18.330 Handout on **Householder reflections** Wed 28 Apr 04

... cribbed from this course in '87 or '88; hence the timings from the speedy 8 MHz IBM PC/AT! AT #2

Continuing with the 5x5 **Pascal matrix** P =

$$P = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 3 & 6 & 10 & 15 \\ 1 & 4 & 10 & 20 & 35 \\ 1 & 5 & 15 & 35 & 70 \end{bmatrix},$$

here is how the EISPACK subroutine TRED1 (which "reduces a real symmetric matrix to a symmetric tridiagonal matrix using orthogonal similarity transformations" of the Householder kind) actually achieves its goal:

<u>Stage 1</u>: Using the unit vector $v_1 = (0,3,1,1,1)^T/\mathrm{sqrt}(12)$ suggested by the first row or column of P, the Householder matrix $H_1 = I - 2v_1v_1^T$ produces a new symmetric matrix

$$P_1 = H_1PH_1 = \begin{bmatrix} 1 & -2 & 0 & 0 & 0 \\ -2 & 60.5 & 5.5 & -12 & -40 \\ 0 & 5.5 & 1.167 & -1 & -5.667 \\ 0 & -12 & -1 & 2.833 & 8.167 \\ 0 & -40 & -5.667 & 8.167 & 33.5 \end{bmatrix}$$

here cited only to 3 decimals, but in principle with the same spectrum as P.

<u>Stage 2</u>: To produce two similar zeroes at the bottom of the second column (and at the right-hand end of the second row), TRED1 employs a new unit vector $v_2 = w_2/\mathrm{sqrt}(w_2)$, where $w_2^T = (0,0,5.5+\mathrm{root},-12,40)$ and root = sqrt(5.5**2 + 12**2 + 40**2). Hence via $H_2 = I - 2v_2v_2^T$:

$$P_2 = H_2P_1H_2 = \begin{bmatrix} 1 & -2 & 0 & 0 & 0 \\ -2 & 60.5 & -42.122 & 0 & 0 \\ 0 & -42.122 & 36.358 & -1.935 & -0.547 \\ 0 & 0 & -1.935 & 1.070 & 0.213 \\ 0 & 0 & -0.547 & 0.213 & 0.072 \end{bmatrix}$$

Stage 3: Finally, to cleanse also the bottom of the third column, TRED1 employs a third Householder vector $v_3 = w_3/\mathrm{sqrt}(w_3)$, where $w_3^T = (0,0,0, -1.935-\mathrm{root},-0.547)$ and now root = sqrt(1.935**2 + 0.547**2) , of course to several more decimals. Use of the orthogonal matrix $H_3 = I - 2v_3v_3^T$ in a third successive similarity transformation then yields a final matrix P^3 with diagonal elements

1.0, 60.5, 36.35825, 1.107790, 0.033964

and a subdiagonal consisting of

-2.0, -42.12185, 2.010798, -0.079630 .

And that's where some other EISPACK subroutine like TQLRAT (that "finds the eigenvalues of a symmetric tridiagonal matrix by the rational QL method") or say the TQLI (= tridiagonal QL with implicit shifts) algorithm from the book by Press et al will normally take over and finish the job in a relatively short time of $O(N^2)$, as opposed to the $O(N^3)$ operations typically involved in the above triangularizing.

To give a better feeling for these relative times, a 40x40 Hilbert matrix on my 8 MHz PC/AT last week required 5.2 sec for the TRED1 tridiagonalizing, and 1.9 sec for the QL finishing touches, whereas for an 80x80 Hilbert matrix those times swelled to 36.4 and 7.1 sec, respectively.

Actually I fibbed a little, for the sake of clarity, in describing the inner workings of TRED1. What I omitted were two small but welcome extra tricks for reducing roundoff errors and/or cancellation: a) Well-constructed numerical algorithms always worry about "scaling" or "balancing" various entries to make them more fairly comparable, and this one is no exception. And b) on the stated assumption that the absolutely largest elements of the input matrix **P** reside near the bottom righthand corner, TRED1 actually begins its Householder zero-making efforts with the last row or column whose present entries read (1,5,15,35,70), and it marches from there toward the upper left.

In other words, the first Householder vector preferred by this program for treating our current patient **P** is $\mathbf{v}_1 = (1,5,15,35+\mathrm{sqrt}(1476),0)^T/\mathrm{const}$, and the eventual diagonal elements produced are

0.068577, 0.603072, 2.338515, 25.98984, 70.0

and the subdiagonal ones are

0.113383, 0.538719, 2.746376, -38.41874 .

It is certainly not obvious at first sight that these nine numbers imply the same eigenvalues as before ... but stranger things have been known to happen!

AT