deflate if

$$|\alpha \beta_1 \beta_2| \leq C \epsilon (|\alpha \beta_2^2 + \alpha_2 \beta_1^2| + |\alpha_1 \beta_1^2 + \alpha_2 \beta_2^2|)$$

for some constant $C$. We can simplify this condition by noting that

$$|\alpha_1 + \alpha_2| h^2 \leq |\alpha_1 \beta_2^2 + \alpha_2 \beta_1^2| + |\alpha_1 \beta_1^2 + \alpha_2 \beta_2^2|$$

by the triangle inequality. This leads to a simpler but slightly more restrictive test where if

$$|\alpha_1 \beta_1 \beta_2| \leq C \epsilon (|\alpha_1 + \alpha_2| h^2$$

we invoke the Wielandt-Hoffman theorem and ignore it yielding a deflation where we accept

$$\frac{\alpha_1 \beta_2^2 + \alpha_2 \beta_1^2}{h^2}$$

as an eigenvalue. The rest of the spectrum can be recovered by solving the $2 \times 2$ eigenproblem for

$$\begin{bmatrix} \frac{\alpha_1 \beta_2^2 + \alpha_2 \beta_1^2}{h^2} & h \\ h & \gamma \end{bmatrix}$$

If the matrix does not deflate then we can assume that it is both ordered and reduced, that is the strict inequality $\alpha_1 > \alpha_2$ holds, and further $\beta_j \neq 0$ for $i = 1, 2$. 


1.2 Solving the Eigenproblem for a $3 \times 3$ Ordered and Reduced Arrow

The interlacing property for real symmetric matrices combined with the fact that $\alpha_1 > \alpha_2$ implies that there is a rightmost eigenvalue $\lambda_1 > \alpha_1$ of multiplicity one with associated eigenvector $u_1$. If we shift $A$ by $\alpha_1$ we find that $\mu = \lambda_1 - \alpha_1$ is the only positive eigenvalue of

$$A - \alpha_1 I = \begin{bmatrix} 0 & 0 & \beta_1 \\ 0 & -\bar{\alpha} & \beta_2 \\ \beta_1 & \beta_2 & \gamma - \alpha_1 \end{bmatrix}$$

where $\bar{\alpha} = \alpha_1 - \alpha_2$. We shall call a symmetric matrix of this form (all zeros above the main counterdiagonal and a strictly negative middle element) a fully reduced arrow. The eigenvector associated with $\mu$ is identical with $u_1$ and it is easily verified that it is given by

$$u_1 = \begin{bmatrix} \beta_1 (\mu + \bar{\alpha}) \\ \beta_2 \mu \\ \mu (\mu + \bar{\alpha}) \end{bmatrix}$$

Observe next that

$$P(\alpha_2 I - A)P = \begin{bmatrix} 0 & 0 & -\beta_2 \\ 0 & -\bar{\alpha} & -\beta_1 \\ -\beta_2 & -\beta_1 & \alpha_2 - \gamma \end{bmatrix}$$

where $P$ is the simple permutation that swaps the first and second rows, is also a fully reduced arrow and that its only positive eigenvalue, $\nu$, satisfies $\nu = \alpha_2 - \lambda_3$. One can verify that its associated eigenvector is

$$\begin{bmatrix} \beta_2 (\nu + \bar{\alpha}) \\ \beta_1 \nu \\ -\nu (\nu + \bar{\alpha}) \end{bmatrix}$$

and hence we can recover that corresponding leftmost eigenvector of $A$ by applying the permutation $P$ to get

$$u_3 = \begin{bmatrix} \beta_1 \nu \\ \beta_2 (\nu + \bar{\alpha}) \\ -\nu (\nu + \bar{\alpha}) \end{bmatrix}$$

Because the eigenvectors must be orthogonal we can compute the middle eigenvector directly by taking the cross product $u_1 \times u_3$. This is better accomplished after a bit of algebra by using the form:

$$u_2 = \begin{bmatrix} -\beta_2 \mu (\nu + \bar{\alpha}) \\ \beta_1 \nu (\nu + \bar{\alpha}) \\ \beta_1 \beta_2 \bar{\alpha} \end{bmatrix}$$
It is worth noting that, discounting errors accrued in finding \( \mu \) and \( \nu \), there is no cancellation in the construction of any of the eigenvectors beyond the benign cancellation in computing \( \bar{\alpha} \).

Finally, we can use the trace identity

\[
\text{tr}(A) = \lambda_1 + \lambda_2 + \lambda_3
\]

to compute the corresponding eigenvalue. A bit of algebra yields

\[
\lambda_2 = \nu - \mu + \gamma
\]

In section 1.4 we develop an algorithm that computes the dominant eigenvalue of a fully reduced arrow.

1.3 Orthogonality of the Computed Eigenvectors

In this section we show that the computed eigenvectors will be numerically orthogonal provided that

\[
\hat{\mu} = \mu(1 + \delta_1)
\]

\[
\hat{\nu} = \nu(1 + \delta_2)
\]

where \( |\delta_i| < \epsilon \) for some small positive number \( \epsilon \).

We begin by noting that since \( u_2 \) is a cross product it will be necessarily be numerically orthogonal to \( u_1 \) and \( u_3 \) provided that these two are themselves numerically orthogonal to each other (see [5]) and so we only need demonstrate that. To that end we note that

\[
\hat{u}_1 = \begin{bmatrix}
\beta_1(\hat{\mu} + \hat{\alpha}) \\
\beta_2\hat{\mu} \\
\hat{\mu}(\hat{\mu} + \hat{\alpha})
\end{bmatrix}
= (1 + \delta_1)u_1 + \begin{bmatrix}
0 \\
0 \\
\mu^2(\delta_1 + \delta_1^2)
\end{bmatrix}
\]

and similarly

\[
\hat{u}_3 = (1 + \delta_2)u_3 + \begin{bmatrix}
0 \\
0 \\
\nu^2(\delta_2 + \delta_2^2)
\end{bmatrix}
\]

This leads us to the useful fact that

\[
\hat{u}_1^T\hat{u}_3 = \nu(\nu + \hat{\alpha})\mu^2(1 + \delta_2)(\delta_1 + \delta_1^2) + \mu(\mu + \hat{\alpha})\nu^2(1 + \delta_1)(\delta_2 + \delta_2^2) + \mu^2\nu^2(\delta_1 + \delta_1^2)(\delta_2 + \delta_2^2)
\]

\[
= \nu(\nu + \hat{\alpha})\mu^2\delta_1 + \mu(\mu + \hat{\alpha})\nu^2\delta_2 + O(\epsilon^2)
\]

Finally we note that \( \|u_1\| \geq \mu(\mu + \hat{\alpha}) \) and \( \|u_2\| \geq \nu(\nu + \hat{\alpha}) \) whence

\[
\frac{\hat{u}_1^T\hat{u}_3}{\|u_1\|\|u_2\|} \leq \frac{\mu^2}{\mu(\mu + \hat{\alpha})}\delta_1 + \frac{\nu^2}{\nu(\nu + \hat{\alpha})}\delta_2 + O(\epsilon^2) \leq 2\epsilon + O(\epsilon^2)
\]

and the computed eigenvectors are numerically orthogonal provided that condition [4] is satisfied.
1.4 Finding the Rightmost Eigenvalue of a Fully Reduced Arrow

In this section we develop two stable and efficient methods for finding the rightmost eigenvalue of a fully reduced arrow matrix

\[
\bar{A} = \begin{bmatrix}
0 & 0 & \beta_1 \\
0 & -\bar{\alpha} & \beta_2 \\
\beta_1 & \beta_2 & \bar{\gamma}
\end{bmatrix}
\]

Note that the block Gauss factorization of \( \bar{A} - \lambda I \) is

\[
\bar{A} - \lambda I = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
-\frac{\beta_1}{\lambda} & -\frac{\beta_2}{\lambda + \bar{\alpha}} & 1
\end{bmatrix}
\begin{bmatrix}
-\lambda & 0 & \beta_1 \\
0 & -(\lambda + \bar{\alpha}) & \beta_2 \\
0 & 0 & -f(\lambda)
\end{bmatrix}
\]

where \( f \), the spectral function of \( \bar{A} \), is given by

\[
f(\lambda) = \lambda - \bar{\gamma} - \frac{\beta_1^2}{\lambda} - \frac{\beta_2^2}{\lambda + \bar{\alpha}}.
\]

This is a rational Pick function with a pole at infinity. Inspection of the graph of the spectral function reveals that the elements of the shaft interlace the eigenvalues

\[
\lambda_1 > 0 > \lambda_2 > -\bar{\alpha} > \lambda_3.
\]

Moreover, the derivative of the spectral function is

\[
f'(\lambda) = 1 + \frac{\beta_1^2}{\lambda^2} + \frac{\beta_2^2}{(\lambda + \bar{\alpha})^2}.
\]

and is clearly bounded below by one so that its zeros are, in a certain sense, well determined. Furthermore, we note that the second derivative of the spectral function

\[
f''(\lambda) = -2 \left( \frac{\beta_1^2}{\lambda^3} + \frac{\beta_2^2}{(\lambda + \bar{\alpha})^3} \right)
\]

is strictly negative over the interval \((0, \infty)\) and therefore \( f'(\lambda) \) is strictly decreasing over the same interval.

1.4.1 Using the Borges-Gragg zero finder

One approach to finding the unique zero of \( f \) in the interval \((0, \infty)\) is to use the zero-finder developed in [2]. Let \( x_0 \) be an initial approximation to the eigenvalue. If \( x_j \) is known let our approximating function be

\[
\phi_j(x) = \omega_0 x - \sigma - \frac{\omega_1}{x}.
\]
If we select the constants $\sigma$, $\omega_0$, and $\omega_1$ so that

$$\phi_j^{(i)}(x_j) = f^{(i)}(x_j), \quad i = 0, 1, 2.$$  

then we will be able to guarantee cubic convergence. Therefore, we solve

$$
\begin{bmatrix}
-1 & x_j & -1/x_j & \omega_0 \\
0 & 1 & 1/x_j^2 & \omega_1 \\
0 & 0 & -2/x_j^3 & \sigma 
\end{bmatrix}
\begin{bmatrix}
\sigma \\
\omega_0 \\
\omega_1 
\end{bmatrix}
= 
\begin{bmatrix}
 f(x_j) \\
 f'(x_j) \\
 f''(x_j) 
\end{bmatrix}
$$

and find

$$\omega_1 = \beta_1^2 + \beta_2^2 \left(\frac{x_j}{x_j + \alpha}\right)^3,$$

$$\omega_0 = 1 + \beta_2^2 \frac{\alpha}{(x_j + \alpha)^3},$$

$$\sigma = \omega_0 x_j - \frac{\omega_1}{x_j} - f(x_j).$$

Note that $\omega_1 > \beta_1^2 > 0$ and also that $\omega_0 > 1$. The inequalities are strict and since both $\omega_0 > 0$ and $\omega_1 > 0$ it follows that $\phi_j$ is a Pick function and has a unique zero $x_{j+1} \in (0, +\infty)$.

Casual inspection of the error function

$$f(x) - \phi_j(x) = (1 - \omega_0)x - (\gamma - \sigma) - \frac{\beta_0^2 - \omega_1}{x} - \frac{\beta_2^2}{x + \alpha}$$

over the interval $(0, +\infty)$ reveals that it converges to $+\infty$ at the left boundary (as $x \to 0^+$) and to $-\infty$ at the right boundary (as $x \to +\infty$) and therefore crosses zero in the interval. Moreover, it is not hard to verify by differentiation and a bit of calculus that its derivative never changes sign in the interval. These facts imply that it crosses zero exactly once in the interval and further that we obtain monotonic convergence from any starting guess whatsoever $x_0 \in (0, +\infty)$. The cubic rate of convergence follows from (4).

Successive iterates can be found by solving quadratic equations. Rather than solve $\phi_j(x) = 0$ for $x_{j+1}$ it is better to solve

$$\phi_j(x_j - \Delta) = 0$$

for the increment $\Delta = x_j - x_{j+1}$. Some rearrangement using (4) reduces this to

$$a\Delta^2 + b\Delta - f = 0,$$

with

$$a = -\frac{\omega_0}{x_j},$$

$$b = f'(x_j) + \frac{f(x_j)}{x_j}.$$
If \( x_0 > \lambda_1 \) then \( f(\lambda) > 0 \) so \( b > f' > 1 \) and \( \Delta \) may be computed stably using

\[
\Delta = \frac{2f/b}{1 + \sqrt{1 + \frac{2a}{b} \frac{2f}{b}}},
\]

and we are therefore inclined to start to the right of \( \lambda_1 \). We can guarantee this by using the fact that the Borges-Gragg zero finder can start from \(+\infty\) and choosing the first iterate from \(+\infty\) to be our starting point. To do so, note that as \( x \to +\infty \) the approximate Pick function tends to

\[
\phi(x) = x - \tilde{\gamma} - \frac{\beta_1^2 + \beta_2^2}{x}.
\]

We propose to take \( x_0 \) to be the zero of \((7)\) in \((0, +\infty)\) which is

\[
x_0 = \frac{\tilde{\gamma}}{2} + \sqrt{\left(\frac{\tilde{\gamma}}{2}\right)^2 + \beta_1^2 + \beta_2^2}.
\]

If \( \tilde{\gamma} < 0 \) then we may wish to multiply top and bottom by the conjugate to avoid cancellation.

Termination in this case is straightforward. The mean value theorem gives

\[
f'(c) = \frac{f(x_j) - f(\lambda_1)}{x_j - \lambda_1} = \frac{f(x_j)}{x_j - \lambda_1}
\]

for some point \( c \in (\lambda_1, x_j) \). However, since the iteration is monotonic our iterates are on the right so that \( x_j > l_1 \). Since \( f'(\lambda) \) is strictly decreasing over the same interval we can use \( f'(x_j) \) as a lower bound on \( f'(c) \)

\[
f'(x_j) < \frac{f(x_j)}{x_j - \lambda_1}
\]

and we conclude that the absolute error\(^\dagger\) is bounded by

\[
|x_j - \lambda_1| < \frac{f(x_j)}{f'(x_j)}.
\]

It is therefore reasonable to terminate when

\[
\frac{f(x_j)}{f'(x_j)} < C e x_j
\]

for some constant \( C \) since that will imply that \( x_j \) has been found to high relative precision.

\(^\dagger\)The absolute value is unnecessary here but we leave it in for pedagogical reasons.
1.4.2 Using Newton’s Method

It is remarkably easy to find the unique zero of \( f \) in the interval \((0, +\infty)\) using Newton’s method. It is worth observing that because the derivative of the spectral function \( f'(x) \) is strictly decreasing and bounded below by 1 on the interval \((0, +\infty)\) Newton’s method will converge *monotonically* to the unique zero in the interval provided our initial guess, \( x_0 \), lies to the left of the zero. We can find an appropriate starting guess by finding the point where the spectral function intersects

\[
-\frac{\beta^2}{x + \alpha}
\]

since the graph of this function lies strictly below zero over the interval. After a bit of algebra and invoking the quadratic formula we find that this leads to a starting guess of

\[
x_0 = \frac{\gamma}{2} + \sqrt{\left(\frac{\gamma}{2}\right)^2 + \beta^2}.
\]

Termination in this case is also straightforward. Since this iteration is monotonic on the left we use the fact that \( f'(\lambda) > 1 \) over the interval \((x_j, \lambda_1)\) and we conclude that the absolute error is bounded by \( |x_j - \lambda_1| < |f(x_j)| \). It is therefore reasonable to terminate when \( |f(x_j)| < C \epsilon x_j \) for some constant \( C \) since that will imply that \( x_j \) has been found to high relative precision.

1.5 Testing

In this section we test the algorithm we have developed herein, which we call \texttt{threig}, versus the MATLAB function \texttt{eig} which uses routines from LAPACK. Our test is designed to compare the performance of our algorithm to that of the industry standard and therefore the protocol is simple.

1. Create a set of 100,000 random real symmetric test matrices where the elements of each test matrix \( T \) are randomly drawn from a known distribution (the distributions we test are uniform, normal, and chi-square).
2. Run both \texttt{threig} and \texttt{eig} on each test matrix \( T \).
3. Compute two measures of accuracy for each case:
   (a) \( \|I - V^T V\|_F \) which measures the orthogonality of the computed eigenvectors.
   (b) \( \|TV - V\Lambda\|_F \) which measures the accuracy of the computed spectral factorization (i.e. how well it reconstructs the test matrix.)
4. Finally we compute difference between the results from \texttt{eig} and those from \texttt{threig}, so that the output will be negative when the measured error in \texttt{threig} exceeds that of \texttt{eig} and positive otherwise. We then sort these values and generate a plot of the results. Since the results are sorted before being displayed the plot reveals how often and by how much one algorithm is *superior* to the other.
On viewing the figures we see that the proposed algorithm is consistently more accurate in both measures.

References

[1] C. F. Borges, An improved formula for Jacobi rotations, tech. report, Naval Postgraduate School, 2017.

[2] C. F. Borges and W. B. Gragg, A parallel divide and conquer algorithm for the generalized real symmetric definite tridiagonal eigenproblem, in Numerical linear algebra (Kent, OH, 1992), de Gruyter, Berlin, 1993, pp. 11–29.
Figure 2: Test matrix elements distributed $N(0, 1)$.

[3] J. de la Puente, M. Kser, M. Dumbser, and H. Igel, *An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes - iv. anisotropy*, Geophysical Journal International, 169 (2007), pp. 1210–1228.

[4] G. H. Golub and C. F. Van Loan, *Matrix computations*, vol. 3 of Johns Hopkins Series in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, second ed., 1989.

[5] W. Kahan, *Computing cross-products and rotations in 2- and 3-dimensional Euclidean spaces* tech. report, University of California at Berkeley, 2016.

[6] J. H. Wilkinson, *The algebraic eigenvalue problem*, Monographs on Numerical Analysis, The Clarendon Press, Oxford University Press, New York,
Figure 3: Test matrix elements distributed $\chi_1^2$.