Hessenberg Similarity

One of the most important condensed forms in numerical linear algebra is the Hessenberg matrix. An upper Hessenberg matrix is almost upper triangular, having zeros below the first subdiagonal.

Every square matrix is orthogonally similar to a Hessenberg matrix. There are methods for reducing $A \in \mathbb{R}^{n \times n}$ to $H = V^T AV$ based on Gram-Schmidt (the *Arnoldi method*) and on Householder reflectors (a slight modification to the Householder QR method). These methods are about twice as expensive as their QR analogs (the Householder Hessenberg reduction requires about $\frac{10}{3}n^3$ flops), but the resulting matrix is *similar* to A.

There are several reasons Hessenberg matrices are important, but I would suggest that the following is the most fundamental: Suppose that you wanted to solve $(A - sI)x = b$ for m different values of s. Gaussian elimination (or QR) would require a new factorization for each s, resulting in a cost of $\mathrm{O}(mn^3)$ flops. But

$$(A - sI)x = b \quad \Rightarrow \quad V^T(A - sI)VV^T x = V^T b \quad \Rightarrow \quad (H - sI)y = z,$$

which can be solved for y in $\mathrm{O}(n^2)$ flops, and x can be recovered as $x = Vy$ in another $\mathrm{O}(n^2)$ flops. This gives a total cost of $\mathrm{O}(n^3 + mn^2)$ flops (which is more efficient than G.E. on $A - sI$ for all values of m bigger than about 6)!

Since H is similar to A, they have the same eigenvalues. If you now consider the shifted inverse power method, you see that even shifting several times for each eigenvalue, we could find all eigenvalues in $\mathrm{O}(n^3)$ flops (rather than the $\mathrm{O}(n^4)$ complexity of G.E. in this context).

As another application, consider evaluating $p(A)$ for some polynomial p. There are many ways to approach this problem (and Horner's method isn't optimal for matrices), but $p(A) = p(VHV^T) = Vp(H)V^T$ suggests reduction to Hessenberg form and then evaluating $p(H)$. If Horner's method is used, the Hessenberg approach is faster than Horner on A if the degree of p is bigger than about 6.

In fact, there is another Hessenberg-based method for finding $p(A)$: Compute the QR factorization of $H - s_1 I$: $Q_1 R_1 = H - s_1 I$. Now define $H_1 = R_1 Q_1 + s_1 I$. Repeat this:

$$Q_k R_k = H_k - s_k I \quad \text{and then} \quad H_{k+1} = R_k Q_k + s_k I.$$

If this process is repeated for $k = 1, 2, \ldots, m$, then

$$(Q_1 Q_2 \cdots Q_m)(R_m \cdots R_2 R_1) = (H - s_1 I)(H - s_2 I) \cdots (H - s_m I)$$

giving the QR factorization $QR = p(H)$, where $p(x) = \prod_{i=1}^m (x - s_i)$.

Under mild hypotheses on H, the iteration above with $s_i = 0$, converges to a (block 2×2) upper triangular matrix T: $H_k \longrightarrow T$ as $k \to \infty$. The H_k are all similar to H, and thus the real eigenvalues of H are the real diagonal elements of T. This is the idea behind the *QR algorithm*.