BIT 34 (1994), 165–176.

# THE RATIONAL KRYLOV ALGORITHM FOR NONSYMMETRIC EIGENVALUE PROBLEMS. III: COMPLEX SHIFTS FOR REAL MATRICES

AXEL RUHE

*Department of Computing Science*
*Chalmers Institute of Technology and the University of Göteborg*
*S-41296 Göteborg, Sweden*

*Dedicated to Carl-Erik Fröberg on the occasion of his 75th birthday.*

**Abstract.**

A new algorithm for the computation of eigenvalues of a nonsymmetric matrix pencil is described. It is a generalization of the shifted and inverted Lanczos (or Arnoldi) algorithm, in which several shifts are used in one run. It computes an orthogonal basis and a small Hessenberg pencil. The eigensolution of the Hessenberg pencil, gives Ritz approximations to the solution of the original pencil. It is shown how complex shifts can be used to compute a real block Hessenberg pencil to a real matrix pair.

Two applicationx, one coming from an aircraft stability problem and the other from a hydrodynamic bifurcation, have been tested and results are reported.

*AMS subject classifications:* 65F15

*Key words:* eigenvalue, matrix pencil, nonsymmetric, Krylov, double shift, inverse iteration, Lanczos method, Arnoldi method

## 1. Introduction and review.

The Rational Krylov algorithm computes a selected set of eigenvalues to a nonsymmetric pencil,

$$(A - \lambda B)x = 0, \tag{1}$$

where the matrices are too large to be treated by standard similarity transformations as in e.g. the Householder-QR algorithm, but not too large for a sparse Gaussian factorization. It extends the Spectral Transformation Lanczos algorithm [3, 4, 6] by using several shifts $\mu_j$ in one run, replacing the Lanczos polynomials in the Krylov space description of the algorithm by a rational function,

Received November 1993.

$$r(\lambda) = \frac{p_{j-1}(\lambda)}{(\lambda - \mu_1)(\lambda - \mu_2)\dots(\lambda - \mu_j)} = \frac{c_1}{\lambda - \mu_1} + \dots + \frac{c_j}{\lambda - \mu_j}. \tag{2}$$

The poles, $\mu_j$, are the shifts, and the zeros of the numerator approximate the eigenvalues. A careful choice of shifts gives a much faster convergence to eigenvalues close to the poles.

This approach comes to advantage when a factorization of the matrix is reasonably cheap, compared to solution of a system and vector recursion. It may be contrasted to strategies where the Arnoldi algorithm is restarted several times for the same shift, as in the recent contribution by Sorensen [17]. When no factorization is possible, only much slower algorithms are available, built either around minimization of the Rayleigh quotient, see e.g. [10], direct Lanczos on a nearby standard problem, see Scott [16], preferably with some preconditioning in the spirit of Davidson [7], or a two level iteration as expounded by Szyld [19].

The first outline of Rational Krylov was given in [12], but with no numerical experience. In the first report in this series [13], we describe the Rational Krylov algorithm for the standard case, when $B$ is regarded as a weight matrix of full rank, and the second [14] we show how to treat cases when both $A$ and $B$ may be singular, and how to apply the perturbation theory for regular pencils from [18] to bound the errors. In this third report, we show how to apply complex double shifts to a real pencil (1), extending an idea of Partlett and Saad [8].

We will continue in section 2, by reviewing the basic recursion that gives an orthogonal basis $V$, where the pencil $(A, B)$, (1) is represented by a Hessenberg pencil $(K, H)$. In section 3, we show how to get Ritz approximations to eigenvalues and eigenvectors from this Hessenberg pencil. In section 4, we show to apply two complex conjugate shifts by first doing one complex solve, and then adding two real columns to the basic $V$ and the real Hessenberg pencil $(K, H)$. Though we reason as if $B$ is nonsingular, in our implementation we have used the recursion from [14] that is applicable also to a regular pencil with singular $B$ matrix.

In the final section 5, we apply the real and complex variants of the algorithm to two numerical test cases. The tests show that the real double shift variant comes to advantage, when we get a mixture of real and complex shifts, as happens when most of the eigenvalues of interest are in a region around the real axis. When we seek eigenvalues far out in the imaginary, the real variant needs to work longer, since it must get double the number of eigenvalues, while the complex variant can keep shifting in the upper half plane, and one can get those below by conjugation.

A word about notation: We let $V_j$ stand for a matrix with $j$ columns, the first $j$ columns of $V$ if nothing else is stated, $A_{jk}$ is a $j \times k$ matrix, but we avoid subscripts when all rows or columns are referred to. Column $j$ of the matrix $V$ is $v_j$. We denote by $\bar{\eta}$ the complete conjugate of the number $\eta$, and by $A^H$ the conjugate transpose of the matrix $A$. We will always use the Euclidean vector norm, denoted by $\| \ \|$.

## 2. The Rational Krylov iteration.

Starting with a vector $v_1$, we build up an orthonormal basis $V_j$, one column vector $v_j$ at a time, using the following:

ALGORITHM RKS: *Choose starting vector* $v_1$
*For* $j = 1, 2, \ldots$ *until convergence*

1. *Choose shift* $\mu_j$ *and starting combination* $r = V_j t_j$
2. *Operate* $r := (A - \mu_j B)^{-1} Br$
3. *Orthogonalize* $r := r - V_j h_j$ *where* $h_j = V_j^H r$
4. *Get new vector* $v_{j+1} := r/h_{j+1,j}$, *where* $h_{j+1,j} = \|r\|$
5. *Compute approximative solution and test for convergence*

This algorithm differs from the shifted and inverted Arnoldi algorithm [1, 15, 11] only in that the shift $\mu_j$ in step 1 may very with $j$, and that the iteration is continued, not with the last available vector $v_j$, but with a combination, $V_j t_j$, of all the vectors already computed. Most often we take either the first vector $v_1$ or the last vector $v_j$.

For economic reasons, it is advisable to keep the shift $\mu_j$ constant for several steps, since then we can use the same factorized matrix

$$A - \mu_j B = LU \tag{3}$$

in all of those.

Now let us follow what happens. Eliminate the intermediate vector $r$ and get,

$$V_{j+1} h_j = (A - \mu_j B)^{-1} B V_j t_j.$$

Multiply from the left by $(A - \mu_j B)$,

$$(A - \mu_j B) V_{j+1} h_j = B V_j t_j. \tag{4}$$

Separate terms with $A$ to the left and $B$ to the right,

$$A V_{j+1} h_j = B V_{j+1} (h_j \mu_j + t_j),$$

now with a zero added to the bottom of the vector $t_j$ giving it length $j + 1$.

This is the relation for the $j$th step, now put the corresponding vectors from the previous steps in front of this and get,

$$A V_{j+1} H_{j+1,j} = B V_{j+1} K_{j+1,j}, \tag{5}$$

with two $(j + 1) \times j$ Hessenberg matrices, $H_{j+1,j}$ consisting of the Gram Schmidt orthogonalization coefficients, and

$$K_{j+1,j} = H_{j+1,j} \operatorname{diag}(\mu_i) + T_{j+1,j}. \tag{6}$$

### 3. Finding an approximate eigensolution.

Now let us describe how to find an approximative solution and test for convergence in step 5 of ALGORITHM RKS. In the exceptional case of total convergence, we would get $h_{j+1,j} = 0$ and a zero last row in both $H$ and $K$. Otherwise find an approximative solution $\theta$ by solving the problem,

$$(K_{j,j} - \theta H_{j,j})s = 0, \tag{7}$$

by means of the QZ-algorithm. For a given solution $(\theta, s)$ of this, we take the vector,

$$x = V_{j+1} H_{j+1,j} s, \tag{8}$$

as a Ritz vector for the original problem (1). Its residual is,

$$\begin{aligned}(A - \theta B)x &= (A - \theta B) V_{j+1} H_{j+1,j} s \\ &= B V_{j+1}(K_{j+1,j} - \theta H_{j+1,j}) s \\ &= B v_{j+1}(k_{j+1,j} - \theta h_{j+1,j}) s_j \\ &= B v_{j+1}(\mu_j - \theta) h_{j+1,j} s_j \\ &= B v_{j+1} \beta_{j,i},\end{aligned} \tag{9}$$

the second equality following from (5), the third from (7), and the fourth from (6). We let

$$\beta_{j,i} = (\mu_j - \theta_i) h_{j+1,j} s_{j,i} \tag{10}$$

estimate the norm of the residual for the $i$th eigenvalue approximation $\theta_i$.

Compare this to the well known expression for the Lanczos algorithm [9, eq. (13-2-1) on p. 260], and note that we can estimate the norm of the residual using only short vectors of length $j$, Moreover, the residual is $B$-orthogonal to the subspace spanned by $V_j$.

### 4. Complex shifts for real pencils.

When the matrices $A$ and $B$ are both real, the pencil (1) has complex pairs of eigenvalues. We can make sure that also the Ritz approximations $\theta$ in (7) occur in pairs, by computing a real basis $V$, and a real pencil $(K, H)$ in the following way.

When the shift $\mu$ is real, all other quantities become real, and we can proceed with ALGORITHM RKS. When we have a complex shift $\mu$, we do all the factorization and solution operations on the shifted matrix $(A - \mu B)$ in complex arithmetic, but we separate the resulting vector $r$, in step 2 of the algorithm, into real and imaginary parts giving *two* new vectors $v_{j+1}$ and $v_{j+2}$. Each of this is a different rational function, and we see that for any real vector $x$,

$$\operatorname{Re}((A-\mu B)^{-1}Bx)=\frac{1}{2}[((A-\mu B)^{-1}+(A-\bar{\mu}B)^{-1})Bx]$$

$$\operatorname{Im}((A-\mu B)^{-1}Bx)=\frac{1}{2i}[((A-\mu B)^{-1}+(A-\bar{\mu}B)^{-1})Bx].$$

A linear combination corresponds to taking a double shift, adding the quadratic term,

$$\frac{c_1}{\lambda-\mu}+\frac{c_2}{\lambda-\bar{\mu}}=\frac{p_1(\lambda)}{(\lambda^2-2p\lambda+q)},$$

To do this, we replace steps 3 and 4 of ALGORITHM RKS by:

*3' Separate* $r=r_1+ir_2$,

*3a Orthogonalize* $r_1=r_1-V_jh_j$, *where* $h_j=V_j^Tr_1$,

*4a Normalize* $v_{j+1}:=r_1/h_{j+1,j}$, *where* $h_{j+1,j}=\|r_1\|$,

*3b Orthogonalize* $r_2=r_2-V_{j+1}h_{j+1}$, *where* $h_{j+1}=V_{j+1}^Tr_2$,

*4b Normalize* $v_{j+2}:=r_2/h_{j+2,j+1}$, *where* $h_{j+2,j+1}=\|r_2\|$,

all in real arithmetic.

We will get the modified basic recursion by noting that,

$$\operatorname{Re}((A-\mu B)^{-1}BV_jt_j)=V_{j+1}h_j,$$
$$\operatorname{Im}((A-\mu B)^{-1}BV_jt_j)=V_{j+2}h_{j+1},$$

and then sum,

$$V_{j+2}(h_j+ih_{j+1})=(A-\mu B)^{-1}BV_jt_j,$$

with a zero added to the bottom of the $j+1$ vector $h_j$ to give both vectors length $j+2$. Precisely as before (4), multiply and get

$$(A-\mu_jB)V_{j+2}(h_j+ih_{j+1})=BV_jt_j,$$

but then separate real and imaginary parts again, setting

$$\mu=\rho+i\theta,$$

and get

$$AV_{j+2}[h_j,h_{j+1}]=BV_{j+2}\quad[h_j,h_{j+1}]\begin{pmatrix}\rho_j & \theta_j\\ -\theta_j & \rho_j\end{pmatrix}+\begin{bmatrix}t_j & 0\\ 0 & 0\\ 0 & 0\end{bmatrix}.$$

Put earlier columns in front and get the modified basic recursion (5),

$$AV_{j+2}H_{j+2,j+1} = BV_{j+2}K_{j+2,j+1}, \tag{11}$$

with the real Hessenberg matrix $H$, and

$$K = HM + T,$$

which is now real block Hessenberg since

$$M = \operatorname{diag}\left(\mu_j, \ldots, \begin{pmatrix} \rho_j & \theta_j \\ -\theta_j & \rho_j \end{pmatrix}\right)$$

is block diagonal. It contains $2 \times 2$ blocks where complex shifts have been used, for instance a sequence (*real*, *complex*, *real*, *real*, *complex*) will give a $8 \times 7$ matrix,

$$K = \begin{pmatrix} x & x & x & & & & \\ x & x & x & & & & \\ 0 & x & x & & & & \\ 0 & x & x & x & & & \\ 0 & 0 & 0 & x & x & & \\ 0 & 0 & 0 & 0 & x & x & x \\ 0 & 0 & 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & 0 & x & x \end{pmatrix}$$

## 5. Two numerical examples.

We have tested the Rational Krylov algorithm using MATLAB4 on SUN4 workstations. The linear system computations in step 2 of ALGORITHM RKS were done with the sparse matrix option in MATLAB4. Reorthogonalization was done in step 2 whenever necessary. We took advantage of the complex arithmatic in MATLAB, even when we had real matrices.

We started with the shift at a point *goal* in that part of the complex plane, where we were interested in the eigenvalues. We used (10) to follow the residuals, and flagged an eigenvalue as converged whenever it got smaller than *tolconv* $= .5_{10} - 8$. Then we took the best eigenvalue, that had a residual larger than *tolshift* $= .5_{10} - 4 \approx$ *tolconv*$^{1/2}$, as the next shift, this in order to save some factorizations and avoid nearly singular shifted matrices. We kept the same shift at most 5 steps.

Let us first report some runs on the test matrix TOLOSA taken from [2]. It is typical for those matrices one obtains when calculating the stability of an aircraft structure, and for a small $n$ it looks like figure 1 left, with the spectrum as in figure 1 right. In our tests we took a larger $n = 2000$, and sought the eigenvalues out in the upper left end of the spectrum by choosing *goal* $= -750 + 2400i$. These eigenvalues are the worst conditioned and also critical for the actual design.

First we used the complex algorithm, ALGORITHM RKS, as described in section 2. In figure 2, the residuals for the different eigenvalues are shown as functions of the

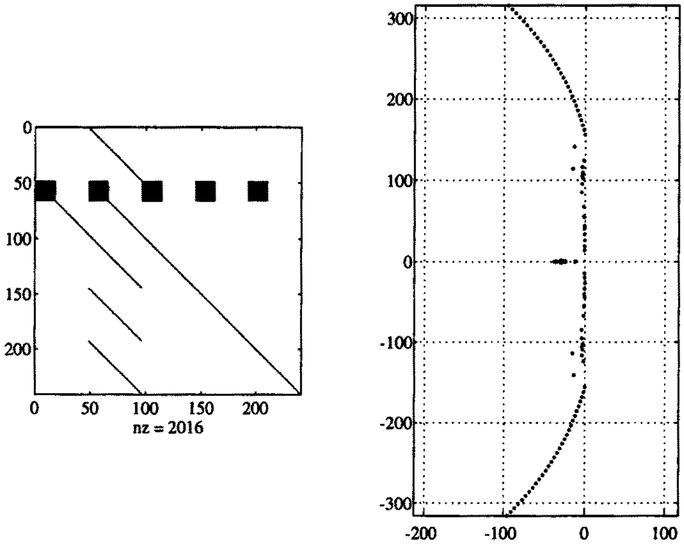


Figure 1. Tolosa matrix $n = 240$. Pattern of nonzero elements, and spectrum.

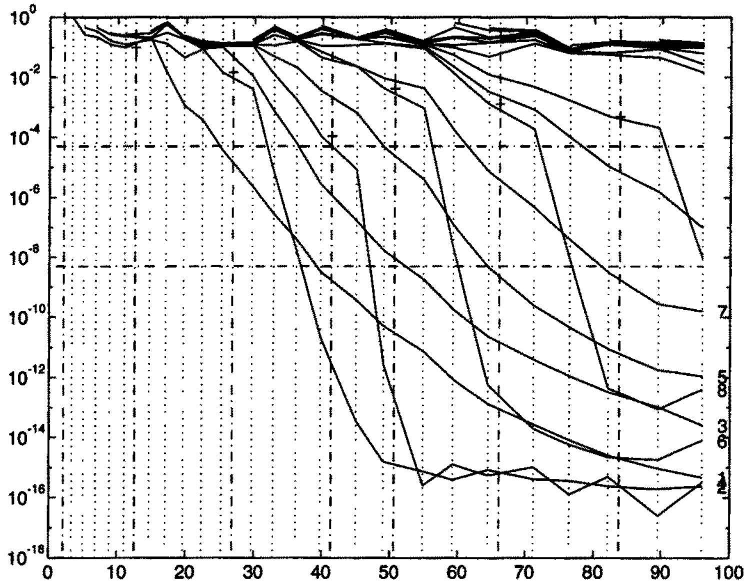


Figure 2. Tolosa matrix $n = 2000$. Follow convergence of Rational Krylov with cpu seconds on SUN4. Complex algorithm.

cputime in seconds. The dotted lines mark the end of each step $j$, and we see that they get successively more time consuming, since both the Gram Schmidt orthogonalization and the QZ-algorithm increase rapidly in cost with $j$. The dashdotted lines

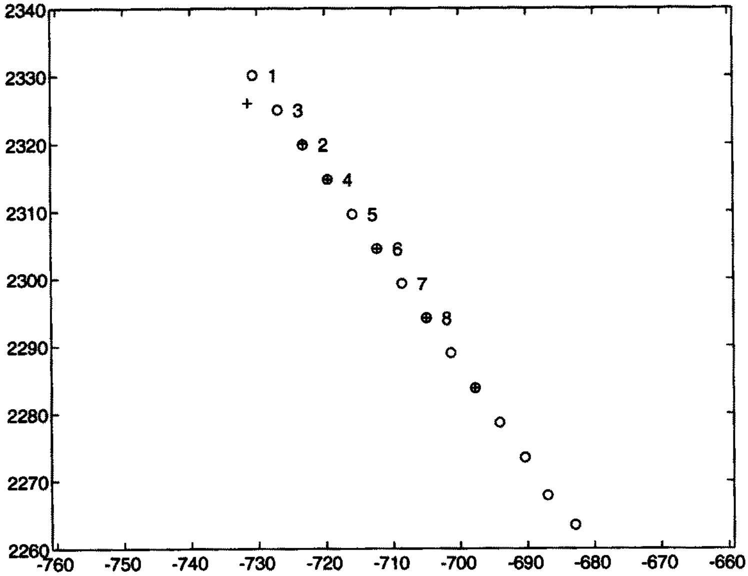


Figure 3. Tolosa matrix $n = 2000$. Eigenvalue approximations o, shifts $+$.

mark factorizations, which are cheap in this case, since the matrix is very sparse. The plus signs mark which of the eigenvalues that is chosen as the new shift. At the 11-th second, at $j = 6$, the best eigenvalue is taken, but next time after 27 seconds, at $j = 11$, we take the second best, since the best has already dropped below *tolshift*, which is marked by the upper dashdotted line. This new shift makes its eigenvalue converge much faster, so that two eigenvalues converge at step $j = 15$. We then choose a new shift at the fourth eigenvalue, which is converged at step $j = 17$ after 50 seconds. It continues in a similar fashion, and when we stop after 97 seconds at $j = 24$, we have flagged 8 eigenvalues as converged, and we plot them in figure 3. Circles mark eigenvalues and plus signs shifts, and note that the later shifts are close indeed. The first shift at *goal* falls outside the plot, as does all but 14 of the approximate eigenvalues. We computed left vectors and measured the reciprocals of the condition numbers $s_i^{-1}$ to around $7.8_{10} - 4$, bad but not very ill conditioned.

We also used the real double shift algorithm of section 4, see figure 4. After 5 factorizations and 15 double steps we got 2 pairs of converged eigenvalues, as well as 4 pairs on their way. Note that we seek eigenvalues in both ends of the spectrum now, and that there are many eigenvalues between the members of the comlex conjugate pairs. This may explain why the complex algorithm works much better for this matrix; it got 8 eigenvalues in the positive imaginary end of the spectrum.

Let us also report some results on the hydrodynamic bifurcation problem that we have used earlier [5, 13, 14]. Here both matrices $A$ and $B$ consists of partial derivatives of the nonlinear function $f(x, \alpha)$, evaluated for different values of the parameter $\alpha$, and the eigenvalues are used to predict for which value of $\alpha$, the derivative matrix

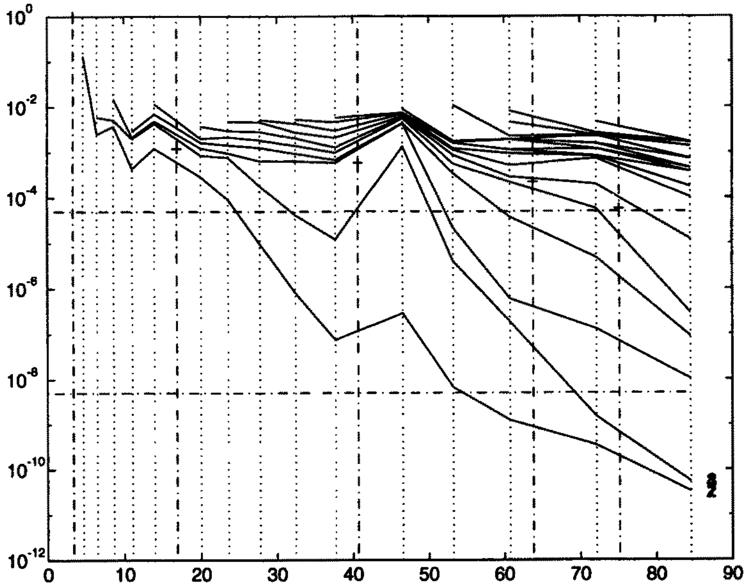


Figure 4. Tolosa matrix $n = 2000$. Real double shift algorithm.

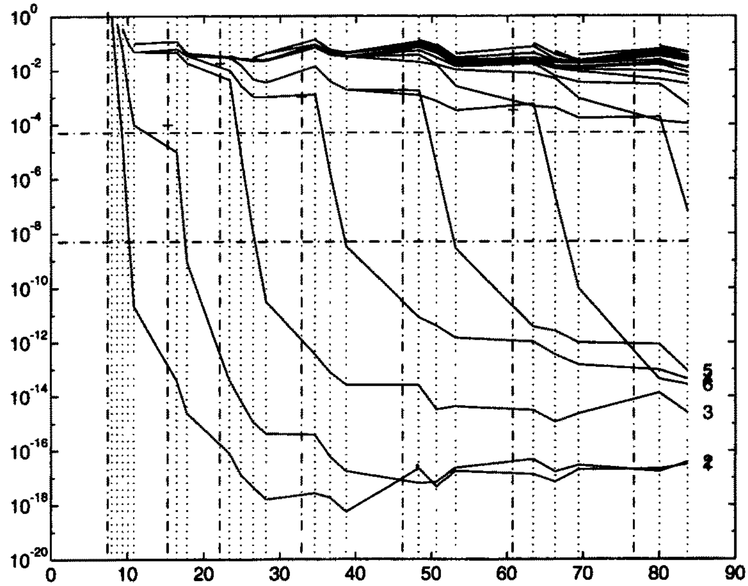


Figure 5. Hydrodynamic matrix, $n = 403$, follow complex RKS.

$f_x = A(\alpha)$ is singular. For $(A, B)$ equal to $(A(\alpha_1))$ we predict the singularity at $\alpha = \alpha_0 + \dfrac{\lambda}{\lambda - 1}(\alpha_1 - \alpha_0)$, so there will be many eigenvalues around $\lambda = 1$, cor-

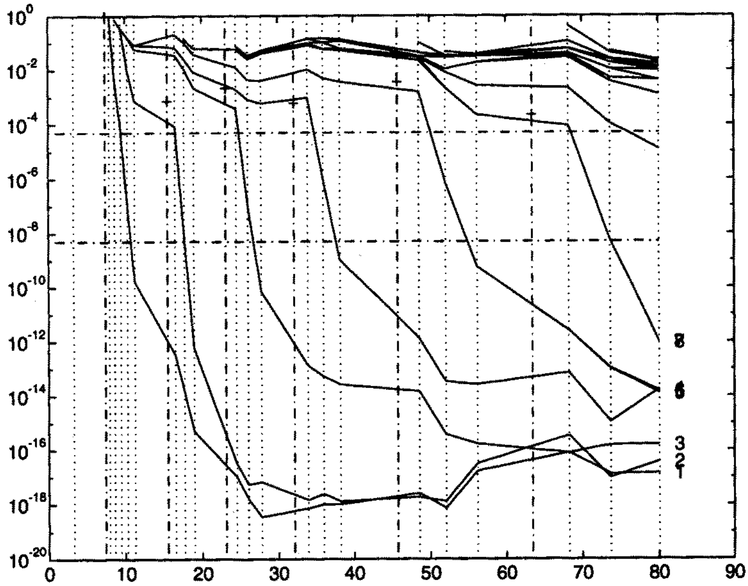


Figure 6. Hydrodynamic matrix, follow real double shift algorithm.

responding to singularities far away from the $(\alpha_0, \alpha_1)$ interval. We are interested in the closest singularities that correspond to large values of $\lambda$.

We have chosen a discretization that gives matrices of order $n = 403$. They are much more filled that in the previous example, so it was necessary to set `spparms ('tight')` in MATLAB to get any sparsity at all in the factored matrices.

First we used the unchanged (complex) version of ALGORITHM RKS. The progress is followed in figure 5. We took *goal* $= 4$ which is rather close to the largest eigenvalue $\lambda = 3.9450$, and it is flagged as converged already at step $j = 6$. The next $\lambda = 2.4803$ is soon to follow at step $j = 8$, and then they follow one eigenvalue every 3 or 4 steps, until we stopped after 22 steps which took 85 seconds.

We plotted the eigenvalue approximations in figure 7, cutting away the outer part containing the outer eigenvalues $\lambda_1$, $\lambda_2$, and $\lambda_3 = 0.31129$. In this small example, we could compute all the eigenvalues with the MATLAB `eig` command, it took 391 seconds, and plotted them as dots. Note that also some of the not yet converged approximations are rather close to the correct eigenvalues, and that the approximations mark up the part of the complex plane where there are eigenvalues. Some of the less accurate approximations are not complex conjugate in pairs. The eigenvalues are slightly better conditioned than the TOLOSA case, we got $s_i^{-1}$ as $1.30_{10} - 3$ and $1.34_{10} - 3$ for the first two eigenvalues $\lambda_1$ and $\lambda_2$.

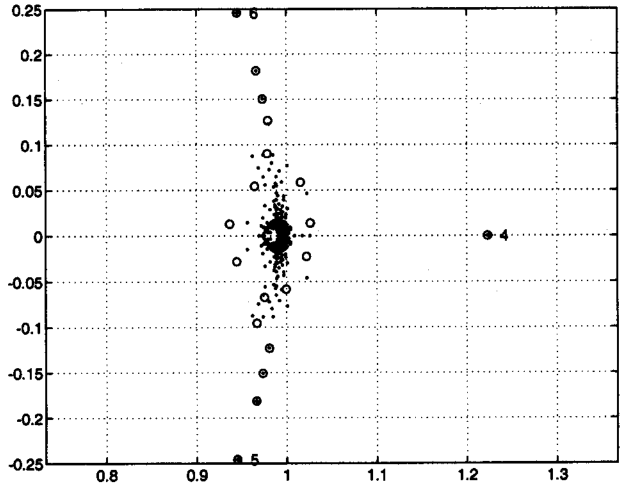


Figure 7. Hydrodynamic matrix, complex algorithm, Ritz apprs.

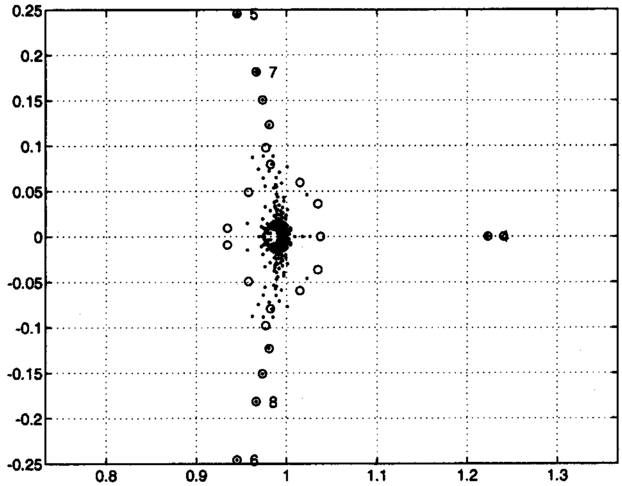


Figure 8. Hydrodynamic matrix, real algorithm, computed approximations.

In figure 6, we follow the progress of the real variant described in section 4. We let it run until $j = 26$ which took 80 seconds. The 4 first shifts were real, and the next 3 shifts were complex. We got convergence to 4 real eigenvalues and two complex pairs, see the perfectly symmetric plot in figure 8! In this case the real algorithm works to advantage, one reason for that may be that work reasonably chose to be the real axis.

## REFERENCES


1. W. Arnoldi, *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, Quart. Appl. Math., 9 (1951), pp. 17–29.
2. F. Chatelin and S. Godet-Thobie, *Stability analysis in aeronautical industries*, in Proceedings of the 2nd Symposium on High-Performance Computing Montpellier, France, M. Durand and F. E. Dabaghi, eds., Elsevier/North-Holland, 1991, pp. 415–422.
3. T. Ericsson and A. Ruhe, *The spectral transformation Lanczos method for the numerical solution of large sparse generalized symmetric eigenvalue problems*, Math. Comp., 35 (1980), pp. 1251–1268.
4. R. G. Grimes, J. G. Lewis, and H. D. Simon, *A shifted block Lanczos algorithm for solving sparse symmetric generalized eigenproblems*, tech. report, Appl. Math. Stat. RTS div. Boeing Comp. Serv., M/S 7L-21, Seattle WA 98124, USA, 1991.
5. J. Huitfeldt and A. Ruhe, *A new algorithm for numerical path following applied to an example from hydrodynamical flow*, SIAM J. Sci. Stat. Comput., 11 (1990), pp. 1181–1192.
6. M. T. Jones and M. L. Patrick, *LANZ: Software for solving the large sparse symmetric generalized eigenproblem*, Tech. Report MCS-P158-0690, Argonne National Lab, Argonne, IL. Also available from netlib., 1990.
7. R. B. Morgan and D. Scott, *Generalizations of Davidson's method for computing eigenvalues of sparse symmetric matrices*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 817–825.
8. B. Parlett and Y. Saad, *Complex shift and invert strategies for real matrices*, Lin. Alg. Appl., 88/89 (1987), pp. 575–595.
9. B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice Hall, 1980.
10. A. Ruhe, *Numerical methods for the solution of large sparse eigenvalue problems*, in Sparse Matrix Techniques, Lect Notes Math 572, V. Barker, ed., Springer Berlin-Heidelberg-New York, 1976, pp. 130–184.
11. A. Ruhe, *The two-sided Arnoldi algorithm for nonsymmetric eigenvalue problems*, in Matrix Pencils, LNM 973, B. Kågström and A. Ruhe, eds., Springer-Verlag, Berlin Heidelberg New York, 1983, pp. 104–120.
12. A. Ruhe, *Rational Krylov sequence methods for eigenvalue computation*, Lin. Alg. Appl., 58 (1984), pp. 391–405.
13. A. Ruhe, *Rational Krylov algorithms for nonsymmetric eigenvalue problems*, in Recent Advances in Iterative Methods, IMA Volumes in Mathematics and its Applications 60, G. Golub, A. Greenbaum, and M. Luskin, eds., Springer-Verlag, New York, 1993, pp. 149–164.
14. A. Ruhe, *Rational Krylov algorithms for nonsymmetric eigenvalue problems, II: Matrix pairs*, Lin. Alg. Appl., 197/198, (1994), pp. 283–296.
15. Y. Saad, *Variations of Arnoldi's method for computing eigenelements of large unsymmetric matrices*, Lin. Alg. Appl., 34 (1980), pp. 269–295.
16. D. S. Scott, *Solving sparse symmetric generalized eigenvalue problems without factorization*, SIAM J. Numer. Anal., 18 (1981), pp. 102–110.
17. D. C. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Anal Appl. 13 (1992), pp. 357–385.
18. G. W. Stewart and J. G. Sun, *Matrix Perturbation Theory*, Academic Press, 1990.
19. D. B. Szyld, *Criteria for combining inverse and Rayleigh quotient iteration*, SIAM J. Numer. Anal., 25 (1988), pp. 1369–1375.