Fast and backward stable computation of the zeros of polynomials

Jared L. Aurentz[1], Thomas Mach[2], Raf Vandebril[3], and David S. Watkins[4]

[1]University of Oxford, aurentz@maths.ox.ac.uk
[2]KU Leuven, thomas.mach@cs.kuleuven.be
[3]KU Leuven, raf.vandebril@cs.kuleuven.be
[4]Washington State University, watkins@math.wsu.edu

We present a fast and backward stable method for computing eigenvalues of upper Hessenberg unitary-plus-rank-one matrices, that is, matrices of the form $A = \tilde{U} + \tilde{x}\tilde{y}^T$, where $\tilde{U}$ is unitary, and A is upper Hessenberg. This includes the class of Frobenius companion matrices, so this method can be used to find the zeros of a polynomial.

The unitary-plus-rank-one structure is preserved by any method that performs unitary similarity transformations, including Francis's implicitly-shifted QR algorithm. We present a new implementation of Francis's algorithm that acts on a data structure that stores the matrix in $O(n)$ space and performs each iteration in $O(n)$ time. The method is backward stable.

We store A in the form $A = QR$, where Q is unitary and R is upper triangular. In this sense our method is similar to one proposed by Chandrasekaran et. al. [1], but our method stores R differently. Since Q is a unitary upper-Hessenberg matrix, it can be stored as a product $Q = Q_1 Q_2 \cdots Q_{n-1}$, where each Q_j is a Givens-like unitary tranformation that acts only on rows j and $j+1$. We call these Q_j *core transformations*. Both our algorithm and that of Chandrasekaran et. al. use this representation of Q. For R, they use a quasiseparable generator representation. Our representation scheme factors R in the form

$$R = C_{n-1} \cdots C_1 (B_1 \cdots B_{n-1} + e_1 y^T),$$

where the C_j and B_j are unitary core transformations. This is possible because R is also unitary-plus-rank-one.

The Hessenberg matrix A takes the form

$$A = QR = Q_1 \cdots Q_{n-1} C_{n-1} \cdots C_1 (B_1 \cdots B_{n-1} + e_1 y^T),$$

and thus is represented by about $3n$ core transformations plus the rank-one part. In fact there is some redundancy in the representation. The information about the rank-one part is also encoded in the core transformations, so it is not necessary to store the rank-one part explicitly. Performing a Francis iteration on a matrix stored in this form is entirely a matter of manipulating core transformations. We will show how to do this.

Our method is about as accurate as and much faster than the (slow) Francis algorithm applied to the companion matrix without exploiting the structure. It is faster than other fast and (allegedly) backward stable methods that have been proposed, and it has comparable or better accuracy.

References

[1] S. Chandrasekaran, M. Gu, J. Xia, and J. Zhu, *A fast QR algorithm for companion matrices,* Operator Theory: Advances and Applications, 179, pp. 111–143, 2007