A Survey of Numerical Aspects of Plane Rotations

Hammarling, Sven

1977

MIMS EPrint: **2008.69**

Manchester Institute for Mathematical Sciences
School of Mathematics

The University of Manchester

Reports available from: http://eprints.maths.manchester.ac.uk/
And by contacting: The MIMS Secretary
School of Mathematics
The University of Manchester
Manchester, M13 9PL, UK

ISSN 1749-9097

Middlesex Polytechnic Report Maths. 1.

A Survey of Numerical Aspects of Plane Rotations*

Sven Hammarling
Middlesex Polytechnic, Engineering, Science and Mathematics Resource Centre,
Enfield, Middlesex

October 1977

Abstract

In recent years the use of plane rotations in orthogonal factorizations has been increasing in popularity. This is in part due to modifications which enable computations with plane rotations to be carried out more quickly and in part due to the use of plane rotations in updating matrix factorizations and in other sparse applications.

A review of Jacobi, Givens and modified plane rotations and of products of plane rotations is given. The review includes discussion of the computational details required to avoid underflow and overflow when computing plane rotations, storage of plane rotations and the stability of plane rotations. Mention is also made of the possibility of using plane rotations for pivoting.

*This is a reprint of the original report, typeset using LaTeX, but otherwise essentially unchanged. April 2nd, 1996. The author's current address: The Numerical Algorithms Group, Wilkinson House, Jordan Hill Road, Oxford, OX2 8DR, UK. Input of this report to LaTeXwas performed while the author was on sabbatical leave at the Computer Science Department of the University of Tennessee at Knoxville and the kind hospitaility is warmly acknowledged.

Contents

1 Givens Plane Rotations

The 2×2 matrix R given by

$$R = \begin{pmatrix} c & s \\ -s & c \end{pmatrix}, \; c = \cos\theta, s = \sin\theta \tag{1.1}$$

is called a **plane rotation matrix** because geometrically the transformation

$$Rx = y, \quad \text{where } x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \tag{1.2}$$

represents a plane rotation through an angle θ, as shown in Figure 1.1. The matrix R has

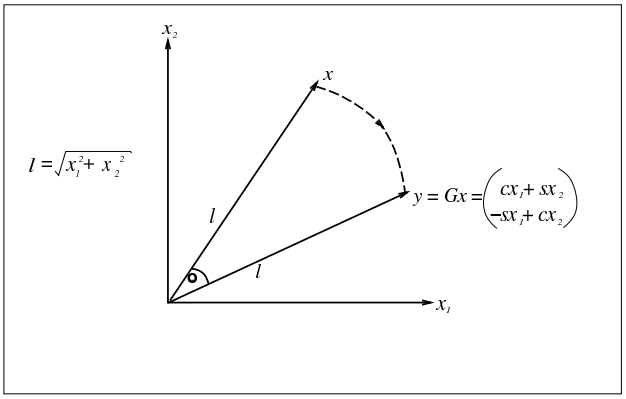

Figure 1.1: Plane Rotation

the important property that

$$R^T R = \begin{pmatrix} c & -s \\ s & c \end{pmatrix} \begin{pmatrix} c & s \\ -s & c \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I$$

so that R *is orthogonal*.

By choosing θ so that y lies along the x_1-axis we can make the x_2 component of y zero, as in Figure 1.2. When used in this way for the introduction of zeros the rotation is generally termed a **Givens plane rotation** (Givens, 1954). The usual choice of θ is that of Figure 1.2 given by

$$c = x_1/l, \; s = x_2/l, \quad \text{where } l = \sqrt{x_1^2 + x_2^2} \tag{1.3}$$

for which

$$y = Rx = \begin{pmatrix} l \\ 0 \end{pmatrix}, \tag{1.4}$$

but there are in fact two choices of θ which will make the second component of y zero, as illustrated in Figure 1.3. Thus the alternative choice of θ is such that

$$c = -x_1/l, \; s = -x_2/l, \quad \text{where } l = \sqrt{x_1^2 + x_2^2} \tag{1.5}$$

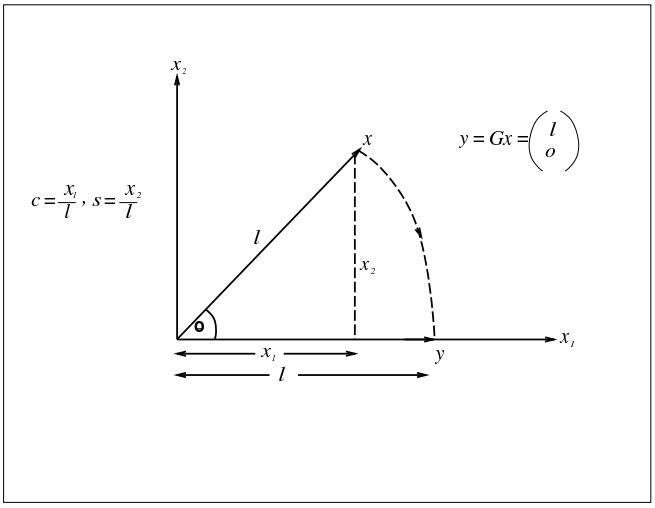

Figure 1.2: Givens Plane Rotation

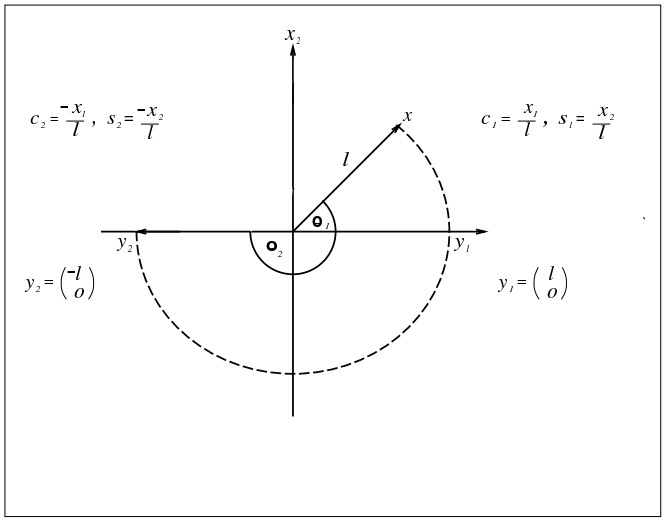

Figure 1.3: Choice of Rotations

for which

$$y = Rx = \begin{pmatrix} -l \\ 0 \end{pmatrix}. \tag{1.6}$$

Notice that we can always choose θ so that $c \geq 0$.

The two dimensional case is simply extended to n dimensions by embedding a 2×2 plane rotation matrix within the $n \times n$ unit matrix. The $n \times n$ plane rotation matrix, R_{ij}, which has the effect of rotating in the x_i, x_j plane, is the unit matrix except for the positions given by

$$r_{ii} = r_{jj} = c \quad \text{and} \quad r_{ij} = -r_{ji} = s, \; i < j. \tag{1.7}$$

As with the two dimensional case we have that

$$R_{ij}^T R_{ij} = I \tag{1.8}$$

so that R_{ij} *is an orthogonal matrix*. The transformation $R_{ij}A$ affects only rows i and j of A as is illustrated below.

$$R_{24}A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & c & 0 & s & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -s & 0 & c & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \text{x} & \text{x} & \text{x} & \text{x} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ \text{x} & \text{x} & \text{x} & \text{x} \\ a_{41} & a_{42} & a_{43} & a_{44} \\ \text{x} & \text{x} & \text{x} & \text{x} \\ \text{x} & \text{x} & \text{x} & \text{x} \end{pmatrix} = \begin{pmatrix} \text{x} & \text{x} & \text{x} & \text{x} \\ a'_{21} & a'_{22} & a'_{23} & a'_{24} \\ \text{x} & \text{x} & \text{x} & \text{x} \\ a'_{41} & a'_{42} & a'_{43} & a'_{44} \\ \text{x} & \text{x} & \text{x} & \text{x} \\ \text{x} & \text{x} & \text{x} & \text{x} \end{pmatrix}.$$

The positions indicated by x's are unaffected by the transformation. We can choose θ so that one of the elements $a'_{41}, a'_{42}, a'_{43}, a'_{44}$ is zero.

The essential part of an $n \times n$ Givens plane rotation

$$R_{ij}A = B \tag{1.9}$$

is given by

$$\begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} a_1 & a_2 & \dots & a_r \\ b_1 & b_2 & \dots & b_r \end{pmatrix} = \begin{pmatrix} a'_1 & a'_2 & \dots & a'_r \\ 0 & b'_2 & \dots & b'_r \end{pmatrix}, \tag{1.10}$$

where a_k and a'_k are elements of the ith rows of A and B respectively, b_k and b'_k are elements of the jth rows of A and B respectively and if we put

$$l = \sqrt{a_1^2 + b_1^2} \tag{1.11}$$

then we can choose θ so that

$$c = a_1/l \text{ and } s = b_1/l, \quad \text{or so that} \quad c = -a_1/l \text{ and } s = -b_1/l, \tag{1.12}$$

which gives

$$\begin{aligned} a'_i &= ca_i + sb_i, \quad && i = 1, 2, \dots, r, \\ b'_i &= -sa_i + cb_i, && i = 2, 3, \dots, r. \end{aligned} \tag{1.13}$$

Some care is necessary in computing c and s in order to avoid problems of overflow and underflow when a_1 and b_1 are very small or very large and l is computed as in equation (1.11).[1] One possibility, for which we shall have $c \geq 0$, is to compute c and s by using:

$$\begin{array}{llll} c = 1, & s = 0 & & \text{when } b_1 = 0 \\ c = 1/\sqrt{1+z^2}, & s = z/\sqrt{1+z^2}, & z = b_1/a_1 & \text{when } 0 < |b_1| \leq |a_1| \\ c = |z|/\sqrt{1+z^2}, & s = \text{sign}\,(z)\,/\sqrt{1+z^2}, & z = a_1/b_1 & \text{when } |a_1| < |b_1| \end{array} \tag{1.14}$$

A number of otherwise excellent algorithms have caused annoying failures through not taking this simple precaution.

[1] If a_1 and b_1 are small then a_1^2 and b_1^2 may be computed as zero.

We have so far mentioned only the real plane rotation. The extension to the complex case is straightforward (Wilkinson, 1965; Wilkinson, 1977) and here we content ourselves with just defining a plane rotation matrix in the complex case. The $n \times n$ complex plane rotation matrix, R_{ij}, is the unit matrix except for the positions given by

$$r_{ii} = \bar{r}_{jj} = e^{i\alpha} \cos\theta, \quad r_{ij} = -\bar{r}_{ij} = e^{i\beta} \sin\theta, \tag{1.15}$$

where $i = \sqrt{-1}$ and $\bar{a}$ denotes the complex conjugate of a. For this matrix

$$\bar{R}_{ij}^T R_{ij} = I$$

so that R_{ij} *is a unitary matrix.*

2 Givens Triangularization

An $m \times n$ matrix A can be transformed to upper triangular form by applying a sequence of Givens plane rotations. In particular we can introduce zeros in exactly the same order as with the reduction to upper triangular form by Gaussian elimination, we simply replace the elementary row transformation by a plane rotation. In the case $m > n$, A is transformed to the form

$$U = \begin{pmatrix} \tilde{U} \\ 0 \end{pmatrix}, \quad \tilde{U} \text{ an } n \times n \text{ upper triangular matrix,} \tag{2.1}$$

in n major steps and in the case $m \leq n$, A is transformed to the form

$$U = \begin{pmatrix} \tilde{U} & X \end{pmatrix}, \quad \tilde{U} \text{ an } m \times m \text{ upper triangular matrix,} \tag{2.2}$$

where X does not exist when $m = n$, in $(m-1)$ major steps. If we let R_{ij} denote the Givens plane rotation in the x_i, x_j plane designed to introduce a zero into the (i, j) position and we put

$$R_r = R_{mr} \ldots R_{r+2,r} R_{r+1,r}, \tag{2.3}$$

then the rth major step in transforming A to upper triangular form can be expressed as

$$A_r = R_r A_{r-1},\ A_0 = A,\ r = 1, 2, \ldots, k,\ k = \begin{cases} n, & m > n \\ m-1, & m \leq n. \end{cases} \tag{2.4}$$

This process is called **Givens triangularization** (Givens, 1958) and on completion we shall have

$$A_k = U. \tag{2.5}$$

If we also put

$$Q^T = R_k \ldots R_2 R_1 \tag{2.6}$$

then we have the factorization

$$A = QU, \tag{2.7}$$

where Q is orthogonal and U is of upper triangular form. The QU factorization of a matrix always exists.

The innermost loop of a Givens triangularization algorithm will be executed approximately $\frac{1}{6}n^2(3m-n)$ times when $m > n$ and approximately $\frac{1}{6}m^2(3n-m)$ times when $m \leq n$. Each execution will include four multiplications which is twice the number required by Householder triangularization (Householder, 1958), but unless multiplications dominate other operations this comparison is misleading. For instance in Algol implementations it is not unusual for Givens triangularization to be slightly faster than Householder triangularization.

3 Error Analysis for Givens Plane Rotations

One of the important features of orthogonal and near orthogonal matrices is that they provide us with numerically stable transformations (Wilkinson, 1965) and this is easily verified in the case of plane rotations.

Let $\ell_2(A)$ and $\ell_E(A)$ denote respectively the spectral and Euclidean norms of the $m \times n$ real matrix A so that

$$\ell_2(A) = \rho^{1/2}(A^T A) \quad \text{and} \quad \ell_E(A) = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^2 \right)^{1/2}, \tag{3.1}$$

where $\rho(A)$ denotes the spectral radius of A. Also let

$$\bar{a} = \text{fl}\,(a)$$

be the floating point value of a and assume that we use t-digit binary arithmetic such that for floating point numbers x and y we have

$$\text{fl}\,(x * y) = (x * y)(1 + \epsilon) = \frac{x * y}{1 + \gamma}, \ |\epsilon|, |\gamma| \leq 2^{-t}, \tag{3.2}$$

where $*$ represents any one of the operations $+, -, \times, \div$ and $\text{fl}\,(x * y)$ has neither underflowed nor overflowed.

We shall also consistently ignore second order and higher terms so that, for example, we shall assume that a bound of the form

$$(1 - 2^{-t})^r \leq 1 + \epsilon \leq (1 + 2^{-t})^r \tag{3.3}$$

can, without undue optimism, be replaced by

$$\left| \frac{\epsilon}{r} \right| \leq 2^{-t}. \tag{3.4}$$

If we compute c and s according to equation (1.14) then we need to compute the expression $(1 + \bar{z}^2)$, where $0 \leq \bar{z} \leq 1$ and $\bar{z}$ is the result of a single division so that

$$\bar{z} = \frac{z}{1 + \delta_1}, \ |\delta_1| \leq 2^{-t}.$$

This gives

$$\text{fl}\left(1 + z^2\right) = \frac{1}{1 + \delta_2} + \frac{\bar{z}^2}{1 + 2\delta_3} = \frac{1}{1 + \delta_2} + \frac{z^2}{1 + 3\delta_4}, \ |\delta_2|, |\delta_3|, |\delta_4| \leq 2^{-t}.$$

This can be expressed in the form

$$\text{fl}\left(1 + z^2\right) = \frac{1 + z^2}{1 + \delta_5}, \ \delta_5 = \frac{(1 + 3\delta_4)\delta_2 + 3(1 + \delta_2)\delta_4 z^2}{(1 + 3\delta_4) + (1 + \delta_2)z^2}$$

so that

$$|\delta_5| = \left| \frac{\delta_2 + 3\delta_4(\frac{1+\delta_2}{1+3\delta_4})z^2}{1 + (\frac{1+\delta_2}{1+3\delta_4})z^2} \right| \approx \left| \frac{\delta_2 + 3\delta_4 z^2}{1+z^2} \right| \leq \left(\frac{1+3z^2}{1+z^2} \right) 2^{-t} \leq 2 \cdot 2^{-t}.$$

Hence we can put

$$\text{fl}\left(1+z^2\right) = \frac{1+z^2}{1+2\delta_6},\ |\delta_6| \leq 2^{-t}. \tag{3.5}$$

Assuming that square roots satisfy $\text{fl}(a^{1/2}) = a^{1/2}(1+\epsilon), |\epsilon| \leq 2^{-t}$, this gives

$$\text{fl}\left(\frac{1}{\sqrt{1+z^2}}\right) = \frac{1+3\delta_7}{\sqrt{1+z^2}} \quad \text{and} \quad \text{fl}\left(\frac{z}{\sqrt{1+z^2}}\right) = \frac{z(1+3\delta_8)}{\sqrt{1+z^2}},\ |\delta_7|, |\delta_8| \leq 2^{-t}$$

and hence the computed cosine and sine satisfy

$$\bar{c} = c(1+3\alpha_1),\ |\alpha_1| \leq 2^{-t}, \quad \bar{s} = s(1+3\alpha_2),\ |\alpha_2| \leq 2^{-t}. \tag{3.6}$$

If $\bar{R}_{ij}$ denotes the computed rotation matrix R_{ij} and we put

$$H_{ij} = \bar{R}_{ij} - R_{ij} \tag{3.7}$$

then H_{ij} is the zero matrix except for the elements

$$h_{ii} = h_{jj} = \bar{c} - c = 3\alpha_1 c \quad \text{and} \quad h_{ij} = -h_{ji} = \bar{s} - s = 3\alpha_2 s. \tag{3.8}$$

Because $H_{ij}^T H_{ij}$ is diagonal with diagonal elements of zero and $(9\alpha_1^2 c^2 + 9\alpha_2^2 s^2)$ it follows that

$$\ell_2(H_{ij}) \leq 3 \cdot 2^{-t} = 3 \cdot 2^{-t} \ell_2(R_{ij}) \tag{3.9}$$

which shows that *the computed plane rotation matrix is very close to an orthogonal matrix.*

We now change notation from equation (1.9) slightly and instead put

$$B = \bar{R}_{ij} A \quad \text{so that} \quad \bar{B} = \text{fl}\left(\bar{R}_{ij} A\right). \tag{3.10}$$

Then, using similar notation to that of equation (1.10) and assuming that $\bar{b}_1'$ is set to zero, we have that

$$\begin{array}{ll} \bar{a}_i' = \text{fl}\left(\bar{c}a_i + \bar{s}b_i\right), & i = 1, 2, \ldots, r, \\ \bar{b}_1' = 0, \quad \bar{b}_i' = \text{fl}\left(-\bar{s}a_i + \bar{c}b_i\right), & i = 2, 3, \ldots, r. \end{array}$$

It follows straightforwardly that the $\bar{a}_i'$ and $\bar{b}_i'$ other than $\bar{b}_1'$ satisfy equations of the form

$$\left.\begin{array}{ll} \bar{a}_i' = \bar{a}_i' + 2\epsilon_i \bar{c} a_i + 2\beta_i \bar{s} b_i, & i = 1, 2, \ldots, r \\ \bar{b}_i' = \bar{b}_i' - 2\eta_i \bar{s} a_i + 2\theta_i \bar{c} b_i, & i = 2, 3, \ldots, r \end{array}\right\} |\epsilon_i|, |\beta_i|, |\eta_i|, |\theta_i| \leq 2^{-t}. \tag{3.11}$$

For $\bar{b}_1'$ we have that

$$\begin{aligned} \bar{b}_1' &= b_1' - (-\bar{s}a_1 + \bar{c}b_1') \qquad = b_1' + sa_1(1+3\alpha_2) - cb_1(1+3\alpha_1) \\ &= b_1' + 3\alpha_2 s a_1 - 3\alpha_1 c b_1 = b_1' + \frac{3\alpha_2 \bar{s} a_1}{1+3\alpha_2} - \frac{3\alpha_1 \bar{c} b_1}{1+3\alpha_1} \\ &= b_1' - 3\eta_1 \bar{s} a_1 + 3\theta_1 \bar{c} b_1, \quad |\eta_1|, |\theta_1| \leq 2^{-t}. \end{aligned} \tag{3.10a}$$

Equations (3.11) and (3.10a) can be expressed as

$$\begin{pmatrix} \bar{a}_1' - a_1' \\ \bar{b}_1' - b_1' \end{pmatrix} = \begin{pmatrix} 2\epsilon_1\bar{c} & 2\beta_1\bar{s} \\ 3\eta_1\bar{s} & 3\theta_1\bar{c} \end{pmatrix} \begin{pmatrix} a_1 \\ b_1 \end{pmatrix}, \begin{pmatrix} \bar{a}_i' - a_i' \\ \bar{b}_i' - b_i' \end{pmatrix} = \begin{pmatrix} 2\epsilon_i\bar{c} & 2\beta_i\bar{s} \\ -2\eta_i\bar{s} & 2\theta_i\bar{c} \end{pmatrix} \begin{pmatrix} a_i \\ b_i \end{pmatrix}, i > 1.$$

We can see that we certainly have

$$\ell_E(\bar{B} - B) \leq \sqrt{13} \cdot 2^{-t}\ell_E(A) < 4 \cdot 2^{-t}\ell_E(A). \tag{3.12}$$

If we define the error matrix E as

$$E = \bar{R}_{ij}^{-1}(\bar{B} - B) \tag{3.13}$$

then since

$$\ell_2(\bar{R}_{ij}^{-1}) \leq 1 + 3 \cdot 2^{-t} \tag{3.14}$$

we have that the computed matrix $\bar{B}$ satisfies the equation

$$\bar{B} = \bar{R}_{ij}(A + E), \quad \text{where } \ell_E(E) \leq 4 \cdot 2^{-t}\ell_E(A) \tag{3.15}$$

which confirms the stability of the plane rotation.

For some applications it is useful to express $\bar{B}$ as the result of an exactly orthogonal transformation. For this case we have that

$$\bar{B} = (R_{ij} + H_{ij})(A + E) = R_{ij}(A + E + R_{ij}^T H_{ij}(A + E))$$

and hence if we put

$$F = E + R_{ij}^T H_{ij} A + R_{ij}^T H_{ij} E$$

then

$$\bar{B} = R_{ij}(A + F), \quad \text{where } \ell_E(F) \leq 7 \cdot 2^{-t}\ell_E(A). \tag{3.16}$$

It is worth noting that plane rotations are independent of column scaling in the sense that, if D is a non-singular diagonal matrix and we put

$$\tilde{A} = AD, \tag{3.17}$$

then the computed matrices $\bar{B}$ and $\bar{R}_{ij}$ obtained by applying a Givens plane rotation to A are such that

$$\bar{B}D = \bar{R}_{ij}(\tilde{A} + \tilde{E}), \quad \text{where } \ell_E(\tilde{E}) \leq 4 \cdot 2^{-t}\ell_E(\tilde{A}). \tag{3.18}$$

Thus it is immaterial, from the stability point of view, whether we apply plane rotations to A or to $\tilde{A}$. This is most certainly not true of row scaling and A should always be reasonably row scaled before applying plane rotations.

For example, if we apply a Givens plane rotation, computed according to equation (1.14) with three significant figure decimal arithmetic, to the poorly scaled[2] equations $Ax = b$ given by

$$\begin{pmatrix} 1 & 2 \times 10^6 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4 \times 10^6 \\ 11 \end{pmatrix}$$

[2]A problem is poorly scaled if scaling will significantly improve the condition of that problem.

then we obtain the well scaled, but ill-conditioned upper triangular equations $Ux = c$ given by

$$\begin{pmatrix} 3.18 & 6.34 \times 10^5 \\ 0 & -1.9 \times 10^6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1.27 \times 10^6 \\ -3,81 \times 10^6 \end{pmatrix}$$

and backward substitution gives the computed solution

$$\bar{x}_1 = 0, \quad \bar{x}_2 = 2.01$$

in place of the correct three figure solution of $x_1 = 1.00,\ x_2 = 2.00$.

On the other hand if we first row equilibrate A with the scaling factor

$$D = \begin{pmatrix} 5 \times 10^{-7} & 0 \\ 0 & 0.143 \end{pmatrix}$$

to form the equations $\tilde{A}x = \tilde{b}$, where $\tilde{A} = DA$ and $\tilde{b} = Db$, given by

$$\begin{pmatrix} 5 \times 10^{-7} & 1 \\ 0.429 & 0.572 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 1.57 \end{pmatrix}$$

and now perform a Givens plane rotation we obtain the equations $\tilde{U}x = \tilde{c}$ given by

$$\begin{pmatrix} 0.429 & 0.572 \\ 0 & -1.00 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1.57 \\ -2.00 \end{pmatrix}$$

which gives the computed solution

$$\bar{x}_1 = 1.00, \quad \bar{x}_2 = 2.00.$$

In both cases we have performed stable transformations, it is just that the original equations are unnecessarily sensitive to small perturbations.

The error analysis is easily extended to a sequence of plane rotations (Wilkinson, 1965; Gentleman, 1975). If we perform the two plane rotations

$$B = R_{pq}R_{ij}A, \quad p \neq i, j \text{ and } q \neq i, j \tag{3.19}$$

then the two transformations are quite independent since R_{ij} affects only rows i and j of A and R_{pq} affects only rows p and q. These two plane rotations are said to be **disjoint**. Clearly disjoint plane rotation matrices commute, that is

$$R_{pq}R_{ij} = R_{ij}R_{pq},\ p \neq i, j \text{ and } q \neq i, j. \tag{3.20}$$

Because of the independent nature of disjoint plane rotations, if we put

$$\bar{B} = \text{fl}\left(\bar{R}_{pq}\bar{R}_{ij}A\right), \tag{3.21}$$

then we can immediately extend the result of equation (3.15) to give

$$\bar{B} = \bar{R}_{pq}\bar{R}_{ij}(A + E), \quad \text{where } \ell_E(E) \leq 4 \cdot 2^{-t}\ell_E(A). \tag{3.22}$$

The simplest error bounds for sequences of plane rotations come from expressing the sequence as groups of disjoint plane rotations. For example if we reduce the 6×5 matrix A to upper triangular form U, we perform the sequence of plane rotations

$$U = (R_{65})(R_{64}R_{54})(R_{63}R_{53}R_{43})(R_{62}R_{52}R_{42}R_{32})(R_{61}R_{51}R_{41}R_{31}R_{21})A.$$

Using the commutative property of disjoint plane rotations we can express this as

$$U = (R_{65})(R_{64})(R_{63}R_{54})(R_{62}R_{53})(R_{61}R_{52}R_{43})(R_{51}R_{42})(R_{41}R_{32})(R_{31})(R_{21})A,$$

where now the rotations in each bracket are disjoint.

Even in the presence of rounding errors both sequences will yield exactly the same upper triangular matrix.

In general we can replace equations (2.3) and (2.4) by the equations

$$A_r = R_r A_{r-1},\ A_0 = A,\ r = 1, 2, \ldots, k,\ k = \begin{cases} m+n-2, & m > n \\ 2n-3, & m \leq n, \end{cases} \tag{3.23}$$

where this time

$$R_r = \prod_{i+j=r+2} R_{ij}. \tag{3.24}$$

Here the matrix R_r is a product of disjoint plane rotation matrices. If we now put

$$\bar{R}_r = \prod_{i+j=r+2} \bar{R}_{ij} \quad \text{and} \quad A_r = \mathrm{fl}\,(R_r A_{r-1}) \tag{3.25}$$

then we have computationally that

$$A_r = \bar{R}_r(A_{r-1} + E_r), \quad \text{where } \ell_E(E_r) \leq 4 \cdot 2^{-t} \ell_E(A_{r-1}) \tag{3.26}$$

and putting

$$\bar{R} = \bar{R}_k \ldots \bar{R}_2 \bar{R}_1,\ E = E_1 + \bar{R}_1^{-1} E_2 + \cdots + \bar{R}_1^{-1} \bar{R}_2^{-1} \ldots \bar{R}_{k-1}^{-1} E_k,\ U = A_k \tag{3.27}$$

equation (3.26) leads directly to the result

$$U = \bar{R}(A + E), \quad \text{where } \ell_E(E) \leq 4k \cdot 2^{-t} \ell_E(A), \tag{3.28}$$

which confirms the remarkable stability of the Givens triangularization method. Note that the computed matrix U is exactly the same as the one we obtain computationally from equations (2.3) and (2.4).

Once again, by expressing $\bar{R}_r$ in the form $\bar{R}_r = R_r + H_r$, we can convert equation (3.28) to the form

$$QU = A + \tilde{E}, \quad \text{where } \ell_E(\tilde{E}) \leq 7k \cdot 2^{-t} \ell_E(A) \tag{3.29}$$

and Q is an exactly orthogonal matrix.

It should be appreciated that the actual bounds that we have obtained are not of any great importance since they are certainly rather pessimistic in practice. *The error analysis is important only in so far as it demonstrates the stability of plane rotations and indicates the reasons for that stability.*

4 Storage of Plane Rotations

In performing a QU factorization of a matrix A it is frequently desirable to overwrite details of the factorization on A without having to use a significant amount of additional storage space. If we wish to store details of the plane rotation R_{ij} in the location a_{ij} then we have only space for one of c or s, but we cannot use the obvious solution of storing c and recovering s as

$$s = \sqrt{1 - c^2}$$

because when c is close to unity the computed sine

$$\bar{s} = \mathrm{fl}\left(\sqrt{1 - \bar{c}^2}\right)$$

is likely to have a high relative error. There are various solutions to this storage problem. For instance (Stewart, 1976), if we always choose θ so that $c \geq 0$ and we let δ be a small real number such that $1.0/\delta$ does not overflow, then we can store the number t given by

$$t = \begin{cases} 0, & |s| < \delta \\ 1/s, & \delta \leq |s| < c \\ \mathrm{sign}\,(s)\,, & c = 0 \\ c \cdot \mathrm{sign}\,(s)\,, & c \leq |s|. \end{cases} \tag{4.1}$$

Corresponding to these four cases we shall have $|t| = 0, |t| > \sqrt{2}, |t| = 1$ and $0 < |t| \leq 1/\sqrt{2}$ respectively. We can then recover c and s by using

$$\begin{array}{llll} c = 1, & s = 0 & & \text{when } |t| = 0 \\ s = 1/t, & c = \sqrt{1 - s^2} & & \text{when } |t| > 1 \\ c = 0, & s = t & & \text{when } |t| = 1 \\ c = |t|, & s = \mathrm{sign}\,(t)\sqrt{1 - c^2} & & \text{when } 0 < |t| < 1. \end{array} \tag{4.2}$$

Another possibility is to alter slightly the way in which we compute c and s. If we let ω be a large machine representable real number and δ a small real number such that $1.0/\delta < \omega$ then we compute c and s and store the number z given as follows.

$$\begin{array}{ll} z = \omega, c = 1, s = 0 & \text{when } |b_1| \leq \delta |a_1| \\ z = \frac{a_1}{b_1} \left\{ \begin{array}{llll} c = \frac{1}{\sqrt{1+(1/z)^2}}, & s = \frac{(1/z)}{\sqrt{1+(1/z)^2}} & \text{when } |z| > 1 \\ c = \frac{|z|}{\sqrt{1+z^2}}, & s = \frac{\mathrm{sign}(z)}{\sqrt{1+z^2}} & \text{when } |z| \leq 1 \end{array} \right\} & \text{when } |b_1| > \delta |a_1|. \end{array} \tag{4.3}$$

We can then recover c and s by using

$$\begin{array}{llll} c = 1, & s = 0 & & \text{when } z = \omega \\ c = \frac{1}{\sqrt{1+(1/z)^2}}, & s = \frac{(1/z)}{\sqrt{1+(1/z)^2}} & & \text{when } 1 < |z| < \omega \\ c = \frac{|z|}{\sqrt{1+z^2}}, & s = \frac{\mathrm{sign}(z)}{\sqrt{1+z^2}} & & \text{when } |z| \leq 1. \end{array} \tag{4.4}$$

With this scheme we can recover exactly the same c and s from equation (4.4) as we computed in equation (4.3).

5 Modified Givens Plane Rotations

We can obtain computational savings with plane rotations by aiming for a QDU factorization, where D is a non-singular diagonal matrix, in place of the standard QU factorization.

If we factorize the matrices A and B given by

$$A = \begin{pmatrix} a_1 & a_2 & \dots & a_r \\ b_1 & b_2 & \dots & b_r \end{pmatrix}, \quad B = \begin{pmatrix} a'_1 & a'_2 & \dots & a'_r \\ 0 & b'_2 & \dots & b'_r \end{pmatrix}, \tag{5.1}$$

as

$$A = D\tilde{A}, \quad B = K\tilde{B}, \tag{5.2}$$

where

$$D = \begin{pmatrix} d_1 & 0 \\ 0 & d_2 \end{pmatrix}, \quad \tilde{A} = \begin{pmatrix} \tilde{a}_1 & \tilde{a}_2 & \dots & \tilde{a}_r \\ \tilde{b}_1 & \tilde{b}_2 & \dots & \tilde{b}_r \end{pmatrix},$$
$$K = \begin{pmatrix} d'_1 & 0 \\ 0 & d'_2 \end{pmatrix}, \quad \tilde{B} = \begin{pmatrix} \tilde{a}'_1 & \tilde{a}'_2 & \dots & \tilde{a}'_r \\ 0 & \tilde{b}'_2 & \dots & \tilde{b}'_r \end{pmatrix} \tag{5.3}$$

then with the modified Givens plane rotation, instead of working with A and B explicitly, we work with their factors $D\tilde{A}$ and $K\tilde{B}$. Equations (1.13) and (5.2) give that

$$\begin{aligned} \tilde{a}'_i &= (cd_1\tilde{a}_i + sd_2\tilde{b}_i)/d'_1, \quad i = 1, 2, \dots, r \\ \tilde{b}'_i &= (-sd_1\tilde{a}_i + cd_2\tilde{b}_i)/d'_2, \quad i = 2, 3, \dots, r. \end{aligned} \tag{5.4}$$

Since A is given, we must regard D and $\tilde{A}$ as given so that we have choice just of the matrix K. various choices of d'_1 and d'_2 lead to computational savings over the standard Givens plane rotation, (Hammarling, 1974) so that modified plane rotations are sometimes termed fast plane rotations.

Two examples are the following two multiplication formulae. Firstly the choice

$$d'_1 = cd_1, \quad d'_2 = cd_2 \tag{5.4a}$$

gives

$$\left.\begin{aligned} \tilde{a}'_i &= \tilde{a}_i + (\tfrac{1}{z} \cdot \tfrac{d_2}{d_1})\tilde{b}_i, \quad i = 1, 2, \dots, r \\ \tilde{b}'_i &= \tilde{b}_i - (\tfrac{1}{z} \cdot \tfrac{d_1}{d_2})\tilde{a}_i, \quad i = 2, 3, \dots, r \end{aligned}\right\} \quad \text{where } z = \cot\theta. \tag{5.5a}$$

Secondly the choice

$$d'_1 = sd_2, \quad d'_2 = -sd_1 \tag{5.4b}$$

gives

$$\left.\begin{aligned} \tilde{a}'_i &= \tilde{b}_i + (z \cdot \tfrac{d_1}{d_2})\tilde{a}_i, \quad i = 1, 2, \dots, r \\ \tilde{b}'_i &= \tilde{a}_i - (z \cdot \tfrac{d_2}{d_1})\tilde{b}_i, \quad i = 2, 3, \dots, r \end{aligned}\right\} \quad \text{where } z = \cot\theta. \tag{5.5b}$$

If we are prepared to work with the matrices D^2 and K^2 in place of D and K when we can also avoid the square root required to compute c and s, which has lead to the term square root free plane rotations being used for the modified plane rotations. Putting

$$k_1 = d_1^2, \; k_2 = d_2^2, \quad k'_1 = (d'_1)^2, \; k'_2 = (d'_2)^2 \tag{5.7}$$

then for equations (5.4*a*a) and (5.5*a*a) we have that

$$c^2 = \frac{k_1\tilde{a}_1^2}{k_1\tilde{a}_1^2 + k_2\tilde{b}_1^2}, \quad k_1' = c^2 k_1, \quad k_2' = c^2 k_2 \tag{5.7a}$$

$$\left.\begin{array}{ll} \tilde{a}_i' = \tilde{a}_i + \left(\frac{1}{\tilde{z}} \cdot \frac{k_2}{k_1}\right)\tilde{b}_i, & i = 1, 2, \ldots, r \\ \tilde{b}_i' = \tilde{b}_i - \left(\frac{1}{\tilde{z}}\right)\tilde{a}_i, & i = 2, 3, \ldots, r \end{array}\right\} \text{ where } \tilde{z} = \frac{\tilde{a}_1}{\tilde{b}_1}. \tag{5.8a}$$

From equations (5.4*b*b) and (5.5*b*b) we have that

$$s^2 = \frac{k_2\tilde{b}_1^2}{k_1\tilde{a}_1^2 + k_2\tilde{b}_1^2}, \quad k_1' = s^2 k_2, \quad k_2' = s^2 k_1 \tag{5.7b}$$

$$\left.\begin{array}{ll} \tilde{a}_i' = \tilde{b}_i + \left(\tilde{z} \cdot \frac{k_1}{k_2}\right)\tilde{a}_i, & i = 1, 2, \ldots, r \\ \tilde{b}_i' = \tilde{a}_i - (\tilde{z})\,\tilde{b}_i, & i = 2, 3, \ldots, r \end{array}\right\} \text{ where } \tilde{z} = \frac{\tilde{a}_1}{\tilde{b}_1}. \tag{5.8b}$$

Notice that in working with K^2 we cannot tell the correct signs of d_1' and d_2', but this is not important because we can always assume that we have used the plane rotation that gives positive values for d_1' and d_2'.

We clearly cannot tolerate a singular diagonal factor arising, but, particularly when a sequence of plane rotations is involved, there is considerable danger of underflow in computing k_1' and k_2' from equations (5.7*a*a) and (5.7*b*b). We can dramatically lessen the danger by using a combination of the (a) and (b) equations (Wilkinson, 1977). Since we must have either $c^2 \geq \frac{1}{2}$ or $s^2 \geq \frac{1}{2}$ we can use the (a) equations when $c^2 \geq \frac{1}{2}$ and the (b) equations when $c^2 < \frac{1}{2}$. We must still be prepared to normalize occasionally if a large sequence of rotations is involved with one row.

As with the standard plane rotation it is important also to be aware of the danger of underflow and overflow in the computation of $k_1\tilde{a}_1^2$ $(= a_1^2)$ and $k_2\tilde{b}_1^2$ $(= b_1^2)$. One way round this danger (Cox, personal communication) is to use the computing scheme, comparable to that of equation (4.3) for the standard plane rotation, given by

$$\begin{array}{ll} c^2 = 1,\ s^2 = 0 & \text{when } |\tilde{b}_1| \leq \delta|\tilde{a}_1| \\ \tilde{z} = \frac{\tilde{a}_1}{\tilde{b}_1} \left\{\begin{array}{l} c^2 = \frac{1}{1+w},\ s^2 = c^2 w,\ w = \frac{1}{\tilde{z}}\left(\frac{k_2}{k_1} \cdot \frac{1}{\tilde{z}}\right) \text{ when } |\tilde{z}| > 1 \\ s^2 = \frac{1}{1+w},\ c^2 = s^2 w,\ w = \tilde{z}\left(\frac{k_1}{k_2} \cdot \tilde{z}\right) \text{ when } |\tilde{z}| \leq 1 \end{array}\right\} & \text{when } |\tilde{b}_1| > \delta|\tilde{a}_1|. \end{array} \tag{5.10}$$

If we also put $\tilde{z} = \omega$ when $|\tilde{b}_1| \leq \delta|\tilde{a}_1|$ then storage of $\tilde{z}$ will allow us to recover details of the modified plane rotation.

Stability of the modified pane rotation is comparable to that of the standard plane rotation. If we define the computed matrix B as $\bar{B}$ given by

$$\bar{B} = \text{fl}^{1/2}(K^2)\text{fl}\left(\tilde{B}\right) \tag{5.11}$$

then corresponding to the equation (3.15), we can show that

$$\bar{B} = R_{ij}(A + F), \quad \text{where } \ell_E(F) \leq 14 \cdot 2^{-t}\ell_E(A) \tag{5.12}$$

and the factor 14 is probably rather generous.

Two other choices of K that are of some interest (Gentleman, 1973), particularly in applications such as LDL^T updates (Gill, Golub, Murray and Saunders, 1974; Fletcher and Powell, 1974; Gill and Murray, 1977) are given by

$$d_1' = 1, \quad d_2' = cd_2, \tag{5.12a}$$

where l is as given by equation (1.11) and by

$$d_1' = 1, \quad d_2' = -sd_1. \tag{5.12b}$$

For both of these choices we find that

$$\tilde{a}_1' = 1 \tag{5.14}$$

and so these modifications lead to QDU factorizations where U is of unit upper triangular form. For these two choices we can develop three multiplication square root free formulae similar to the formulae given above for choices (5.4aa and 5.4bb).

The rth step of modified Givens triangularization can be expressed as

$$D_r\tilde{A}_r = R_rD_{r-1}\tilde{A}_{r-1},\ D_0\tilde{A}_0 = A,\ r = 1, 2, \ldots, k,\ k = \begin{cases} n, & m > n \\ m-1, & m \leq n, \end{cases} \tag{5.15}$$

where R_r is given by equation (2.3), D_r is diagonal and an appropriate modification is used to implement each plane rotation. On completion of the triangularization we shall have factorized A as

$$A = QDU, \tag{5.16}$$

where

$$Q^T = R_k \ldots R_2R_1,\ D = D_k \quad \text{and} \quad U = \tilde{A}_k. \tag{5.17}$$

Whenever possible an appropriate choice is to take D_0 so that A is approximately row equilibrated.

Considerably more computational experience in a variety of applications is required before proper judgment on modified plane rotations can be given. In the QDU factorization of a large full matrix the avoidance of square roots is unimportant since these are not computed within the innermost loop of the algorithm, and the saving due to halving the number of multiplications will depend upon the time taken for a multiplication compared with the time taken for other operations such as array referencing, typically we might expect a saving of 10% or 15% over the QU factorization obtained by standard Givens plane rotations. On the other hand the code for the QDU factorization will certainly be longer than that for the QU factorization.

It seems possible that the modified plane rotations will be at their best in "semi-sparse" situations such as the factorization of an upper-Hessenberg matrix where each row is only involved in two plane rotations, but has an average of $\frac{1}{2}n$ non-zero elements.

6 Miscellaneous Givens Plane Rotations

Even with the standard Givens plane rotation it is possible to re-arrange the computation to give three multiplication formulae. Such version are only worthwhile if multiplications take longer than additions and element references. For example (Gill et al., 1974), if we put

$$\mu = \frac{-s}{c+1} \tag{6.1}$$

then we can replace equation (1.13) by

$$\begin{array}{ll} a'_i = ca_i + sb_i, & i = 1, 2, \ldots, r, \\ b'_i = \mu(a'_i + a_i) + b_i, & i = 2, 3, \ldots, r. \end{array} \tag{6.2}$$

Notice that μ satisfies

$$|\mu| \leq 1 \tag{6.3}$$

and if we always choose the rotation for which $c \geq 0$ then we can always compute μ with a low relative error, which implies that this version of the Given plane rotation is stable.

Another possibility is to compute z and y given by

$$z = \frac{a_1 + b_1}{l}, \quad y = \frac{a_1 - b_1}{l}, \quad \text{where } l = \sqrt{a_1^2 + b_1^2} \tag{6.4}$$

and then we can replace equation (1.13) by

$$\left.\begin{array}{ll} a'_i = a_i z + \gamma_i, & i = 1, 2, \ldots, r, \\ b'_i = b_i y + \gamma_i, & i = 2, 3, \ldots, r. \end{array}\right\} \text{ where } \gamma_i = s(b_i - a_i). \tag{6.5}$$

Notice that $z = c + s$ and $y = c - s$, but to compute z and y with low relative error they should be found as in equation (6.4) and in this case this version of the plane rotation is once again stable.

A transformation matrix which may be of occasional use is the matrix given by

$$P = \begin{pmatrix} c & s \\ -s & c \end{pmatrix}, \quad \text{where } |c| + |s| = 1. \tag{6.6}$$

Gwen Peters has termed this the **Poorman's plane rotation matrix**. $P^T P$ is diagonal, but is only the unit matrix if $c = \pm 1$ or $c = 0$ so that P is generally *not* orthogonal.

We can introduce zeros, as in equation (1.10), in just the same way as with a plane rotation matrix. We choose c and s as in equation (1.12) except that here we must define l as

$$l = |a_1| + |b_1|. \tag{6.7}$$

That is, if we choose c and s as

$$c = a_1/l \text{ and } s = b_1/l, \quad \text{or} \quad c = -a_1/l \text{ and } s = -b_1/l, \tag{6.8}$$

then

$$\begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} a_1 & a_2 & \ldots & a_r \\ b_1 & b_2 & \ldots & b_r \end{pmatrix} = \begin{pmatrix} a'_1 & a'_2 & \ldots & a'_r \\ 0 & b'_2 & \ldots & b'_r \end{pmatrix}, \tag{6.9}$$

where

$$\begin{aligned} a_i' &= ca_i + sb_i, \quad i = 1, 2, \ldots, r, \\ b_i' &= -sa_i + cb_i, \quad i = 2, 3, \ldots, r. \end{aligned} \tag{6.10}$$

The advantages of the Poorman's plane rotation are that no special precautions are necessary in computing c and s since we cannot underflow in computing l from equation (6.7) and no square root is required. As with the standard plane rotation we can modify the Poorman's plane rotation in order to obtain two multiplication versions and again these do not suffer quite the same underflow problems so that code tends to be shorter. Of course loss of orthogonality will frequently be unacceptable.

Stability of an individual Poorman's plane rotation is comparable to that of the standard plane rotation. Corresponding to equation (3.15), here we find that

$$\bar{B} = \bar{P}_{ij}(A + E), \quad \text{where } \ell_E(E) \leq 3 \cdot 2^{-t} \ell_E(A) \tag{6.11}$$

P_{ij} being a Poorman's plane rotation in the (i, j) plane. For a sequence of Poorman's plane rotations we have the possibility of a build up of errors, rather like that of Gaussian elimination with partial pivoting. It is interesting to note that this build up of errors is associated with decay in the elements of the transformed matrices, in direct contrast to Gaussian elimination where the build up of errors is associated with growth. In the case of Poorman's plane rotations the growth occurs in the inverses of the transformed matrices.

The point at which the error analysis for a sequence of Poorman's plane rotations departs from the error analysis for a sequence of standard plane rotations comes when we take the norm of the error matrix E of equation (3.27). For the standard plane rotation we have that

$$\ell_2(\bar{R}_i^{-1}) = 1 + 3 \cdot 2^{-t}$$

so that, ignoring second order terms, we have that

$$\ell_E(E) = \sum_{i=1}^{k} \ell_E(E_i).$$

For the Poorman's plane rotation the corresponding error matrix is

$$E = E_1 + \bar{P}_1^{-1} E_2 + \cdots + \bar{P}_1^{-1} \bar{P}_2^{-1} \ldots \bar{P}_{k-1}^{-1} E_k, \quad \text{where } \bar{P}_r = \prod_{i+j=r+2} \bar{P}_{ij}, \tag{6.12}$$

but unfortunately the best we can say is that

$$\ell_2(\bar{P}_i^{-1}) \leq \sqrt{2} + 2 \cdot 2^{-t} \tag{6.13}$$

so that

$$\ell_E(E) \leq \sum_{i=1}^{k} \left(\sqrt{2}\right)^{i-1} \ell_E(E_i)$$

and hence corresponding to equation (3.28) we have that

$$U = \bar{P}(A + E), \quad \text{where } \ell_E(E) \leq 3 \left(\frac{(\sqrt{2})^k - 1}{\sqrt{2} - 1} \right) 2^{-t} \ell_E(A), \tag{6.14}$$

As with Gaussian elimination and partial pivoting such growth is extremely unlikely in practice and it is not completely clear as to whether or not such growth is even possible.

7 Updating Matrix Factorizations and Products of Plane Rotations

Plane rotations are of particular importance in applications where a modification such as a rank-one change is made to a matrix and it is required to update the factorization of that matrix (Gill and Murray, 1977). Plane rotations are preferable to Householder transformations in most sparse situations since plane rotations generally lead to significantly less fill-in than do Householder transformations. Such a case, which could arise for example in replacing a column of a matrix, is illustrated in Figure 7.1. One Householder transformation will introduce the column of zeros, but complete fill-in occurs. If we use the sequence of plane rotations

$$B = R_{21}R_{32}R_{43}R_{54}R_{65}R_{76}A, \tag{7.1}$$

where $R_{i,i-1}$ introduces a zero into the $(i,1)$ position so that we introduce the zeros from the final element upwards, then we only fill-in the indicated sub-diagonal.

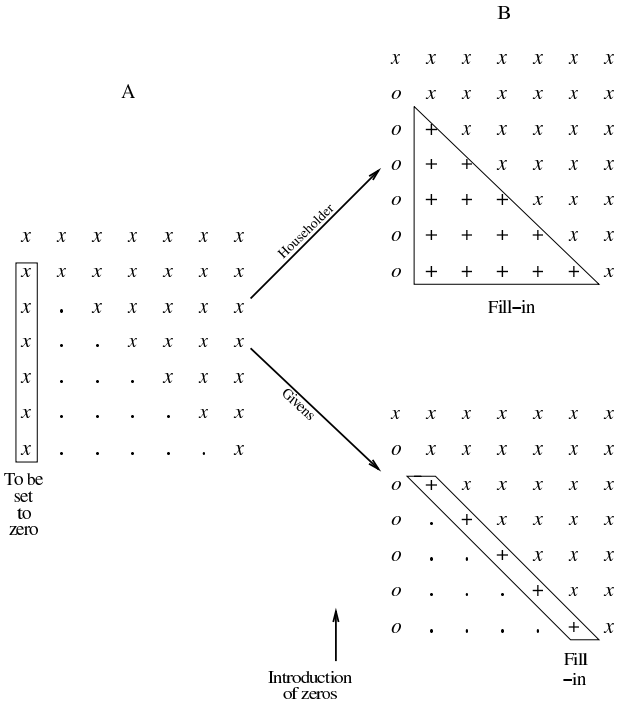

Figure 7.1: Removing a Spike

Products of plane rotations of the type illustrated in Figure 7.1 arise frequently in such applications and although something of the structure of products of plane rotations was known previously (Wilkinson, 1965), it is in the study of updating matrix factorizations that a fuller understanding of products of plane rotations has been realised (Gill et al., 1974).

To illustrate the type of matrix that arises consider the product of equation (7.1) given by

$$R = R_{21}R_{32}R_{43}R_{54}R_{65}R_{76}. \tag{7.2}$$

If we let c_i and s_i denote the sine and cosine that define $R_{i+1,i}$ then we find that R is the matrix

$$R = \begin{pmatrix} c_1 & s_1c_2 & s_1s_2c_3 & s_1s_2s_3c_4 & s_1s_2s_3s_4c_5 & s_1s_2s_3s_4s_5c_6 & s_1s_2s_3s_4s_5s_6 \\ -s_1 & c_1c_2 & c_1s_2c_3 & c_1s_2s_3c_4 & c_1s_2s_3s_4c_5 & c_1s_2s_3s_4s_5c_6 & c_1s_2s_3s_4s_5s_6 \\ 0 & -s_2 & c_2c_3 & c_2s_3c_4 & c_2s_3s_4c_5 & c_2s_3s_4s_5c_6 & c_2s_3s_4s_5s_6 \\ 0 & 0 & -s_3 & c_3c_4 & c_3s_4c_5 & c_3s_4s_5c_6 & c_3s_4s_5s_6 \\ 0 & 0 & 0 & -s_4 & c_4c_5 & c_4s_5c_6 & c_4s_5s_6 \\ 0 & 0 & 0 & 0 & -s_5 & c_5c_6 & c_5s_6 \\ 0 & 0 & 0 & 0 & 0 & -s_6 & c_6 \end{pmatrix} \tag{7.3}$$

so that R is an upper Hessenberg matrix. An important property of R is that it can be expressed in the special form

$$R = \begin{pmatrix} \beta_1\alpha_1 & \beta_1\alpha_2 & \beta_1\alpha_3 & \beta_1\alpha_4 & \beta_1\alpha_5 & \beta_1\alpha_6 & \beta_1\alpha_7 \\ -s_1 & \beta_2\alpha_2 & \beta_2\alpha_3 & \beta_2\alpha_4 & \beta_2\alpha_5 & \beta_2\alpha_6 & \beta_2\alpha_7 \\ 0 & -s_2 & \beta_3\alpha_3 & \beta_3\alpha_4 & \beta_3\alpha_5 & \beta_3\alpha_6 & \beta_3\alpha_7 \\ 0 & 0 & -s_3 & \beta_4\alpha_4 & \beta_4\alpha_5 & \beta_4\alpha_6 & \beta_4\alpha_7 \\ 0 & 0 & 0 & -s_4 & \beta_5\alpha_5 & \beta_5\alpha_6 & \beta_5\alpha_7 \\ 0 & 0 & 0 & 0 & -s_5 & \beta_6\alpha_6 & \beta_6\alpha_7 \\ 0 & 0 & 0 & 0 & 0 & -s_6 & \beta_7\alpha_7 \end{pmatrix} \tag{7.4}$$

where there are various vectors α and β that can be generated to give R. For instance we can use the forward recurrence given by

$$\begin{array}{llll} \alpha_1 = c_1, & \beta_1 = 1, & \eta_1 = s_1, & c_7 = 1, s_7 = 0 \\ \alpha_i = c_i\eta_{i-1}, & \beta_i = c_{i-1}/\eta_{i-1}, & \eta_i = s_i\eta_{i-1}, & i = 2, 3, \ldots, 7, \end{array} \tag{7.5}$$

or we can use the backward recurrence given by

$$\begin{array}{llll} \alpha_7 = 1, & \beta_7 = c_6, & \eta_7 = s_6, & c_0 = 1, s_0 = 0 \\ \alpha_i = c_i/\eta_{i+1}, & \beta_i = c_{i-1}\eta_{i+1}, & \eta_i = s_{i-1}\eta_{i+1}, & i = 6, 5, \ldots, 1, \end{array} \tag{7.6}$$

It should be noted that in the application illustrated in Figure 7.1 we can assume that $a_{71} \neq 0$, because if a_{71} were zero we could omit R_{76} and hence we can assume that $s_6 \neq 0$ from which it follows that $s_i \neq 0, i = 6, 5, \ldots, 1$ so that both the above recurrences are defined, although computationally we may meet problems of overflow if any of the s_i are small.

The matrix R is an example of a type of matrix called a **special matrix** (Gill et al., 1974) in this case a special upper Hessenberg matrix. Notice that if R is an $n \times n$ special upper Hessenberg matrix of the form illustrated by equation (7.4) and X is an n element vector, then

$$Rx = \begin{pmatrix} \beta_1 \sum_{i=1}^{n} \alpha_i x_i \\ -s_1x_1 + \beta_2 \sum_{i=2}^{n} \alpha_i x_i \\ -s_2x_2 + \beta_3 \sum_{i=3}^{n} \alpha_i x_i \\ \vdots \\ -s_{n-1}x_{n-1} + \beta_n \sum_{i=n}^{n} \alpha_i x_i \end{pmatrix} \tag{7.7}$$

so that the vector Rx can be generated in $3n$ multiplications in place of $4n$ multiplications for the separate plane rotations.

We can extract diagonal factors when using these special matrices just as for single plane rotations and it was in this context that modified plane rotations first appeared. (They are implicitly used in the method for modifying the LDL^T factors of a matrix described in Gill and Murray (1970) as can be seen rather more clearly in Gill et al. (1974).)

8 Similarity Transformations and Jacobi Rotations

We have so far just looked at plane rotations of the form

$$B = R_{ij}A, \tag{8.1}$$

but plane rotations are also frequently used for the introduction of zeros in similarity transformations, in which case we are interested in the transformation

$$C = R_{ij}AR_{ij}^T. \tag{8.2}$$

In many applications we choose R_{ij} to be a Givens plane rotation so that we introduce a zero into B just as before and whether or not that zero is preserved in completing the similarity transformation depends upon the position of that zero. Two examples are illustrated below.

a)

$$R_{42}AR_{42}^T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c & 0 & s \\ 0 & 0 & 1 & 0 \\ 0 & -s & 0 & c \end{pmatrix} \begin{pmatrix} \mathrm{x} & \mathrm{x} & \mathrm{x} & \mathrm{x} \\ \mathrm{x} & \mathrm{x} & \mathrm{x} & \mathrm{x} \\ \mathrm{x} & \mathrm{x} & \mathrm{x} & \mathrm{x} \\ \mathrm{x} & \mathrm{x} & \mathrm{x} & \mathrm{x} \end{pmatrix} R_{42}^T = \begin{pmatrix} \mathrm{x} & \mathrm{x} & \mathrm{x} & \mathrm{x} \\ \rightarrow & \rightarrow & \rightarrow & \rightarrow \\ \mathrm{x} & \mathrm{x} & \mathrm{x} & \mathrm{x} \\ 0 & \rightarrow & \rightarrow & \rightarrow \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c & 0 & -s \\ 0 & 0 & 1 & 0 \\ 0 & s & 0 & c \end{pmatrix}$$

$$= \begin{pmatrix} \mathrm{x} & \downarrow & \mathrm{x} & \downarrow \\ \rightarrow & \downarrow\!\!\rightarrow & \rightarrow & \downarrow\!\!\rightarrow \\ \mathrm{x} & \downarrow & \mathrm{x} & \downarrow \\ 0 & \downarrow\!\!\rightarrow & \rightarrow & \downarrow\!\!\rightarrow \end{pmatrix} = C. \qquad \begin{array}{l} c \text{ and } s \text{ chosen to make } b_{41} \text{ zero} \\ \rightarrow \text{ modified by } R_{42} \\ \downarrow \quad \text{modified by } R_{42}^T \end{array}$$

Notice that if A is symmetric then c_{14} will also be zero.

b)

$$R_{42}AR_{42}^T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c & 0 & s \\ 0 & 0 & 1 & 0 \\ 0 & -s & 0 & c \end{pmatrix} \begin{pmatrix} \mathrm{x} & \mathrm{x} & \mathrm{x} & \mathrm{x} \\ \mathrm{x} & \mathrm{x} & \mathrm{x} & \mathrm{x} \\ \mathrm{x} & \mathrm{x} & \mathrm{x} & \mathrm{x} \\ \mathrm{x} & \mathrm{x} & \mathrm{x} & \mathrm{x} \end{pmatrix} R_{42}^T = \begin{pmatrix} \mathrm{x} & \mathrm{x} & \mathrm{x} & \mathrm{x} \\ \rightarrow & \rightarrow & \rightarrow & \rightarrow \\ \mathrm{x} & \mathrm{x} & \mathrm{x} & \mathrm{x} \\ \rightarrow & 0 & \rightarrow & \rightarrow \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c & 0 & -s \\ 0 & 0 & 1 & 0 \\ 0 & s & 0 & c \end{pmatrix}$$

$$= \begin{pmatrix} \mathrm{x} & \downarrow & \mathrm{x} & \downarrow \\ \rightarrow & \downarrow\!\!\rightarrow & \rightarrow & \downarrow\!\!\rightarrow \\ \mathrm{x} & \downarrow & \mathrm{x} & \downarrow \\ \rightarrow & \downarrow\!\!\rightarrow & \rightarrow & \downarrow\!\!\rightarrow \end{pmatrix} = C. \quad c \text{ and } s \text{ chosen to make } b_{42} \text{ zero}$$

$$\uparrow$$

zero lost

The first type of Givens similarity transformation is used in applications such as the reduction of a matrix to upper Hessenberg form, or, in the case of a symmetric matrix, to tridiagonal form (Givens, 1954; Wilkinson, 1965).

If here we let R_{ij} denote the Givens plane rotation in the x_i, x_j plane that is designed to introduce a zero into the $(i, j-1)$ position and we put

$$R_r = R_{n,r+1} \ldots R_{r+3,r+1}R_{r+2,r+1}, \tag{8.3}$$

then the rth major step in transforming the $n \times n$ matrix A to Hessenberg form can be expressed as

$$A_r = R_r A_{r-1} R_r^T, \quad A_0 = A, \quad r = 1, 2, \ldots, n-2 \tag{8.4}$$

and on completion A_{n-2} will be upper Hessenberg. If A is symmetric then each A_r will also be symmetric and hence A_{n-2} will be tridiagonal. We can obviously take advantage of symmetry during the computation.

The second type of Givens similarity transformation is used in iterative processes such as the QR algorithm (Francis, 1961; Francis, 1962; Kublanovskaya, 1961; Wilkinson, 1965), and in a variant of the QR algorithm used to find the singular value decomposition (Golub and Kahan, 1968; Golub and Reinsch, 1970). The aim of the QR algorithm is to reduce a matrix to upper triangular form by means of similarity transformations, this being diagonal form if the matrix is also symmetric and at the heart of the algorithm is an iterative step of the form

$$B_i = Q_i^T B_{i-1} Q_i, \tag{8.5}$$

where Q_i is chosen, as in equations (2.3) - (2.7), so that $Q_i^T B_{i-1}$ is of upper triangular form. This upper triangular form is of course lost in completing the similarity transformation. If B_{i-1} is an upper Hessenberg matrix then B_i will also be upper Hessenberg and so if we wish to upper triangularize an $n \times n$ matrix A it is usual to first reduce it to upper Hessenberg form. Various refinements such as shifts or origin are necessary to ensure success of the method.

The modifications of Section 5 are easily extended to these similarity transformations. In the modified versions we factorize the matrices A and C of equation (8.2) as

$$A = D_1 \tilde{A} D_2, \quad C = K_1 \tilde{C} K_2 \tag{8.6}$$

where D_1, D_2, K_1 and K_2 are all non-singular diagonal matrices and apply the plane rotations as described in Section 5. If A is symmetric then we naturally choose $D_1 = D_2$ and assuming that we use the same modification for both the left-hand and right-hand plane rotations, this makes $K_1 = K_2$.

Plane rotations can also be used to introduce zeros into positions that are affected by both the left-hand and right-hand transformations of a similarity transformation, such a transformation generally being applied to symmetric matrices. This naturally requires a different choice of c and s that used in the Givens plane rotation.

The part of the similarity transformation of a *symmetric* matrix that is affected by both the left and right hand transformation matrix is given by

$$\begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} a & b \\ b & d \end{pmatrix} \begin{pmatrix} c & -s \\ s & c \end{pmatrix} = \begin{pmatrix} \alpha & \beta \\ \beta & \delta \end{pmatrix} \tag{8.7}$$

where

$$\alpha = c^2 a + 2csb + s^2 d, \quad \beta = (c^2 - s^2)b + cs(d - a), \quad \delta = c^2 d - 2csb + s^2 a. \tag{8.8}$$

We can make $\beta = 0$ by choosing θ so that

$$(c^2 - s^2)b = cs(a - d) \tag{8.9}$$

and if $b \neq 0$ this gives

$$\cot 2\theta = \frac{a-d}{2b}, \quad b \neq 0 \tag{8.10}$$

Putting

$$\gamma = \cot 2\theta \quad \text{and} \quad t = \tan\theta \tag{8.11}$$

and since $\tan 2\theta = 2t/(1-t^2)$, we have that t is a root of the quadratic equation

$$t^2 + 2\gamma t - 1 = 0. \tag{8.12}$$

Having found t we can find c and s as

$$c = \frac{1}{\sqrt{1+t^2}}, \quad s = \frac{t}{\sqrt{1+t^2}}. \tag{8.13}$$

Despite the fact that this choice is somewhat more complicated than the Givens plane rotation, this is the classical use of plane rotations and when used in this way for the introduction of zeros the rotation is generally termed a **Jacobi plane rotation** (Jacobi, 1846). Such rotations are the basis of Jacobi's method of diagonalizing a symmetric matrix (Jacobi, 1846; Wilkinson, 1965) a remarkable method in that despite its age it has successfully survived the computer revolution, which cannot be said about many pre-computer eigenvalue methods.

Indeed Jacobi's method has been generalized in various ways to unsymmetric matrices of which the most promising development seems to be the norm reducing Jacobi type method (Eberlein, 1962; Eberlein, 1970).

It is of interest to note that, for any choice of θ, equation (8.8) gives

$$\begin{aligned} \alpha &= a + 2csb + s^2(d-a) = a + t(2c^2 b + cs(d-a)) \\ &= a + t((c^2 - s^2 + 1)b + cs(d-a)) \\ &= a + t(b + \beta) \end{aligned} \tag{8.14}$$

and since the trace of a matrix is preserved by a similarity transformation we also have that

$$\delta = a + d - \alpha = d - t(b+\beta). \tag{8.15}$$

For the particular case where we choose θ so that $\beta = 0$ this means that

$$\alpha = a + tb \quad \text{and} \quad \delta = d - tb, \quad \beta = 0, \tag{8.16}$$

a result that seems first to have been noticed by Rutishauser (Rutishauser, 1966). If we put

$$\tau = 1/t \tag{8.17}$$

then we can also show that for any choice of θ

$$\alpha = d + \tau(b-\beta), \quad \delta = a - \tau(b-\beta), \tag{8.18}$$

so that when $\beta = 0$ we have

$$\alpha = d + \tau b \quad \text{and} \quad \delta = a - \tau b, \quad \beta = 0. \tag{8.19}$$

As with the Givens plane rotation some care is necessary in computing c and s. In many applications of the Jacobi plane rotation it is the root of smallest modulus of equation (8.12) that is required and in this case we can compute t, c and s by using

$$\begin{array}{llllll} c=1, & t=s=0 & & & & \text{when } b=0 \\ t=\frac{z}{1+\sqrt{1+z^2}}, & c=\frac{1}{\sqrt{1+t^2}}, & s=\frac{t}{\sqrt{1+t^2}}, & z=\frac{2b}{a-d} & & \text{when } 0<|b|\leq\frac{1}{2}|a-d| \\ t=\frac{\text{sign}(z)}{|z|+\sqrt{1+z^2}}, & c=\frac{1}{\sqrt{1+t^2}}, & s=\frac{t}{\sqrt{1+t^2}}, & z=\frac{a-d}{2b} & & \text{when } \frac{1}{2}|a-d|<|b|. \end{array} \tag{8.20}$$

Notice that this root satisfies

$$|t|\leq 1, \quad c\geq 1/\sqrt{2}, \quad |s|\leq 1/\sqrt{2}. \tag{8.21}$$

If we require the root of largest modulus of equation (8.12) then we can compute $1/t, c$ and s by using

$$\begin{array}{llllll} c=1, & \tau=s=0 & & & & \text{when } b=0 \\ \tau=\frac{-z}{1+\sqrt{1+z^2}}, & c=\frac{|\tau|}{\sqrt{1+\tau^2}}, & s=\frac{\text{sign}(\tau)}{\sqrt{1+\tau^2}}, & z=\frac{2b}{a-d} & & \text{when } 0<|b|\leq\frac{1}{2}|a-d| \\ \tau=\frac{-\text{sign}(z)}{|z|+\sqrt{1+z^2}}, & c=\frac{|\tau|}{\sqrt{1+\tau^2}}, & s=\frac{\text{sign}(\tau)}{\sqrt{1+\tau^2}}, & z=\frac{a-d}{2b} & & \text{when } \frac{1}{2}|a-d|<|b|. \end{array} \tag{8.22}$$

For this root we of course have that

$$|\tau|=\left|\frac{1}{t}\right|\leq 1, \quad 0\leq c\leq 1/\sqrt{2}, \quad |s|\geq 1/\sqrt{2}. \tag{8.23}$$

If A is an $n\times n$ symmetric matrix, R_{ij} is a Jacobi plane rotation matrix in the (i,j) plane and we put

$$B=R_{ij}AR_{ij}^T, \quad i<j \tag{8.24}$$

then of course only the elements of B in rows and columns i and j are different from A and these elements are given by

$$\begin{array}{l} b_{ii}=a_{ii}+ta_{ij}=a_{jj}+\tau a_{ij}, \quad b_{jj}=a_{jj}-ta_{ij}=a_{ii}-\tau a_{ij} \\ b_{ij}=b_{ji}=0 \\ b_{ik}=ca_{ik}+sa_{jk}, \quad b_{jk}=-sa_{ik}+ca_{jk}, \quad k\neq i,j \\ b_{ki}=ca_{ki}+sa_{kj}, \quad b_{kj}=-sa_{ki}+ca_{kj}, \quad k\neq i,j. \end{array} \tag{8.25}$$

Since B is symmetric we of course only have to compute one triangle of B.

Although the principal use of plane rotations is in the introduction of zeros into matrices, there are occasions when we wish to choose the angle θ determining the plane rotation on the basis of some other criterion. For example let us put

$$A=\begin{pmatrix} a_1 & a_2 & \dots & a_r \\ b_1 & b_2 & \dots & b_r \end{pmatrix}=\begin{pmatrix} a^T \\ b^T \end{pmatrix}, \quad B=\begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_r \\ \beta_1 & \beta_2 & \dots & \beta_r \end{pmatrix}=\begin{pmatrix} \alpha^T \\ \beta^T \end{pmatrix},$$
$$R=\begin{pmatrix} c & s \\ -s & c \end{pmatrix} \tag{8.26}$$

and suppose that

$$B = RA \tag{8.27}$$

so that

$$\alpha = ca + sb \quad \text{and} \quad \beta = -sa + cb. \tag{8.28}$$

Then instead of choosing θ so that $\beta_1 = 0$ we might wish to choose θ so that the Euclidean norm of $\alpha, \ell_2(\alpha)$, is maximised (Nash, 1975). Since $\ell_2^2(\alpha) = \alpha^T\alpha$, this is equivalent to choosing θ so that y given by

$$y = \alpha^T\alpha \tag{8.29}$$

is maximised. Now equations (8.28) and (8.29) give that

$$\begin{aligned} \frac{dy}{d\theta} &= 2(ca + sb)^T(-sa + cb) = 2\alpha^T\beta \\ &= 2((c^2 - s^2)a^Tb + cs(b^Tb - a^Ta)) \end{aligned} \tag{8.30}$$

so that a turning value occurs when

$$(c^2 - s^2)a^Tb = cs(a^Ta - b^Tb). \tag{8.31}$$

If $a^Tb \neq 0$ then this gives that

$$\cot 2\theta = \frac{a^Ta - b^Tb}{2a^Tb}, \quad a^Tb \neq 0 \tag{8.32}$$

which is immediately comparable with equation (8.10) for the standard Jacobi plane rotation. Now

$$\frac{d^2y}{d\theta^2} = 2(\beta^T\beta - \alpha^T\alpha) \tag{8.33}$$

and for θ satisfying equation (8.32) we also find that

$$\alpha^T\alpha - \beta^T\beta = \frac{t}{s^2}\, a^Tb, \quad t = \tan\theta \tag{8.34}$$

and hence y is maximised if we choose that value of θ for which

$$\text{sign}\,(t) = \text{sign}\left(a^Tb\right), \quad a^Tb \neq 0. \tag{8.35}$$

Since for θ satisfying equation (8.32)

$$\alpha^T\alpha - a^Ta = t\; a^Tb \tag{8.36}$$

the root satisfying equation (8.35) also ensures that

$$\alpha^T\alpha > a^Ta, \quad a^Tb \neq 0 \tag{8.37}$$

For the case where $a^Tb = 0$ we can see that y will be maximised by choosing θ so that

$$c = \begin{cases} 1 & \text{when } a^Ta \geq b^Tb \\ 0 & \text{when } a^Ta < b^Tb \end{cases} \quad a^Tb = 0. \tag{8.38}$$

Notice from equation (8.30) that maximising y also makes

$$\alpha^T\beta = 0. \tag{8.39}$$

9 Modified Jacobi Plane Rotations

As with the Givens plane rotation, we can obtain computational savings with Jacobi plane rotations by extracting diagonal factors. Let us factorize the matrices A and B of equation (8.24) as

$$A = D\tilde{A}D, \quad B = K\tilde{B}K, \tag{9.1}$$

where

$$D = \begin{pmatrix} d_1 & 0 & \dots & 0 \\ 0 & d_2 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & d_n \end{pmatrix} \quad \text{and} \quad K = \begin{pmatrix} d'_1 & 0 & \dots & 0 \\ 0 & d'_2 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & d'_n \end{pmatrix} \tag{9.2}$$

and instead of working with A and B explicitly we consider working with their factors $D\tilde{A}D$ and $K\tilde{B}K$. Let us also suppose that it is the root of smallest modulus of equation (8.12) that is required so that the bounds of equation (8.21) hold. In particular since we know that $c \geq \frac{1}{\sqrt{2}}$, from Section 5 we expect a sensible choice to be

$$d'_i = cd_i, \quad d'_j = cd_j, \quad d'_k = d_k, \quad k \neq i, j. \tag{9.3}$$

With this choice of K equations (8.25) give that

$$\begin{aligned} &\tilde{b}_{ii} = \tfrac{1}{c^2}\left(\tilde{a}_{ii} + \tfrac{td_j}{d_i}\,\tilde{a}_{ij}\right) = \tilde{a}_{ii} + \tfrac{td_j}{d_i}\left(2\tilde{a}_{ij} + \tfrac{td_j}{d_i}\,\tilde{a}_{jj}\right) \\ &\tilde{b}_{jj} = \tfrac{1}{c^2}\left(\tilde{a}_{jj} - \tfrac{td_i}{d_j}\,\tilde{a}_{ij}\right) = \tilde{a}_{jj} - \tfrac{td_i}{d_j}\left(2\tilde{a}_{ij} - \tfrac{td_i}{d_j}\,\tilde{a}_{ii}\right) \\ &\tilde{b}_{ij} = \tilde{b}_{ji} = 0 \\ &\tilde{b}_{ik} = \tilde{a}_{ik} + \tfrac{td_j}{d_i}\,\tilde{a}_{jk}, \quad \tilde{b}_{jk} = \tilde{a}_{jk} - \tfrac{td_i}{d_j}\,\tilde{a}_{ik}, \quad k \neq i, j \\ &\tilde{b}_{ki} = \tilde{a}_{ki} + \tfrac{td_j}{d_i}\,\tilde{a}_{kj}, \quad \tilde{b}_{kj} = \tilde{a}_{kj} - \tfrac{td_i}{d_j}\,\tilde{a}_{ki}, \quad k \neq i, j. \end{aligned} \tag{9.4}$$

If we are prepared to work with D^2 and K^2 in place of D and K then we can also avoid one of the two square roots required by a standard Jacobi rotation. Once again some care is necessary in the computation to avoid underflow and overflow problems. Let us put

$$k_i = d_i^2, \quad k_j = d_j^2, \quad k'_i = (d'_i)^2 \quad k'_j = (d'_j)^2 \tag{9.5}$$

so that for the modification of equation (9.3) we have

$$k'_i = c^2 k_i, \quad k'_j = c^2 k_j. \tag{9.6}$$

Let us also put

$$m_1 = t\,\frac{d_j}{d_i}, \quad m_2 = t\,\frac{d_i}{d_j} \tag{9.7}$$

so that m_1 and m_2 are the multipliers required in the computation of equations (9.4). Then

one possible computing scheme to obtain m_1, m_2 and c^2 is given by

$$\left.\begin{array}{l} m_1 = m_2 = 0, \quad c^2 = 1 \quad \text{when } |k_i\tilde{a}_{ij}| \le \frac{1}{2}\,\delta|k_i\tilde{a}_{ii} - k_j\tilde{a}_{ij}| \\ \tilde{z} = \frac{k_i\tilde{a}_{ii} - k_j\tilde{a}_{jj}}{2k_i\tilde{a}_{ij}} \\ \alpha_1 = \frac{1}{\tilde{z}}, \qquad \alpha_2 = \frac{k_j}{k_i}\,\alpha_1, \quad \alpha = 1 + \sqrt{1 + \alpha_1\alpha_2} \qquad \text{when } |\tilde{z}| > 1 \\ \alpha_1 = \operatorname{sign}(\tilde{z}), \quad \alpha_2 = \frac{k_j}{k_i}\,\alpha_1, \quad \alpha = |\tilde{z}| + \sqrt{\tilde{z}^2 + |\alpha_2|} \quad \text{when } |\tilde{z}| \le 1 \\ m_1 = \frac{\alpha_2}{\alpha}, \quad m_2 = \frac{\alpha_1}{\alpha}, \quad c^2 = \frac{1}{1 + m_1 m_2} \\ \qquad\qquad \text{when } |k_i\tilde{a}_{ij}| > \frac{1}{2}\,\delta|k_i\tilde{a}_{ii} - k_j\tilde{a}_{ij}|. \end{array}\right\} \tag{9.8}$$

When a sequence of Jacobi rotations is to be used then we must be prepared to normalize occasionally in order to avoid underflow in the diagonal factors.

10 Plane Rotations and Pivoting

When using methods for transforming matrices that involve elementary row operations it is usually important, for the sake of numerical stability, to avoid large multipliers. To demonstrate the point consider the single elementary row transformation,

$$MA = B \tag{10.1}$$

given by

$$\begin{pmatrix} 1 & 0 \\ -m & 1 \end{pmatrix} \begin{pmatrix} a_1 & a_2 & \dots & a_r \\ b_1 & b_2 & \dots & b_r \end{pmatrix} = \begin{pmatrix} a_1 & a_2 & \dots & a_r \\ 0 & b_2' & \dots & b_r' \end{pmatrix} \tag{10.2}$$

where

$$m = \frac{b_1}{a_1}, \quad a_1 \neq 0; \quad b_i' = b_i - ma_i, \quad i = 2, 3, \dots, r. \tag{10.3}$$

If we put

$$\bar{M} = \text{fl}\,(M) \quad \text{and} \quad \bar{B} = \text{fl}\,\big(\bar{M}A\big) \tag{10.4}$$

then, corresponding to equation (3.15) for the plane rotation, here we find for the $1, \infty$ and E norms that (Reid, 1971)

$$\bar{B} = \bar{M}(A + E), \quad \text{where } \ell(E) \leq (\ell(A) + 2\ell(\bar{B}))2^{-t}. \tag{10.5}$$

If the transformation is to be stable we can see that we must not have $\ell(\bar{B})$ large relative to $\ell(A)$. Except in certain special cases, $\ell(\bar{B})$ will be large if the multiplier, m, is large. Thus, except in certain special cases such as positive definite and diagonally dominant matrices where the elements of the transformed matrices are controlled in size irrespective of the size of the multipliers (Wilkinson, 1961), we can expect transformations involving elementary row operations to be stable only if the size of the multipliers are not too large relative to unity.

To control the size of multipliers when using elementary row transformations we normally incorporate a pivoting strategy into the method and traditionally these consist of row interchanges or row and column interchanges. For example if we have the matrix

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{k1} & a_{k2} & \dots & a_{kn} \end{pmatrix} \tag{10.6}$$

and we wish to introduce zeros into the $(2,1), (3,1), \dots, (k,1)$ positions by means of elementary row transformations, then, with partial pivoting, we first pre-multiply A by the permutation matrix, P_{1r} that interchanges rows 1 and r of A, where r is chosen so that[3]

$$|a_{r1}| = \max_i |a_{i1}|. \tag{10.7}$$

[3]We assume that A is reasonably row scaled when using equation (10.7). For transformations involving elementary row operations it is not necessary to scale explicitly, but pivots should always be selected on the basis of a well scaled matrix.

We can then introduce the zeros by pre-multiplying $P_{1r}A$ by the matrix

$$M_1 = \begin{pmatrix} 1 & 0 & \dots & 0 \\ -m_{21} & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ -m_{k1} & 0 & \dots & 1 \end{pmatrix}, \tag{10.8}$$

where the multipliers are given by

$$m_{i1} = \frac{a_{i1}}{a_{r1}},\ i \neq r; \quad m_{r1} = \frac{a_{11}}{a_{r1}}. \tag{10.9}$$

From equation (10.7) this strategy obviously ensures that

$$m_{i1} \leq 1, \quad i = 2, 3, \dots, k \tag{10.10}$$

so that the transformation is stable.

Interchanges are an attractive means of pivoting because no arithmetic is involved when multiplying by a permutation matrix, but other strategies are available and in particular we can make use of plane rotations. Whether or not the increased computation time is significant will depend upon whether or not the time taken for the pivoting strategy is a significant part of the algorithm.

As an example, corresponding to the above row interchange strategy, we might select r so that

$$|a_{r1}| = \max_{i \geq 2} |a_{i1}| \tag{10.11}$$

and then pre-multiply A by the Givens plane rotation matrix R_{1r} chosen to introduce a zero into the $(r, 1)$ position of A.

We can then pre-multiply $R_{1r}A$ by the matrix

$$N_1 = \begin{pmatrix} 1 & 0 & \dots & 0 \\ -n_{21} & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ -n_{k1} & 0 & \dots & 1 \end{pmatrix}, \tag{10.12}$$

where the multipliers are given by

$$n_{i1} = \frac{a_{i1}}{\sqrt{a_{11}^2 + a_{r1}^2}}, \quad i \neq r; \quad n_{r1} = 0. \tag{10.13}$$

Once again this strategy ensures that

$$|n_{i1}| \leq 1, \quad i = 2, 3, \dots, k. \tag{10.14}$$

We also have

$$|n_{i1}| \leq |m_{i1}|, \tag{10.15}$$

so that, in terms of controlling the size of the multipliers, the use of plane rotations is never a worse strategy than the use of row interchanges.

If A is an $n \times n$ symmetric positive definite matrix partitioned as

$$A = \begin{pmatrix} a_{11} & a^T \\ a & \tilde{A} \end{pmatrix}, \tilde{A}^T = \tilde{A} \tag{10.16}$$

and we put

$$M = \begin{pmatrix} 1 & 0 \\ -m & I \end{pmatrix}, \quad \text{where } m = \frac{1}{a_{11}} \cdot a \tag{10.17}$$

then performing the congruence transformation

$$MAM^T = \begin{pmatrix} a_{11} & 0 \\ 0 & \tilde{A} - ma^T \end{pmatrix} \tag{10.18}$$

introduces zeros into the $(2,1),(3,1),\ldots,(n,1)$ positions and also retains symmetry so that only one triangle of $(\tilde{A} - ma^T)$ needs be computed. As has already been mentioned pivoting is not needed here because this transformation is quite stable when A is positive definite. When A is symmetric but indefinite then, as in the unsymmetric case, the transformation of equation (10.18) will generally only be stable if we control the size of the multipliers, that is the elements of m. If we pre-multiply A by the permutation matrix P_{1r} to interchange rows 1 and r of A we shall destroy symmetry, but if we try to retain symmetry by completing the congruence transformation $P_{1r}^T A P_{1r}$ then we are unable to move an off-diagonal element to the pivot position, we are only able to select the pivot from the diagonal elements. The matrix

$$A = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}, \quad \text{for which } A = P_{12} A P_{12}^T = P_{13} A P_{13}^T \tag{10.19}$$

shows that such a strategy is inadequate and hence various alternative strategies for factorizing symmetric indefinite matrices by means of elementary row transformations have been developed (Parlett and Reid, 1970; Bunch and Parlett, 1971; Aasen, 1971; Dax and Kaniel, 1974; Bunch and Kaufman, 1977).

Another possibility that seems likely to be useful is to replace the permutation matrix P_{1r} by a plane rotation matrix R_{1r}. There are various possibilities for choosing r, for example, as with equation (10.11), we might choose r to be such that

$$|a_{r1}| = \max_{i \geq 2} |a_{i1}|. \tag{10.20}$$

If we put

$$B = R_{1r} A R_{1r}^T \tag{10.21}$$

and we choose R_{1r} to be the Jacobi plane rotation matrix that makes

$$b_{r1} = 0 \tag{10.22}$$

then, from equations (8.12), (8.16) and (8.19) we find that

$$b_{11} = \frac{1}{2}(a_{11} + a_{rr}) + \frac{1}{2}(t + \frac{1}{t}) a_{r1} = \frac{1}{2}(a_{11} + a_{rr}) \pm a_{r1}\sqrt{1 + \gamma^2}. \tag{10.23}$$

Thus if we choose the root to be such that

$$\text{sign}\,(t) = \text{sign}\left(\frac{a_{11} + a_{rr}}{a_{r1}}\right) \tag{10.24}$$

then we certainly have that

$$|b_{11}| \geq |a_{i1}|, \quad i \geq 2. \tag{10.25}$$

This does not of course guarantee that $|b_{11}| \geq |b_{i1}|,\ i \geq 2$, but at least if A has been reasonably well scaled then we shall obtain reasonably sized multipliers.

References

Aasen, J. O. (1971). On the reduction of a symmetric matrix to tridiagonal form, *BIT* **11**: 233–242.

Bunch, J. R. and Kaufman, L. (1977). Some stable methods for calculating inertia and solving symmetric linear systems, *Math. Comp.* **31**: 163–179.

Bunch, J. R. and Parlett, B. N. (1971). Direct methods for solving symmetric indefinite systems of linear equations, *SIAM J. Num. Anal.* **8**: 639–655.

Dax, A. and Kaniel, S. (1974). Pivoting techniques for symmetric decomposition, *Technical report*, Institute of Mathematics, The Hebrew University of Jerusalem, Jerusalem, Israel. (Published as (Dax and Kaniel, 1977)).

Dax, A. and Kaniel, S. (1977). Pivoting techniques for symmetric Gaussian elimination, *Numer. Math.* **28**: 221–242.

Eberlein, P. J. (1962). A Jacobi-like method for the automatic computation of eigenvalues and eigenvectors of an arbitrary matrix, *J. SIAM* **10**: 74–88.

Eberlein, P. J. (1970). Solution of the complex eigenproblem by a norm reducing Jacobi type method, *Numer. Math.* **14**: 232–245. (See also (Wilkinson and Reinsch, 1971, pp 404–417)).

Fletcher, R. and Powell, M. J. D. (1974). On the modification of LDL^T factorizations, *Math. Comp.* **28**: 1067–1087.

Francis, J. G. F. (1961). The QR transformation: A unitary analogue to the LR transformation, part I, *Computer J.* **4**: 265–271.

Francis, J. G. F. (1962). The QR transformation: A unitary analogue to the LR transformation, part II, *Computer J.* **4**: 332–345.

Gentleman, W. M. (1973). Least squares computations by Givens transformations without square roots, *J. Inst. Maths Applics* **12**: 329–336.

Gentleman, W. M. (1975). Error analysis of QR decompositions by Givens transformations, *Linear Algebra Appl.* **10**: 189–197.

Gill, P. E., Golub, G. H., Murray, W. and Saunders, M. A. (1974). Methods for modifying matrix factorizations, *Math. Comp.* **28**: 505–535.

Gill, P. E. and Murray, W. (1970). A numerically stable form of the simplex algorithm, *Technical Report DNAM 87*, National Physical Laboratory, Teddington, Middlesex TW11 0LW, UK. (Published as (Gill and Murray, 1973)).

Gill, P. E. and Murray, W. (1973). A numerically stable form of the simplex algorithm, *Linear Algebra Appl.* **7**: 99–138.

Gill, P. E. and Murray, W. (1977). Modification of matrix factorizations after a rank-one change, *in* D. A. H. Jacobs (ed.), *The State of the Art in Numerical Analysis*, Academic Press, London, UK, pp. 55–83. (Proceedings of the IMA Conference, University of York, 1976).

Givens, W. (1954). Numerical computation of the characteristic values of a real symmetric matrix, *Technical Report ORNL-1574*, Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831, USA.

Givens, W. (1958). Computation of plane unitary rotations transforming a general matrix to triangular form, *J. SIAM* **6**: 26–50.

Golub, G. H. and Kahan, W. (1968). Least squares, singular values and matrix approximations, *Aplik. mat.* **13**: 44–51.

Golub, G. H. and Reinsch, C. (1970). Singular value decomposition and least squares solutions, *Numer. Math.* **14**: 403–420. (See also (Wilkinson and Reinsch, 1971, pp 134–151)).

Hammarling, S. (1974). A note on modifications to the Givens plane rotation, *J. Inst. Maths Applics* **13**: 215–218.

Householder, A. S. (1958). Unitary triangularization of a nonsymmetric matrix, *J. ACM* **5**: 339–342.

Jacobi, C. G. J. (1846). Über ein leichtes verfahren die in der theorie der säcularstörungen vorkommenden gleichungen numerisch aufzulösen, *J. für die reine und angewande Mathematik (Crelle's J.)* **30**: 51–94.

Kublanovskaya, V. N. (1961). On some algorithms for the solution of the complete eigenvalue problem, *Zhurnal Vychislitelnoi Matematiki i Matematicheskoi Fiziki* **1**: 555–570. (In Russian. Translation in *USSR Computational Mathematics and Mathematical Physics,* **1**, 637–657, 1962).

Nash, J. C. (1975). A one-sided transformation method for the singular value decomposition and algebraic eigenproblem, *Computer J.* **18**: 74–76.

Parlett, B. N. and Reid, J. K. (1970). On the solution of a system of linear equations whose matrix is symmetric but not definite, *BIT* **10**: 386–397.

Reid, J. K. (1971). A note on the stability of Gaussian elimination, *J. Inst. Maths Applics* **8**: 374–375.

Rutishauser, H. (1966). The Jacobi method for real symmetric matrices, *Numer. Math.* **9**: 1–10. (See also (Wilkinson and Reinsch, 1971, pp 202–211)).

Stewart, G. W. (1976). The economical storage of plane rotations, *Numer. Math.* **25**: 137–138.

Wilkinson, J. H. (1961). Error analysis of direct methods of matrix inversion, *J. ACM* **8**: 281–330.

Wilkinson, J. H. (1965). *The Algebraic Eigenvalue Problem*, Oxford University Press, Oxford, UK. (Also translated into Russian by Nauka, Russian Academy of Sciences, 1970).

Wilkinson, J. H. (1977). Some recent advances in numerical linear algebra, *in* D. A. H. Jacobs (ed.), *The State of the Art in Numerical Analysis*, Academic Press, London, UK, pp. 3–53. (Proceedings of the IMA Conference, University of York, 1976).

Wilkinson, J. H. and Reinsch, C. (eds) (1971). *Handbook for Automatic Computation, Vol.2, Linear Algebra*, Springer-Verlag, Berlin, Germany.