Tridiagonalization Costs of the Bandwidth Contraction and Rutishauser-Schwarz Algorithms

Ian A. Cavers

Department of British Columbia
University of British Columbia

November 4, 1993

Abstract

In this paper we perform detailed complexity analyses of the Bandwidth Contraction and Rutishauser-Schwarz tridiagonalization algorithms using a general framework for the analysis of algorithms employing sequences of either standard or fast Givens transformations. Each algorithm's analysis predicts the number of flops required to reduce a generic densely banded symmetric matrix to tridiagonal form. The high accuracy of the analyses is demonstrated using novel symbolic sparse tridiagonalization tools, `Xmatrix` and `Trisymb`.

1 Introduction

Both the Bandwidth Contraction (BC) algorithm, a generalization of Schwarz's diagonally-oriented algorithm [Sch63], and the column-oriented Rutishauser-Schwarz (R-S) algorithm [Rut63, Sch68] use sequences of Givens similarity transformations to reduce a symmetric banded matrix to tridiagonal form. To simplify the complexity analysis of such algorithms we introduce a general framework for the analysis of algorithms using sequences of either standard or so-called *fast Givens* [Gen73] transformations. Using this framework we provide detailed analyses for standard and fast Givens variants of each algorithm, predicting the number of floating point operations required to reduce an $N \times N$ densely banded symmetric matrix, A, of bandwidth[†] b to tridiagonal form. Using several banded problems, we demonstrate the accuracy of each algorithm's analysis by checking their predicted operation counts with the symbolic sparse tridiagonalization tools `Xmatrix` and `Trisymb`.

Both `Xmatrix` and `Trisymb` estimate the flop[‡] requirements of a tridiagonalization by manipulating sparsity structures to simulate a matrix's reduction. `Xmatrix` is an interactive tool which allows a user to specify a small sparse symmetric matrix and select a sequence of Givens transformations to effect its reduction. Alternatively, `Trisymb` symbolically reduces large sparse problems using one of several preselected algorithms, including R–S and BC. `Xmatrix`, `Trisymb` and the analyses of this paper assume numerical cancellation does not occur.

[†] Bandwidth (or semi-bandwidth) is defined as $b = \max_{i,j \in \{1 \dots N\}, i \neq j} |i - j|$ such that $A_{ij} \neq 0$.

[‡] Following [GL89] a *flop* is defined to be any floating point arithmetic operation.

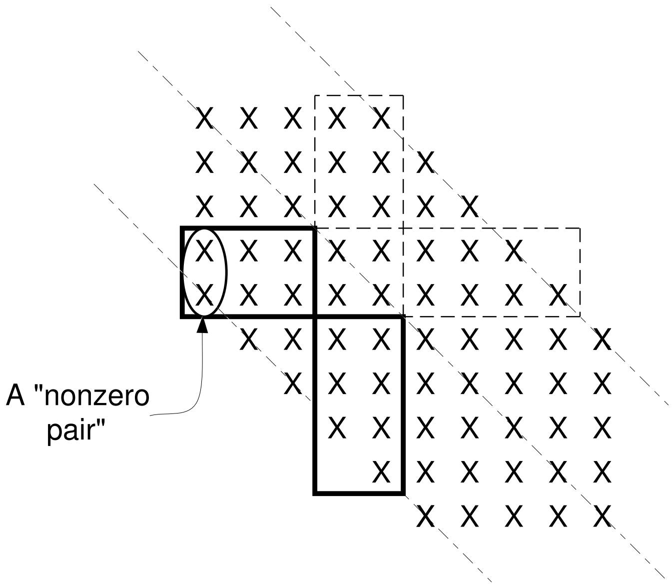

Figure 1: Transformation Length Example

All summations required by this report's analyses were resolved using Mathematica's [Wol91] symbolic summation package.

2 A Framework for Analysis

As previously mentioned, both the Bandwidth Contraction and Rutishauser-Schwarz algorithms use a sequence of Givens similarity transformations to reduce a matrix to tridiagonal form. As a result, we are able to investigate the complexity of both algorithms using a common analysis framework. Each Givens transformation

$$G(i,j,\theta)^T A\, G(i,j,\theta) \tag{1}$$

modifies both rows and columns i and j ($i < j$) of A. To exploit the symmetry of the banded problems and the similarity transformations, however, both algorithms need only consider modifications to the lower triangular portion of a matrix.

Each analysis splits the tridiagonalization operation count into two sub-tasks.

Task 1: Calculate the number of nontrivial transformations, T_{total}, used by the tridiagonalization.

Task 2: Calculate the total number of off-diagonal, lower triangular pairs of nonzero entries modified by the reduction's nontrivial transformations. We refer to this value as the total *transformation length* or L_{total}.

The first sub-task is self-evident but the second requires additional clarification. The length of a single transformation is the number of pairs of lower triangular nonzero entries it modifies, excluding those entries updated by both rotations constituting the transformation. We consider a pair of modified entries *nonzero* if one or both entries are nonzero. As an example, the length of the transformation modifying the highlighted entries of the matrix illustrated by Figure 1 is 7. (Section 2.3 considers a specialized variant of the analysis framework, for densely banded matrices, that exploits the sparsity of a pair of entries creating a bulge.) We note that a transformation's length is equal to the total number of pairs of nonzero entries on both sides of the main diagonal effected by the application of $G(i,j,\theta)^T$. As a result, it is often easier to consider the number of pairs of nonzero entries modified by a single rotation when symmetry is ignored, rather than apply the strict definition of transformation length.

In turn each analysis breaks down sub-tasks T_{total} and L_{total} into smaller sub-tasks to permit separate accounting of the requirements of band nonzeros elimination and bulge chasing. Once T_{total} and L_{total} have been found, we use the following general formula to calculate the algorithm's flop requirements.

$$\text{Total\_flops} \quad = \quad (F_{\text{trans}})(T_{\text{total}}) + (F_{\text{pair}})(L_{\text{total}}) + \text{OTC} \tag{2}$$

F_{trans} represents the number of flops required to construct a transformation and apply it to the entries modified by both the transformation's rotations. F_{pair} represents the number of flops required to apply a rotation to a single pair of nonzero entries. OTC represents one time costs that are not spread over individual transformations. Finally, the total flop count does not include the cost of square roots, which each analysis accounts for separately.

The specific values of F_{trans} , F_{pair} , and OTC are dependent upon whether the tridiagonalization algorithm uses standard Givens or fast Givens transformations. The following subsections refine Equation 2 for each transformation type.

2.1 Standard Givens Transformations

$\underline{F_{\text{pair}}}$

A standard 2×2 Givens rotation has the generic form $\begin{bmatrix} c & -s \\ s & c \end{bmatrix}$. Applying this rotation to a typical pair of entries $\begin{bmatrix} c & -s \\ s & c \end{bmatrix}^T \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$ requires

$$F_{\text{pair}} \ = 6 \text{ flops.} \tag{3}$$

$\underline{F_{\text{trans}}}$

The calculation of c and s requires 5 flops and one square root [GL89]. The cost of updating the 3 lower triangular entries modified by both rotations making up the transformation requires more detailed consideration. By using the following scheme, we save 3 flops over the most obvious approach.

$$\begin{bmatrix} \hat{a}_{ii} & \hat{a}_{ij} \\ \hat{a}_{ji} & \hat{a}_{jj} \end{bmatrix} \quad = \quad \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} a_{ii} & a_{ij} \\ a_{ji} & a_{jj} \end{bmatrix} \begin{bmatrix} c & -s \\ s & c \end{bmatrix}$$

$$
\begin{aligned}
&= \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} ca_{ii}+sa_{ij} & -sa_{ii}+ca_{ij} \\ ca_{ji}+sa_{jj} & -sa_{ji}+ca_{jj} \end{bmatrix} \\
&= \begin{bmatrix} c^2a_{ii}+csa_{ij}+csa_{ji}+s^2a_{jj} & -csa_{ii}+c^2a_{ij}-s^2a_{ji}+csa_{jj} \\ -csa_{ii}-s^2a_{ij}+c^2a_{ji}+csa_{jj} & s^2a_{ii}-csa_{ij}-csa_{ji}+c^2a_{jj} \end{bmatrix} \\
&\quad \text{but } a_{ji}=a_{ij} \\
&= \begin{bmatrix} c^2a_{ii}+2csa_{ji}+s^2a_{jj} & (c^2-s^2)a_{ji}+cs(a_{jj}-a_{ii}) \\ (c^2-s^2)a_{ji}+cs(a_{jj}-a_{ii}) & s^2a_{ii}-2csa_{ji}+c^2a_{jj} \end{bmatrix} \qquad (4)
\end{aligned}
$$

The total number of flops required to compute the final value of the twice modified entries $\hat{a}_{ii}$, $\hat{a}_{jj}$, and $\hat{a}_{ji}$, is summarized in the following table. Each calculation is free to use those values appearing to the left of it in the table.

Calculation	c^2	cs	s^2	$2csa_{ji}$	$\hat{a}_{ii}$	$\hat{a}_{jj}$	$\hat{a}_{ji}$	Total
Flops	1	1	1	2	4	4	5	18

Finally, it is not necessary to calculate the updated value of the eliminated entry, saving 3 flops per transformation. Thus for standard Givens transformations

$$F_{\text{trans}} = 5 + 18 - 3 = 20 \text{ flops.} \qquad (5)$$

<u>OTC</u>

There are no one time costs associated with tridiagonalization algorithms using standard Givens transformations.

<u>Standard Givens Flop Formula</u>

For standard Givens transformations Equation 2 becomes

$$\text{Total\_flops\_SG} = 20(T_{\text{total}}) + 6(L_{\text{total}}) \qquad (6)$$

In addition to this flop count, T_{total} square roots are required by a tridiagonalization.

2.2 Fast Givens Transformations

This section assumes that the reader is familiar with the fast Givens transformation presentation of [GL89]. Suppose that a series of fast Givens transformations are accumulated in a single similarity transformation Q^TAQ. In this case Q is equivalent to the product of a series of Givens rotations. The novel idea behind the fast Givens approach is to represent Q as the product of two matrices $MD^{-1/2}$. D is a diagonal matrix that is initially set to the identity. As the reduction proceeds the effects of each transformation are accumulated in D

$$D_{\text{new}} = M^TDM \qquad (7)$$

and this portion of the transformation is finally applied to the tridiagonal matrix at the end of the reduction.

$$T_{\text{final}} = D^{-1/2} T D^{-1/2} \tag{8}$$

On the other hand, each M is applied to A immediately to effect the elimination of nonzero entries. Following the presentation of [GL89], and using a 2×2 example for simplicity, M can take on one of two forms. We assume that M^T is applied to $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ to zero x_2.

$$\begin{array}{c|c} M_1 = \begin{bmatrix} \beta_1 & 1 \\ 1 & \alpha_1 \end{bmatrix} & M_2 = \begin{bmatrix} 1 & \alpha_2 \\ \beta_2 & 1 \end{bmatrix} \\ \text{where } \alpha_1 = \frac{-x_1}{x_2} \;\; \beta_1 = -\alpha_1(\frac{d_2}{d_1}) & \text{where } \alpha_2 = \frac{-x_2}{x_1} \;\; \beta_2 = -\alpha_2(\frac{d_1}{d_2}) \end{array}$$

$\underline{F_{\text{pair}}}$

Applying M_1 or M_2 to a typical pair of entries

$$\begin{bmatrix} \beta_1 & 1 \\ 1 & \alpha_1 \end{bmatrix}^T \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} 1 & \alpha_2 \\ \beta_2 & 1 \end{bmatrix}^T \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \tag{9}$$

requires

$$F_{\text{pair}} = 4 \text{ flops}. \tag{10}$$

$\underline{F_{\text{trans}}}$

We consider the cost of updating the 3 lower triangular entries modified by both M^T and M in detail. The cost of updating these entries using transformations constructed from either M_1 or M_2 is identical. Without loss of generality the following analysis considers M_1.

$$\begin{aligned}
\begin{bmatrix} \hat{a}_{ii} & \hat{a}_{ij} \\ \hat{a}_{ji} & \hat{a}_{jj} \end{bmatrix} &= \begin{bmatrix} \beta_1 & 1 \\ 1 & \alpha_1 \end{bmatrix} \begin{bmatrix} a_{ii} & a_{ij} \\ a_{ji} & a_{jj} \end{bmatrix} \begin{bmatrix} \beta_1 & 1 \\ 1 & \alpha_1 \end{bmatrix} \\
&= \begin{bmatrix} \beta_1 & 1 \\ 1 & \alpha_1 \end{bmatrix} \begin{bmatrix} \beta_1 a_{ii} + a_{ij} & a_{ii} + \alpha_1 a_{ij} \\ \beta_1 a_{ji} + a_{jj} & a_{ji} + \alpha_1 a_{jj} \end{bmatrix} \\
&= \begin{bmatrix} \beta_1^2 a_{ii} + \beta_1 a_{ij} + \beta_1 a_{ji} + a_{jj} & \beta_1 a_{ii} + \beta_1 \alpha_1 a_{ij} + a_{ji} + \alpha_1 a_{jj} \\ \beta_1 a_{ii} + a_{ij} + \beta_1 \alpha_1 a_{ji} + \alpha_1 a_{jj} & a_{ii} + \alpha_1 a_{ij} + \alpha_1 a_{ji} + \alpha_1^2 a_{jj} \end{bmatrix} \\
&\text{but } a_{ji} = a_{ij} \\
&= \begin{bmatrix} \beta_1^2 a_{ii} + 2\beta_1 a_{ji} + a_{jj} & \beta_1 a_{ii} + \beta_1 \alpha_1 a_{ji} + a_{ji} + \alpha_1 a_{jj} \\ \beta_1 a_{ii} + a_{ji} + \beta_1 \alpha_1 a_{ji} + \alpha_1 a_{jj} & a_{ii} + 2\alpha_1 a_{ji} + \alpha_1^2 a_{jj} \end{bmatrix}
\end{aligned} \tag{11}$$

The total number of flops required to compute the final value of the twice modified entries $\hat{a}_{ii}$, $\hat{a}_{jj}$, and $\hat{a}_{ji}$, is summarized in the following table. Each calculation is free to use those values appearing to the left of it in the table.

Calculation	$\beta_1 a_{ii}$	$\beta_1 a_{ji}$	$2(\beta_1 a_{ji})$	$\beta_1(\beta_1 a_{ii})$	$\alpha_1(\beta_1 a_{ji})$	$\alpha_1 a_{jj}$	$2\alpha_1 a_{ji}$	$\alpha_1(\alpha_1 a_{jj})$	$\hat{a}_{ii}$	$\hat{a}_{jj}$	$\hat{a}_{ji}$	Total
Flops	1	1	1	1	1	1	2	1	2	2	3	16

The next component of F_{trans} is the cost of updating the diagonal matrix D. For the moment we assume the first fast Givens transformation type has been selected.

$$\begin{aligned}\begin{bmatrix} \hat{d}_{ii} & 0 \\ 0 & \hat{d}_{jj} \end{bmatrix} &= M_1^T D M_1 \\ &= \begin{bmatrix} d_{jj}(1-\alpha_1\beta_1) & 0 \\ 0 & d_{ii}(1-\alpha_1\beta_1) \end{bmatrix} \end{aligned} \tag{12}$$

The calculation of $\hat{d}_{ii}$ and $\hat{d}_{jj}$ requires a total of 4 flops.

Determining the cost of constructing a fast Givens transformation is complicated by the required choice between two transformation types. The normal procedure is to first calculate α_1 and β_1 using 3 flops. To check the stability of this first transformation, the magnitude of $(1-\alpha_1\beta_1)$ is evaluated. (The cost of computing $(1-\alpha_1\beta_1)$ is included in the cost of updating D.) If $(1-\alpha_1\beta_1)$ is too large, the second fast Givens transformation type must be used and computing α_2 and β_2 requires 3 additional flops. Assuming the value of $\alpha_1\beta_1$ is saved, the new scaling factor $(1-\alpha_2\beta_2)$ can be computed from $-(1-\alpha_1\beta_1)/\alpha_1\beta_1$ using one additional flop. If we assume that 1/2 of the transformations employed are type 2, constructing the average fast Givens transformation requires

$$\frac{1}{2}(3+3+1) + \frac{1}{2}(3) = 5\text{flops}. \tag{13}$$

Finally, it is not necessary to calculate the updated value of the eliminated entry, saving 2 flops per transformation. Thus for fast Givens transformations

$$F_{\text{trans}} = 16 + 4 + 5 - 2 = 23 \text{ flops}. \tag{14}$$

<u>OTC</u>

When A has been reduced to tridiagonal form, the fast Givens process is completed as shown by equation 8. The calculation of $D^{1/2}$ requires N square roots. The following equation illustrates the modifications made to the tridiagonal matrix by entry $d_i^{-1/2}$.

$$\begin{aligned} &\left[\begin{array}{c|c|c} \ddots & & \\ \hline & d_i^{-1/2} & \\ \hline & & \ddots \end{array}\right] \left[\begin{array}{c|c|c} \ddots & b_i & \\ \hline b_i & a_i & b_{i+1} \\ \hline & b_{i+1} & \ddots \end{array}\right] \left[\begin{array}{c|c|c} \ddots & & \\ \hline & d_i^{-1/2} & \\ \hline & & \ddots \end{array}\right] \\ &= \left[\begin{array}{c|c|c} & b_i/d_i^{-1/2} & \\ \hline b_i/d_i^{-1/2} & a_i/d_i & b_{i+1}/d_i^{-1/2} \\ \hline & b_{i+1}/d_i^{-1/2} & \end{array}\right] \end{aligned} \tag{15}$$

By exploiting symmetry this update requires 3 flops. Generalizing this result to the cost of the entire update

$$\text{OTC} = 3N \text{ flops.} \tag{16}$$

Fast Givens Flop Formula

For fast Givens transformations Equation 2 becomes

$$\text{Total\_flops\_FG} \quad = \quad 23(T_{\text{total}}) + 4(L_{\text{total}}) + 3N. \tag{17}$$

In addition to this flop count, N square roots are required by the reduction.

As discussed in [GL89], fast Givens transformations require periodic rescaling to avoid overflow problems. Rescaling costs are difficult to predict and are not included in the analysis leading to Equation 17. Fortunately, Cavers [Cav93] reports that typically rescaling costs are insignificant when the Bandwidth Contraction or the Rutishauser-Schwarz algorithms are applied to large problems.

2.3 An Enhanced Framework for Densely Banded Matrices

In the general framework described above we increment transformation length if one or both entries in a modified pair are nonzero. For densely banded matrices, those transformations creating a bulge modify a single entry pair with only one nonzero. The zero entry in this pair is filled by the bulge. If the sparsity of this modified pair is exploited, each fast Givens transformation creating a bulge saves 3 flops, while a standard Givens transformation save 4 flops. If CR is the total number of nontrivial bulge chasing transformations used by the reduction then the enhanced flop formulas are given by the following equations.

$$\text{Total\_flops\_SG} \quad = \quad 20(T_{\text{total}}) + 6(L_{\text{total}}) - 4CR \tag{18}$$

$$\text{Total\_flops\_FG} \quad = \quad 23(T_{\text{total}}) + 4(L_{\text{total}}) + 3N - 3CR \tag{19}$$

The analyses of Sections 3 and 4 use the formulas given in Equations 18 and 19.

3 Bandwidth Contraction Tridiagonalization Costs

In this section we analyze the cost of reducing a densely banded matrix to tridiagonal form using Bandwidth Contraction. The analysis considers the cost of reducing the outermost nonzero subdiagonal and then extends this result to the entire tridiagonalization process.

3.1 Analysis Specific Assumptions and Definitions

Assumptions:

- Assume $2 \leq b < \frac{(N+1)}{2}$.

Definitions:

$Mod(x,y)$ $\Rightarrow$ The remainder from the division of integer x by integer y.

k $\Rightarrow$ The current bandwidth, or the k^{th} sub-diagonal which is currently being eliminated. $2 \leq k \leq b$

i $\Rightarrow$ The column index of the band nonzero, in the k^{th} subdiagonal, currently being eliminated from the lower triangular portion of the matrix.

BR_k $\Rightarrow$ The number of nontrivial transformations used to eliminate band nonzeros in the k^{th} subdiagonal.

BL_k $\Rightarrow$ The total length of band zeroing nontrivial transformations used to eliminate the k^{th} subdiagonal, including the twice modified entries.

CR_k **and** CL_k $\Rightarrow$ These variables are defined analogously to BR_k and BL_k but correspond to bulge chasing operations.

3.2 Tridiagonalization Analysis

Using the definitions of the previous subsection

$$T_{\text{total}} = \sum_{k=2}^{b}(BR_k + CR_k) \tag{20}$$

and

$$L_{\text{total}} = \sum_{k=2}^{b}(BL_k + CL_k - 2(BR_k + CR_k)). \tag{21}$$

We will analyze the requirements of BR_k, BL_k, CR_k and CL_k separately and then use Equations 18, 19, 20 and 21 to predict the flop requirements of the standard and fast Givens variants of the Bandwidth Contraction algorithm.

BR_k

$$BR_k = N - k \tag{22}$$

BL_k

Let $len_{i,k}$ be the number of nonzeros in the unioned structure of the two rows modified by the elimination of $A_{i+k,i}$.

$$len_{i,k} = \begin{cases} 2k+1 & 1 \leq i \leq N-2k \\ (N+1)-i & N-2k+1 \leq i \leq N-k \end{cases} \tag{23}$$

$$
\begin{aligned}
BL_k &= \sum_i len_{i,k} \\
&= \sum_{i=1}^{N-2k} (2k+1) + \sum_{i=N-2k+1}^{N-k} (N+1-i) \\
&= (2k+1)N - \frac{5k^2}{2} - \frac{3k}{2}
\end{aligned} \tag{24}
$$

<u>CR_k</u>

Let $bc_{i,k}$ be the number of transformations required to chase the bulge created by the elimination of $A_{i+k,i}$.

$$
\begin{aligned}
bc_{i,k} &= \left\lceil \frac{N-2k+1-i}{k} \right\rceil {}^{\S} \\
&< \frac{N-2k+1-i}{k} + 1 = \frac{N+1-i}{k} - 1
\end{aligned} \tag{25}
$$

If $\frac{N+1-i}{k} - 1$ is accepted as an approximation to $bc_{i,k}$ then, the total number of bulge chasing transformations will be significantly over estimated. Consider the band nonzeros in the k^{th} subdiagonal to be grouped into contiguous blocks of k nonzeros beginning in column $N-k$ and working back up the subdiagonal. The last block has $Mod(N-k,k)$ nonzeros. Within a block of k nonzeros, using $bc_{i,k} = \frac{N+1-i}{k} - 1$ over estimates the number of transformations by

$$
\sum_{j=1}^{k} \frac{j}{k} = \frac{k+1}{2}. \tag{26}
$$

Multiplying this value by the number of blocks of k nonzeros estimates the error in the total number of bulge chasing transformations used during the k^{th} subdiagonal's elimination.

$$
\left(\frac{N-k}{k}\right)\left(\frac{k+1}{2}\right) \tag{27}
$$

Unfortunately, if $N-k$ is not a multiple of k, Equation 27 inaccurately predicts the error introduced by $bc_{i,k}$ for the $Mod(N,k)$ entries of the last block. This final error can be corrected by adding the following nonanalytic term to Equation 27.

$$
\sum_{j=1}^{Mod(N,k)} \left(\frac{j}{k}\right) - \frac{Mod(N,k)}{k}\left(\frac{k+1}{2}\right) \tag{28}
$$

The total number of bulge chasing transformations used in the elimination of the k^{th} subdiagonal is given by the following equation.

[§] The intended definition of ceiling returns the smallest integer $\geq$ to the argument. eg $\lceil -0.2 \rceil = 0$, $\lceil -1.2 \rceil = -1$ and $\lceil 1.2 \rceil = 2$

$$
\begin{aligned}
CR_k &= \sum_i bc_{i,k} - (\text{analysis correction}) \\
&= \sum_{i=1}^{N-k} \left(\frac{N+1-i}{k} - 1 \right) - \left(\left(\frac{N-k}{k} \right) \left(\frac{k+1}{2} \right) + \sum_{j=1}^{Mod(N,k)} \left(\frac{j}{k} \right) - \frac{Mod(N,k)}{k} \left(\frac{k+1}{2} \right) \right) \\
&= \frac{N^2}{2k} - \frac{3N}{2} + k + \frac{Mod(N,k)}{2k} (k - Mod(N,k)) \qquad (29)
\end{aligned}
$$

For symmetric densely banded matrices with $b < (N+1)/2$ this result predicts the required number of bulge chasing transformations exactly.

$\underline{CL_k}$

We now turn to the calculation of the total length of bulge chasing transformations used in the elimination of the k^{th} subdiagonal. Unlike L_{total}, recall that CL_k includes the twice modified entries. As a result, during the analysis of CL_k we refer to *augmented* transformation lengths, which include twice modified entries.

When the column index of the bulge, c, is less than $N-2k$, the augmented length of the bulge chasing transformation is $2k+2$. When $N-2k \leq c \leq N-k-1$ the eliminating transformation has an augmented length in the range $k+2 \leq len_c \leq 2k+1$. Each bulge chasing sequence consists of zero or more transformations of augmented length $2k+2$ and one transformation whose length is in the range $k+2 \leq len_c \leq 2k+1$. The latter transformation chases the bulge off the end of the matrix to complete the sequence.

Once again consider the band nonzeros in the k^{th} subdiagonal to be grouped into contiguous blocks of k nonzeros beginning in column $N-k$ and working back up the subdiagonal. The last block has Mod(N,k) nonzeros. For a complete block of b nonzeros, the average length of the last transformations in each bulge chasing sequence is $\frac{3k+3}{2}$. Considering all complete blocks together, these transformations contribute a total augmented length of

$$
(BR_k - k - Mod(N,k)) \left(\frac{3k+3}{2} \right) \qquad (30)
$$

towards CL_k. Assuming the average length for the last transformation in each of the final block's $Mod(N,k)$ bulge chasing sequences may create significant errors when $Mod(N,k)$ is large relative to N. Alternatively, these transformations collectively contribute

$$
\sum_{j=1}^{Mod(N,k)} (k+1+j) \qquad (31)
$$

towards CL_k. Finally, the number of full length $(2k+2)$ bulge chasing transformations used for the k^{th} subdiagonal's elimination is $CR_k - (BR_k - k)$. ($BR_k - k > 0$ for all k since we assume $b < \left(\frac{N+1}{2}\right)$.)

$$
\begin{aligned}
CL_k &= (BR_k - k - Mod(N,k))\left(\frac{3k+3}{2}\right) + \sum_{j=1}^{Mod(N,k)} (k+1+j) \\
&\quad + (CR_k - (BR_k - k))(2k+2) \\
&= (1+\frac{1}{k})N^2 - (\frac{7}{2})(1+k)N + 3k^2 + 3k \\
&\quad + \left(\frac{1}{k}+\frac{1}{2}\right) Mod(N,k)\,(k - Mod(N,k)) \qquad (32)
\end{aligned}
$$

Tridiagonalization Requirements of Standard Givens Bandwidth Contraction

$$
\begin{aligned}
\text{Total\_flops\_SG} &= 20(T_{\text{total}}) + 6(L_{\text{total}}) - 4\sum_{k=2}^{b} CR_k \\
&= \left(6b - 6 + 8\sum_{k=2}^{b}\left(\frac{1}{k}\right)\right) N^2 + \left(22 - \frac{35b}{2} - \frac{9b^2}{2}\right) N \\
&\quad + b^3 + 4b^2 + 3b - 8 \\
&\quad + \sum_{k=2}^{b}\left(\frac{(8+3k)Mod(N,k)(k - Mod(N,k))}{k}\right) \qquad (33) \\
\text{Total\_roots\_SG} &= T_{\text{total}} \\
&= (\frac{N^2}{2})\sum_{k=2}^{b}(1/k) + (1/2)(1-b)N \\
&\quad + \sum_{k=2}^{b}\left(\frac{Mod(N,k)(k - Mod(N,k))}{2k}\right) \qquad (34)
\end{aligned}
$$

Tridiagonalization Requirements of Fast Givens Bandwidth Contraction

$$
\begin{aligned}
\text{Total\_flops\_FG} &= 21(T_{\text{total}}) + 4(L_{\text{total}}) + 3N - 3\sum_{k=2}^{b} CR_k \\
&= \left(4b - 4 + 10\sum_{k=2}^{b}\left(\frac{1}{k}\right)\right) N^2 + \left(22 - 16b - 3b^2\right) N \\
&\quad + \frac{2b^3}{3} + \frac{5b^2}{2} + \frac{11b}{6} - 5 \\
&\quad + \sum_{k=2}^{b}\left(\frac{(10+2k)Mod(N,k)(k - Mod(N,k))}{k}\right) \qquad (35) \\
\text{Total\_roots\_FG} &= N \qquad (36)
\end{aligned}
$$

Method	Densely Banded Problem	Flops	Nontrivial Transformations	Transformation Length (excluding twice mod)
Xmatrix	N=25,b=4	13832	302	1456
analysis	N=25,b=4	13832	302	1456
Xmatrix	N=35,b=4	28632	612	3076
analysis	N=35,b=4	28632	612	3076
Xmatrix	N=35,b=6	42810	802	4893
analysis	N=35,b=6	42810	802	4893
Xmatrix	N=35,b=10	65612	1028	8020
analysis	N=35,b=10	65612	1028	8020

Table 1: Checking the Accuracy of the BC Analysis (Standard Givens) with `Xmatrix`

Bandwidth	MFlops			Rel. Error
	Complete Analysis	Trisymb	Analytical Analysis	
3	16.280385	16.280385	16.2803743	6.6×10^{-7}
4	22.743429	22.743429	22.7434183	4.7×10^{-7}
6	34.318280	34.318280	34.318240	1.2×10^{-6}
8	44.881143	44.881143	44.8810824	1.4×10^{-6}
10	54.852698	54.852698	54.8526125	1.6×10^{-6}
15	78.292627	78.292627	78.2921249	6.4×10^{-6}
20	100.486957	100.486957	100.4857616	1.2×10^{-5}
25	121.920980	121.920980	121.9186018	2.0×10^{-5}
50	222.820510	222.820510	222.8037234	7.5×10^{-5}
75	317.311077	317.311077	317.2560013	1.7×10^{-4}
100	407.107773	407.107773	406.9876202	3.0×10^{-4}
200	727.951303	727.951303	727.0360045	1.3×10^{-3}
300	995.587953	995.587953	992.2741838	3.3×10^{-3}
400	1215.853395	1215.853395	1208.3886919	6.1×10^{-3}
500	1393.402045	1393.402045	1379.9094793	9.7×10^{-3}

Table 2: An Accuracy Check of the BC Analysis (Fast Givens Transformations) using `Trisymb` and Densely Banded Matrices (N=1000)

3.3 Analysis Verification

To assess the accuracy of the Bandwidth Contraction analysis, we have conducted experiments with `Xmatrix` and `Trisymb`. For 4 small problems, Table 1 compares the flop requirements determined with `Xmatrix` to the values predicted by Equation 33. The table also includes the number and length of transformations used by `Xmatrix` and the values predicted by our analysis. In each case transformation lengths and totals, and flop requirements are predicted exactly.

Similarly, for densely banded matrices of order 1000, columns 2 and 3 Table 2 compare the MFlop requirements predicted by Equation 35 to the corresponding counts predicted by `Trisymb`. Our analysis once again predicts the flop requirements of tridiagonalization exactly.

We can obtain an analytic approximation to BC's flop analysis by dropping the $Mod(N,k)$ terms from Equation 35. Flop counts predicted from the resulting formula are recorded in the fourth column of Table 2, along with their relative error in column 5. The relative error shows a general trend of reduced accuracy with increasing b. The increased error results from the estimate of the number and length of of bulge chasing transformations used by the analysis without the $Mod(N,k)$ terms. Despite this trend, the maximum relative error attained at $b = 500$ is 0.01. The approximating analytical formula is surprisingly accurate at lower bandwidths and when $b \ll N$ the $Mod(N,k)$ terms can be safely ignored without incurring large errors.

4 Rutishauser-Schwarz Tridiagonalization Costs

The analysis detailed in this section calculates the number of floating point operations required to reduce a densely banded symmetric matrix to similar tridiagonal form using the column-oriented Rutishauser-Schwarz algorithm.

4.1 Analysis Specific Assumptions and Definitions

Assumptions:

- Assume $2 < b \leq \frac{N}{2} - 1$.

Definitions:

$Mod(x,y) \Rightarrow$ The remainder from the division of integer x by integer y.

k $\Rightarrow$ The column currently under reduction. $1 \leq k \leq N-2$

i $\Rightarrow$ The row index, relative to the main diagonal, of the band nonzero ($A_{k+i,k}$) currently being eliminated from the lower triangular portion of the matrix. Except for the last b-1 columns, $2 \leq i \leq b$.

BR $\Rightarrow$ The number of nontrivial transformations used to eliminate band nonzeros during the tridiagonalization.

BL $\Rightarrow$ The total length of band zeroing nontrivial transformations used by the tridiagonalization, including the twice modified entries.

CR and CL $\Rightarrow$ These variables are defined analogously to BR and BL but correspond to bulge chasing operations.

4.2 Tridiagonalization Analysis

Using the definitions of the previous subsection

$$T_{\text{total}} = BR + CR \tag{37}$$

and

$$L_{\text{total}} = BL + CL - 2(BR + CR). \tag{38}$$

We analyze the requirements of BR, BL, CR and CL separately and then use Equations 18, 19, 37 and 38 to predict the flop requirements of the standard and fast Givens variants of the Rutishauser-Schwarz algorithm.

BR

$$\begin{aligned} BR &= (b-1)(N-b) + \sum_{j=2}^{b-1}(j-1) \\ &= (b-1)(N - \frac{b}{2} - 1) \end{aligned} \tag{39}$$

BL

Let $len_{k,i}$ be the number of nonzeros in the unioned structure of the two rows modified by the elimination of $A_{k+i,k}$. In addition, let $j = k - (N - 2b)$.

$$len_{k,i} = \begin{cases} b+i+1 & 1 \leq k \leq (N-2b) \\ b+i+1 & (N-2b+1) \leq k \leq (N-b-2) \text{ and } 2 \leq i \leq (b-j) \\ N-k+1 & (N-2b+1) \leq k \leq (N-b-2) \text{ and } (b-j+1) \leq i \leq b \\ N-k+1 & (N-b-1) \leq k \leq (N-2) \end{cases} \tag{40}$$

Assuming $b < \frac{(N)}{2}$:

$$\begin{aligned} BL &= \sum_{k,i} len_{k,i} \\ &= (N-2b)\sum_{i=2}^{b}(b+i+1) + \sum_{k=N-2b+1}^{N-b-2}\left(\sum_{i=2}^{b-j}(b+i+1) + \sum_{i=b-j+1}^{b}(N-k+1)\right) \\ &\quad + \sum_{k=N-b-1}^{N-b}(b-1)(N-k+1) + \sum_{k=N-b+1}^{N-2}(N-k-1)(N-k+1) \\ &= (\frac{3b^2}{2} + \frac{b}{2} - 2)N - \frac{4b^3}{3} - \frac{3b^2}{2} + \frac{5b}{6} + 2 \end{aligned} \tag{41}$$

CR

Let $bc_{k,i}$ be the number of transformations required to chase the bulge created by the elimination of $A_{k+i,k}$.

- if $1 \leq k \leq (N-2b-1)$

$$
\begin{aligned}
bc_{k,i} &= \left\lceil \frac{\begin{pmatrix}\text{column to chase} \\ \text{bulge beyond}\end{pmatrix} - (\text{column of } 1^{st} \text{ bulge})}{(\text{jump per chase})} \right\rceil \\
&= \left\lceil \frac{(N-b)-(k+i-1)}{b} \right\rceil \\
&< \frac{(N-b)-(k+i-1)}{b} + 1 = \frac{N+1-k-i}{b}
\end{aligned}
\tag{42}
$$

If $\frac{N+1-k-i}{b}$ is accepted as an approximation to $bc_{k,i}$ for k in the range $1 \leq k \leq (N-2b-1)$ then, the total number of bulge chasing transformations is over estimated. Consider the band nonzeros, for this range of k, to be grouped into contiguous blocks of b columns of lower triangular nonzeros beginning in column $N-2b-1$ and working back up to column 1. The last block has $Mod(N-1-b,b)$ columns. Each column contains $b-1$ lower triangular nonzeros which are to be eliminated. Within a block of b columns, using $bc_{k,i} = \frac{N+1-k-i}{b}$ over estimates the number of transformations by

$$\sum_{j=1}^{b} \frac{(b-1)j}{b} = \frac{b^2-1}{2}. \tag{43}$$

Multiplying this value by the number of blocks of b columns estimates the error in the total number of transformations predicted by $bc_{k,i}$.

$$\left(\frac{N-2b-1}{b}\right)\left(\frac{b^2-1}{2}\right) \tag{44}$$

Unfortunately, if $N-1-b$ is not a multiple of b, Equation 44 inaccurately predicts the error introduced by $bc_{k,i}$ for the $Mod(N-1,b)$ columns in the last block. This final error is corrected by adding the following term to Equation 44.

$$\sum_{r=1}^{Mod(N-1,b)} \left(\left(\frac{b+1}{2}\right) - \frac{r}{b}\right) - \frac{Mod(N-1,b)}{b}\left(\frac{b^2-1}{2}\right) \tag{45}$$

- if $(N-2b) \leq k \leq (N-b-2)$

Let $j = k - N + 2b$.

$$bc_{k,i} = \begin{cases} 1 & 2 \leq i \leq b-j \\ 0 & \text{otherwise} \end{cases} \tag{46}$$

• if $(N-b-1) \leq k \leq (N-2)$

$$bc_{k,i} = 0 \tag{47}$$

The total number of bulge chasing transformations used by the tridiagonalization is given by the following equation. Let $j = k - N + 2b$.

$$\begin{aligned}
CR &= \sum_k \sum_i bc_{k,i} - (\text{analysis correction}) \\
&= \sum_{k=1}^{N-2b-1} \sum_{i=2}^{b} (\frac{N+1-k-i}{b}) + \sum_{k=N-2b}^{N-b-2} \sum_{i=2}^{b-j} (1) \\
&\quad - \left(\left(\frac{N-2b-1}{b}\right) \left(\frac{b^2-1}{2}\right) + \sum_{r=1}^{Mod(N-1,b)} \left(\left(\frac{b+1}{2}\right) - \frac{r}{b} \right) - \frac{Mod(N-1,b)}{b} \left(\frac{b^2-1}{2}\right) \right) \\
&= \frac{(b-1)(N-b-1)^2}{2b} + \frac{Mod(N-1,b)}{2} \left(\frac{Mod(N-1,b)}{b} - 1 \right)
\end{aligned} \tag{48}$$

For symmetric densely banded matrices with $b \leq N/1 - 1$, this result predicts the required number of bulge chasing transformations exactly.

<u>CL</u>

Finally, we now turn to the calculation of the total length of bulge chasing transformations. Unlike L_{total}, recall that CL includes the twice modified entries. As a result, during the analysis of CL we refer to *augmented* transformation lengths, which include twice modified entries.

When the column index of the bulge, c, is less than $N - 2b$, the augmented length of the bulge chasing transformation is $2b + 2$. When $(N - 2b) \leq c \leq (N - b - 1)$ the eliminating transformation has an augmented length in the range $b + 2 \leq len_c \leq 2b + 1$. Consequently, each bulge chasing sequence consists of zero or more transformations of augmented length $2b + 2$ and one transformation whose length is in the range $b + 2 \leq len_c \leq 2b + 1$. The latter transformation chases the bulge off the end of the matrix.

For columns in the range $N - 2b \leq k \leq N - b - 2$, $\sum_{r=1}^{b-1}(b - r)$ band entries require bulge chasing. Each bulge chasing sequence consists of a single transformation. The augmented length of these bulge chasing transformations contribute

$$\sum_{k=N-2b}^{(N-b-2)} \sum_{r=1}^{(N-b-1-k)} (b + 1 + r) \tag{49}$$

towards CL.

Once again consider the lower triangular band nonzeros in columns $k \leq N - 2b - 1$ to be grouped into contiguous blocks of b columns beginning in column $N - 2b - 1$ and working back up to column 1. The last block has $Mod(N - 1 - b, b)$ columns. Each of the $b - 1$ band entries eliminated from these columns during tridiagonalization requires bulge chasing. For a complete block of b columns, the average length of the last transformation in each bulge chasing sequence is $\frac{3b+3}{2}$. Considering all complete blocks together, these transformations contribute a total augmented length of

$$(BR - b(b-1) - \sum_{r=1}^{b-1}(b-r) - (b-1)Mod(N-1,b))\left(\frac{3b+3}{2}\right) \tag{50}$$

towards CL. We cannot assign the average length to the last transformation in each of the final block's $(b-1)Mod(N-1,b)$ bulge chasing sequences. Alternatively, these transformations collectively contribute

$$\sum_{r=1}^{Mod(N-1,b)} (\sum_{j=1}^{b}(b+1+j) - (b+1+r)) \tag{51}$$

towards CL. Finally, the number of full length $(2b+2)$ bulge chasing transformations is $CR - (BR - b(b-1))$. ($BR - b(b-1) > 0$ since we assume $b \leq N/2 - 1$.)

$$\begin{aligned}
CL &= \sum_{k=N-2b}^{(N-b-2)} \sum_{r=1}^{(N-b-1-k)} (b+1+r) \\
&\quad +(BR - b(b-1) - \sum_{r=1}^{b-1}(b-r) - (b-1)Mod(N-1,b))\left(\frac{3b+3}{2}\right) \\
&\quad + \sum_{r=1}^{Mod(N-1,b)} \left(\sum_{j=1}^{b}(b+1+j) - (b+1+r)\right) + (CR - (BR - b(b-1)))(2b+2) \\
&= (b - \frac{1}{b})N^2 - (\frac{5b^2}{2} + 2b + -\frac{2}{b} - \frac{5}{2})N + \frac{5b^3}{3} + \frac{5b^2}{2} - \frac{2b}{3} - \frac{1}{b} - \frac{5}{2} \\
&\quad +(\frac{1}{b} + \frac{1}{2})Mod(N-1,b)(Mod(N-1,b) - b)
\end{aligned} \tag{52}$$

Tridiagonalization Requirements of the Standard Givens Rutishauser-Schwarz Algorithm

$$\begin{aligned}
\text{Total_flops_SG} &= 20(T_{\text{total}}) + 6(L_{\text{total}}) - 4CR \\
&= (6b - \frac{8}{b} + 2)N^2 - (6b^2 + 5b - \frac{16}{b} + 5)N \\
&\quad +2b^3 + 4b^2 - b - \frac{8}{b} + 3 \\
&\quad +(\frac{8}{b} + 3)Mod(N-1,b)(Mod(N-1,b) - b)
\end{aligned} \tag{53}$$

$$\begin{aligned}
\text{Total_roots_SG} &= T_{\text{total}} \\
&= (\frac{1}{2} - \frac{1}{2b})N^2 + (\frac{1}{b} - 1)N - \frac{1}{2b} + \frac{1}{2} \\
&\quad + \frac{Mod(N-1,b)}{2}\left(\frac{Mod(N-1,b)}{b} - 1\right)
\end{aligned} \tag{54}$$

Method	Densely Banded Problem	Flops	Nontrivial Transformations	Transformation Length (excluding twice mod)
Xmatrix	N=25,b=4	12264	216	1424
analysis	N=25,b=4	12264	216	1424
Xmatrix	N=35,b=4	25474	433	3027
analysis	N=35,b=4	25474	433	3027
Xmatrix	N=35,b=6	36762	481	4741
analysis	N=35,b=6	36762	481	4741
Xmatrix	N=35,b=10	54402	519	7509
analysis	N=35,b=10	54402	519	7509

Table 3: Checking the Accuracy of the Rutishauser-Schwarz Analysis (Standard Givens) with `Xmatrix`

Tridiagonalization Requirements of the Fast Givens Rutishauser-Schwarz Algorithm

$$\begin{aligned}
\text{Total_flops_FG} &= 21(T_{\text{total}}) + 4(L_{\text{total}}) + 3N - 3CR \\
&= (4b - \frac{10}{b} + 6)N^2 - (4b^2 + 3b - \frac{20}{b} + 10)N \\
&\quad + \frac{4b^3}{3} + \frac{5b^2}{2} - \frac{5b}{6} - \frac{10}{b} + 7 \\
&\quad + (\frac{10}{b} + 2)Mod(N-1, b)(Mod(N-1, b) - b) \qquad (55) \\
\text{Total_roots_FG} &= N \qquad (56)
\end{aligned}$$

4.3 Analysis Verification

Once again we assess the accuracy of the Rutishauser-Schwarz analysis using `Xmatrix` and `Trisymb` experiments. Table 3 compares `Xmatrix` results with those predicted by our analysis for the 4 small densely banded matrices of Section 3.3. In each case transformation lengths and totals, and flop requirements are predicted exactly.

Table 4 compares the MFlop predictions of our analysis to those reported by `Trisymb` for Section 3.3's group of densely banded matrices with $N = 1000$. Columns 2 and 3 compare the MFlop requirements predicted by Equation 55 to the corresponding counts predicted by `Trisymb`. Once again, our analysis predicts the flop requirements of tridiagonalization exactly.

As in Section 3, we can construct an analytic approximation to R–S's flop analysis by dropping the $Mod(N-1, b)$ terms from Equation 55. Flop counts predicted by the resulting formula are recorded in the fourth column of Table 4. The fifth column records the relative error of the analytical formula. In addition, Figure 2 plots the relative error of the same analytic formula.

Bandwidth	MFlops			Rel. Error
	Complete Analysis	Trisymb	Analytical Analysis	
3	14.618393	14.618393	14.6183930	0
4	19.419113	19.419113	19.4191265	7.0×10^{-7}
6	28.165012	28.165012	28.1650450	1.2×10^{-6}
8	36.463319	36.463319	36.4633418	6.3×10^{-7}
10	44.563554	44.563554	44.5635810	6.1×10^{-7}
15	64.384579	64.384579	64.3847230	2.2×10^{-6}
20	83.842609	83.842609	83.8426565	5.7×10^{-7}
25	103.038124	103.038124	103.0381816	5.6×10^{-7}
50	195.813174	195.813174	195.8132818	5.5×10^{-7}
75	283.705829	283.705829	283.7084402	9.2×10^{-6}
100	366.948249	366.948249	366.9484569	5.7×10^{-7}
200	656.106199	656.106199	656.1066070	6.2×10^{-7}
300	881.241029	881.241029	881.2814903	4.5×10^{-5}
400	1050.417059	1050.417059	1050.4980570	7.7×10^{-5}
499	1170.758753	1170.758753	1170.75975898	8.6×10^{-7}

Table 4: An Accuracy Check of the Rutishauser-Schwarz Analysis Analysis (Fast Givens Transformations) using `Trisymb` and Densely Banded Matrices (N=1000)

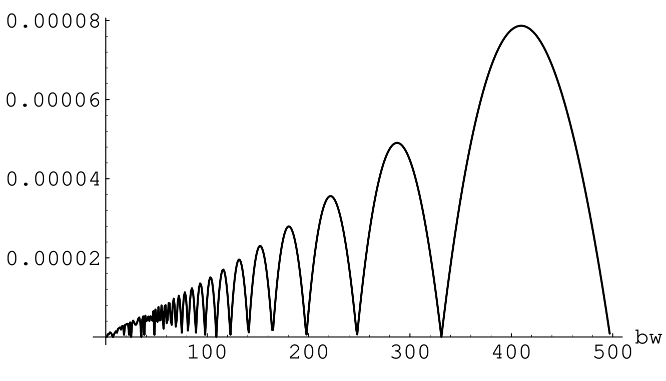

Figure 2: The Relative Error of the Analytical Formula for R–S, N=1000

In general, the relative error is smaller than observed for BC's analytical formula and R–S's approximating analytical analysis is relatively accurate for all experimental bandwidths.

5 Conclusion

This paper began by introducing a general framework for the analysis of algorithms using sequences of either fast or conventional Givens transformations. Using this framework we have provided detailed flop analyses for both the Bandwidth Contraction and Rutishauser-Schwarz tridiagonalization algorithms. Finally, we have shown that our analyses accurately predict the flop requirements of tridiagonalization for densely banded symmetric matrices of varying size and bandwidth.

References

[Cav93] Ian A. Cavers. A hybrid tridiagonalization algorithm for symmetric sparse matrices, 1993. To appear in the Siam Journal on Matrix Analysis.

[Gen73] W. M. Gentleman. Least squares computations by givens transformations without square roots. *J. Inst. of Maths. Applics.*, 12:329–336, 1973.

[GL89] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, second edition, 1989.

[Rut63] H. Rutishauser. On Jacobi rotation patterns. In *Experimental Arithmetic, High Speed Computing and Mathematics*, volume 15 of *Proceedings of Symposia in Applied Mathematics*, pages 219–239. AMS, April 1963.

[Sch63] H. R. Schwarz. Reduction of a symmetric bandmatrix to triple diagonal form. *Comm. ACM*, 6(6):315–316, June 1963.

[Sch68] H. R. Schwarz. Tridiagonalization of a symmetric band matrix. In J. H. Wilkinson and C. Reinsch, editors, *Linear Algebra*, volume II of *Handbook for Automatic Computation*, pages 273–283. Springer-Verlag, 1968.

[Wol91] Stephen Wolfram. *Mathematica: A System for Doing Mathematics by Computer*. Addison-Wesley, second edition, 1991.