Tik lask 3 0143388-36 Anni [illegible]

(1) For orthogonal matrix A $AA^T = I$ where I is unit matrix. Matrix Q_{pq} has the form

$$Q_{pq} = \begin{bmatrix} 1 & & & & & & & & \\ & \ddots & & \vdots & & \vdots & & & \\ & & 1 & & & & & & \\ & \cdots & & c & \cdots & -s & \cdots & & \\ & & & \vdots & 1 & & & & \\ & & & \vdots & & \ddots & & & \\ & & & \vdots & & 1 & & & \\ & \cdots & & s & \cdots & c & & & \\ & & & & & & 1 & & \\ & & & \vdots & & & & \ddots & \\ & & & & & & & & 1 \end{bmatrix}$$

p-1 rows before c

q-1 rows before c

zeroes everywhere else

p-1 cols before c

q-1 cols before c

Therefore row i of $N \times N$ matrix Q_{pq} is

$$[\underbrace{0, 0, \ldots,}_{i-1} 1, \underbrace{0, 0, \ldots, 0}_{N-i}]$$

if $i \neq p$ and $i \neq q$

or

$$[\underbrace{0, 0, \ldots, 0,}_{p-1} s, \underbrace{0, \ldots, 0,}_{q-p-1} c, \underbrace{0, \ldots, 0}_{N-q}]$$

if $i = q$

or

$$[\underbrace{0, 0, \ldots, 0,}_{p-1} c, \underbrace{0, \ldots, 0,}_{q-p-1} -s, \underbrace{0, \ldots, 0}_{N-q}]$$

if $i = p$

(1) And for columns j in matrix Q_{pq}^T

$j \neq p$ and $j \neq q$:

$$\begin{array}{r} j-1 \left\{ \vphantom{\begin{matrix}0\\0\\ \vdots\\0\end{matrix}} \right. \\ \\ N-j \left\{ \vphantom{\begin{matrix}0\\ \vdots\\0\end{matrix}} \right. \end{array} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$j = q$:

$$\begin{array}{r} p-1 \left\{ \vphantom{\begin{matrix}0\\0\\ \vdots\\0\end{matrix}} \right. \\ \\ q-p-1 \left\{ \vphantom{\begin{matrix}0\\ \vdots\\0\end{matrix}} \right. \\ \\ N-q \left\{ \vphantom{\begin{matrix}0\\ \vdots\\0\\0\end{matrix}} \right. \end{array} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ s \\ 0 \\ \vdots \\ 0 \\ c \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

$j = p$

$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ c \\ 0 \\ \vdots \\ 0 \\ -s \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{array}{l} \left. \vphantom{\begin{matrix}0\\0\\ \vdots\\0\end{matrix}} \right\} j-1 \\ \\ \left. \vphantom{\begin{matrix}0\\ \vdots\\0\end{matrix}} \right\} q-p-1 \\ \\ \left. \vphantom{\begin{matrix}0\\ \vdots\\0\end{matrix}} \right\} N-q \end{array}$$

So multiplying row i of Q and column j of Q^T when $i = j$ (diagonal of QQ^T) we get

$$(i-1) \cdot 0 + 1 \cdot 1 + (N-i) \cdot 0 = 1$$

and when $i \neq j$ and $i \neq p$ and $i \neq q$ every index with something nonzero in column j is zeroed out with zero in row i and vice versa. Now for for example 6×6 matrix $Q_{3,5}$ we have filled all but cells marked with $\square$

$$QQ^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ \square & \square & 1 & \square & \square & \square \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \square & \square & \square & \square & 1 & \square \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Only rows $i = p$ and $i = q$ remain unknown.

(1) Next we'll check the case $i = p$ and $j \neq i$:

If $j = q$ multiplication yields

$$0 + 0 + \dots + 0 + cs + 0 + 0 + \dots + 0 - cs + 0 \dots + 0$$

$$= 0 + cs - cs$$

$$= 0$$

And for cells $j \neq p$ we get

$$0 + 0 + \dots + 0 + cs + 0 + \dots + 0 - cs + 0 + \dots + 0$$

$$= cs - cs$$

$$= 0$$

Therefore all for all $i \neq j$ $(QQ^T)_{i,j} = 0$ and for all $i = j$ $(QQ^T)_{i,j} = 1$ which means that $QQ^T = I$ and therefore Q is orthogonal $\square$

2. Values from my implementation and LAPACK function dgeev seem to output numbers very close to each other. For following matrix

```
        4,      -30,      60,       -35,
        -30,    300,      -675,     420,
        60,     -675,     1620,     -1050,
        -35,    420,      -1050,    700
```

output is

```
     jacobi method:
    0.166642861171900275       37.101491365127692745
2585.253810928922121093         1.478054844778173171
LAPACK implementation:
  2585.253810928920302104       37.101491365127600375
0.166642861171833745       1.478054844778099008
```

from which we see that errors are very small as we see them starting from around 13^{th} significant digit. I also ran some other arbitrarily chosen matrices for which the results were similar.

3. For determining effects of perturbations I had to edit my jacobi function so that it stops after $1000*N^2$ iterations even if matrix isn't diagonal yet. This is probably more than enough to get an reasonably close to best possible value with this algorithm.

I ran 100 runs with random diagonal N*N matrices with sizes evenly distributed having N between 5 and 15 (inclusive). Perturbation size was also determined randomly with perturbations ranging from 0.008% to 9% with 4.6% mean. This resulted in f having mean 1.22 and standard deviation 4.1. As one can see when examining raw output in file perturbations.out, for most matrices f is quite small (median being 0.027) but few matrices with f values even as high as over 20 raise the mean significantly.