BIT Numerical Mathematics (2006) 46: S99–S111
DOI: 10.1007/s10543-006-0085-9
Published online: 9 September 2006 – © Springer 2006

RATIONAL KRYLOV FOR EIGENVALUE COMPUTATION AND MODEL ORDER REDUCTION*

K. HENRIK A. OLSSON[1] and AXEL RUHE[2,**]

[1] *Saab Automobile AB, Vehicle Integration, SE-461 80 Trollhättan, Sweden.*
email: Henrik.Olsson@se.saab.com

[2] *KTH – Numerical Analysis and Computer Science, SE-100 44 Stockholm, Sweden.*
email: ruhe@nada.kth.se

Dedicated to Björn Engquist on the occasion of his 60th birthday.

Abstract.

A rational Krylov algorithm for eigenvalue computation and model order reduction is described. It is shown how to implement it as a modified shift-and-invert spectral transformation Arnoldi decomposition. It is shown how to do deflation, locking converged eigenvalues and purging irrelevant approximations. Computing reduced order models of linear dynamical systems by moment matching of the transfer function is considered.

Results are reported from one illustrative toy example and one practical example, a linear descriptor system from a computational fluid dynamics application.

AMS subject classification (2000): 65F15, 65F50, 65P99, 93A30.

Key words: eigenvalue computation, moment matching, Krylov sequence, stability.

1 Introduction.

We find Krylov methods to be our methods of choice when it comes to both eigenvalue computations and model order reduction. In this paper we will describe our latest efforts in developing a new flavour of rational Krylov method and go into some of the details necessary to consider in the implementation.

A linear time-invariant single input single output dynamical system in state space form

$$\begin{aligned} E\dot{x}(t) &= Ax(t) + bu(t), \\ y(t) &= cx(t), \end{aligned} \tag{1.1}$$

* Received December 1, 2005. Accepted April 11, 2006. Communicated by Anna-Karin Tornberg.
** Financial support from the Swedish Science Research Council, Vetenskapsrådet, contract 2003-5431.

connects input $u(t)$ and output $y(t)$ via the n-dimensional state $x(t)$. Here A and E are $n \times n$ matrices, b an n-dimensional column vector and c a row vector. Apply Laplace transformation, assuming zero initial conditions, and get

$$Y(s) = c(sE - A)^{-1}bU(s) \equiv H(s)U(s). \tag{1.2}$$

The transfer function $H(s)$ gives the output $Y(s)$ as a function of the input $U(s)$ at the complex frequency s.

The poles of the transfer function $H(s)$ are eigenvalues of the linear matrix pencil

$$(A - \lambda E)x = 0. \tag{1.3}$$

Any of the matrices A and E may be singular, but the pencil (1.3) is assumed to be regular.

We will study algorithms for the eigenvalue problem (1.3) with a view at getting a reduced order model for the system (1.1) that gives a good approximation to the transfer function $H(s)$ in a relevant range of frequencies s (1.2).

A detailed report of this study, that also includes an algorithm description, can be found in the report [6].

1.1 Rational Krylov.

We separate the algorithm development into three aspects: spectral transformation, subspace expansion, and solution of the projected problem.

A *spectral transformation* is a simple mapping $\theta(\lambda)$ of the complex plane into itself. It corresponds to a matrix function $C(A, E)$ of the matrix pencil (1.3). The eigenvalues $\theta_i = \lambda(C)$ are $\theta_i = \theta(\lambda_i)$, where λ_i are the eigenvalues of the pencil (1.3).

We consider a series of Krylov subspaces of the matrix C starting at a vector x,

$$\mathcal{K}_j(C, x) \equiv \operatorname{span}\{x, Cx, \ldots, C^{j-1}x\}. \tag{1.4}$$

Vectors y in the Krylov space are generated by successive applications of the matrix C to a vector, $x_{k+1} = Cx_k$, and can be expressed as polynomials of C acting at the starting vector, $y = p_{j-1}(C)x$.

The Arnoldi algorithm finds an orthonormal basis V_j of the Krylov subspace (1.4)

$$CV_j = V_{j+1}H_{j+1,j}, \quad \text{where } v_1 = x/\|x\|_2, \text{ and } H_{j,j} = V_j^H C V_j. \tag{1.5}$$

We use subscripts to indicate the size of submatrices, V_j has j columns and $H_{j+1,j}$ is of size $(j+1) \times j$. The Hessenberg matrix $H_{j,j}$ is the projection of C onto the Krylov space spanned by V_j. It is built up by the Gram–Schmidt orthogonalisation coefficients. The eigenvalues of $H_{j,j}$ approximate the eigenvalues of C, best for those at isolated positions at the extremes of the spectrum of C.

The unit spectral transformation, $\theta = \lambda$, $C = E^{-1}A$, will give the extreme eigenvalues of the pencil (1.3), while the shift-and-invert spectral transformation

with $\theta = 1/(\lambda - \sigma)$, $C = (A - \sigma E)^{-1}E$, will look for eigenvalues close to the shift σ.

In rational Krylov [9] [1, Section 8.5], the shifts σ may be chosen at new values in each step j. Each time the shift σ_j is changed, a new sparse LU-factorisation of the shifted matrix is needed, so σ_j is kept constant for an appropriate number of steps. When we change σ_j, we need to modify the simple Arnoldi decomposition (1.5).

After a number of steps j, we decide to test for convergence and deflate. We choose to do this by computing a Schur decomposition of the Hessenberg matrix $H_{j,j}Q_j = Q_jT_{j,j}$. The residuals of the respective Schur vectors q_i are proportional to the absolute values of the elements in the last row of the unitary matrix Q.

We reorder the Ritz values at the diagonal of T by a sequence of unitary similarities; see [7, 4, 2]. After reordering the Ritz values, we can divide them into three sets, the first consisting of converged values, corresponding to small $|q_{j,i}|$, the second set is those that we want to use to continue the iteration to find more eigenvalues, and the third and last those that we want to discard or purge. This is a variant of implicit restart of Sorensen [10], with the deflation as [8] and the Krylov–Schur algorithm of Stewart [11].

1.2 Model order reduction.

We are interested in the values of the transfer function $H(s)$ (1.2) over a practically relevant range of frequencies s. It is computed by solving a linear system with a shifted matrix $sE - A$ and right hand side b. It is natural to start a rational Krylov sequence at $x = (A - \sigma_1 E)^{-1}b$ and then choose shifts σ_j at some positions in the interesting region, in most cases along an interval on the imaginary axis. That way we may match moments, that is interpolate the transfer function, at the shift points σ_j. Repeated use of the same shift leads to interpolation of higher order at that point, as explained by Gallivan, Grimme and Van Dooren [3].

We may regard this as replacing the full transfer function (1.2) by another rational function with poles at the eigenvalues of the reduced pencil.

2 Rational Krylov.

2.1 Arnoldi algorithm.

The rational Krylov algorithm applied to the eigenvalue problem (1.3) is built up around a shift invert Arnoldi decomposition (1.5),

$$(A - \sigma E)^{-1}EV_j = V_{j+1}H_{j+1,j}, \tag{2.1}$$

where the shift σ is changed in some of the steps j. When we change the shift, we update both the orthonormal basis V and the Hessenberg (trapezoidal) matrix H. When some eigenvalues have converged, we may deflate the decomposition (2.1) and continue with a smaller updated basis V.

2.2 Test for convergence.

Test for convergence is done at step j by computing a Schur factorization of the Hessenberg matrix,

$$H_{j+1,j} = \begin{pmatrix} U_j & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} T_{j,j} \\ t_{j+1} \end{pmatrix} U_j^H, \tag{2.2}$$

with the $j \times j$ triangular $T_{j,j}$ and unitary matrix U_j of Schur vectors. The last row is t_{j+1} is proportional to the last row of the unitary matrix U_j of Schur vectors.

An eigenpair (θ_i, y_i) of $T_{j,j}$, $Ty_i = y_i\theta_i$, will correspond to a harmonic Ritz value

$$\lambda_i^{(j)} = \sigma_j + 1/\theta_i \tag{2.3}$$

and Ritz vector $x_i = VUy_i$ of the pencil $(A - \lambda E)$ (1.3). The residual of this Ritz vector is

$$\big(A - \lambda_i^{(j)} E\big)x_i = \theta_i^{-1}(A - \sigma_j E)v_{j+1}t_{j+1}y_i.$$

Its norm will be small for large θ_i, corresponding to small leading elements of the last row t_{j+1} of the Schur decomposition (2.2). Flag those eigenvalues as converged.

2.3 Change shift.

The eigenvalues closest to the shift σ_j will be the first to converge. Continue with the same shift, until the basis size j is so large that the orthogonalization in Arnoldi will be heavy computationally, or the convergence of new eigenvalues gets too slow. Then change to a new shift σ_{j+1} and find eigenvalues close to that new shift. In model reduction contexts, the shifts are placed along an interval of the imaginary axis. In eigenvalue computation, the new shift σ_{j+1} is chosen in the midpoint between two nonconverged Ritz values. This will give the new shifted and inverted pencil two extreme eigenvalues opposite to each other, likely to converge fast.

Reorder the eigenvalues $\theta_i = t_{i,i}$ by means of unitary rotations [7] in order of increasing distance of the corresponding Ritz values $\lambda_i^{(j)}$ (2.3) to the new shift σ_{j+1} getting

$$W_j^H T_{j,j} W_j = \begin{pmatrix} T_{k,k} & T_{k,j-k} \\ 0 & T_{j-k,j-k} \end{pmatrix}.$$

The last row t_{j+1} will also be affected by these rotations and changed into $t_{j+1}W_j$.

Purge the last part of the reordered triangular matrix, keeping those k eigenvalues θ_i that correspond to Ritz values close to the new shift and inside the region of interest. Update the basis V_j with the unitary transformation to the reordered Schur form (2.2) and discard the last columns that correspond to purged

eigenvalues θ_i. Keep the last basis vector v_{j+1} and get back the decomposition (2.1), now with a smaller augmented triangular

$$H_{j+1,j} = \begin{pmatrix} T_{j,j} \\ t_{j+1} \end{pmatrix} \tag{2.4}$$

with a smaller j equal to k the number of kept vectors. The last row is the first k elements of $t_{j+1}W_j$.

2.4 Update Arnoldi decomposition.

The Arnoldi decomposition (2.1) is updated to one with a new shift σ_{j+1} in the following way. This is done without multiplying with any factored matrix or adding vectors to the basis.

Rewrite (2.1) by premultiplying with the shifted matrix $A - \sigma_j E$,

$$\begin{aligned} EV_j &= AV_{j+1}H_{j+1,j} - \sigma_j EV_{j+1}H_{j+1,j} \\ &= (A - \sigma_{j+1}E)V_{j+1}H_{j+1,j} - (\sigma_j - \sigma_{j+1})EV_{j+1}H_{j+1,j} \end{aligned} \tag{2.5}$$

and premultiply again, now with $(A - \sigma_{j+1}E)^{-1}$,

$$(A - \sigma_{j+1}E)^{-1}EV_{j+1}(I_{j+1,j} + (\sigma_j - \sigma_{j+1})H_{j+1,j}) = V_{j+1}H_{j+1,j} \tag{2.6}$$

with $I_{j+1,j}$ the first j columns of the $j+1$-dimensional unit matrix. Do a QR factorization of the last factor on the left hand side

$$I_{j+1,j} + (\sigma_j - \sigma_{j+1})H_{j+1,j} = Q_{j+1}R_{j+1,j} \tag{2.7}$$

where Q_{j+1} is a product of j elementary Givens rotations

$$Q_{j+1} = G_{1,j+1}G_{2,j+1}\dots G_{j,j+1} \tag{2.8}$$

eliminating the elements of the last row of this augmented triangular matrix one by one, and $R_{j+1,j}$ will get a zero last row. Note that the first j columns $Q_{j+1,j}$ of the orthogonal matrix Q_{j+1} will have the same pattern of nonzero elements as $H_{j+1,j}$ (2.4).

Multiply from the right with the inverse triangular $R_{j,j}^{-1}$ and get,

$$\begin{aligned} (A - \sigma_{j+1}E)^{-1}EV_{j+1}Q_{j+1,j} &= V_{j+1}H_{j+1,j}R_{j,j}^{-1} \\ &= V_{j+1}Q_{j+1}Q_{j+1}^H H_{j+1,j}R_{j,j}^{-1} \\ &= V_{j+1}Q_{j+1}M_{j+1,j}. \end{aligned}$$

The resulting rectangular matrix $M_{j+1,j}$ is full, but is restored to Hessenberg form by applying the Householder algorithm backwards [8],

$$M_{j+1,j} = \begin{pmatrix} Z_j & 0 \\ 0 & 1 \end{pmatrix} \widetilde{H}_{j+1,j}Z_j^H$$

where the unitary transformation Z_j is a product of j elementary reflections and $\widetilde{H}_{j+1,j}$ is of Hessenberg form.

The new Arnoldi recursion,

$$(A - \sigma_{j+1}E)^{-1}E\widetilde{V}_j = \widetilde{V}_{j+1}\widetilde{H}_{j+1,j} \tag{2.9}$$

has a new basis

$$\widetilde{V}_{j+1} = V_{j+1}Q_{j+1}\begin{pmatrix} Z_j & 0 \\ 0 & 1 \end{pmatrix} \tag{2.10}$$

and Hessenberg matrix

$$\widetilde{H}_{j+1,j} = \begin{pmatrix} Z_j^H & 0 \\ 0 & 1 \end{pmatrix} Q_{j+1}^H H_{j+1,j} R_{j,j}^{-1} Z_j.$$

Note that this new basis $\widetilde{V}_{j+1}$ spans the same Krylov space as V_{j+1} but that the new Arnoldi recursion (2.9) has its starting vector v_1 changed to another vector $\tilde{v}_1$ in the same subspace.

If there are some leading zero elements of the last row t_{j+1}, the corresponding converged diagonal elements will still be there, the rest of the changed $\widetilde{H}_{j+1,j}$ will be unreduced.

The eigenvalues will be changed. Only in the case of convergence will $\tilde{\theta}_i = \frac{1}{\lambda-\sigma_{j+1}} = \frac{1}{(\lambda-\sigma_j)}\frac{\lambda-\sigma_j}{\lambda-\sigma_{j+1}} = \theta_i\frac{\lambda-\sigma_j}{\lambda-\sigma_{j+1}}$. For other Ritz values, this relation will hold approximately. Those θ_i that correspond to eigenvalues λ_i closer to the new shift σ_{j+1} than to the old shift σ_j will be larger. However no Ritz value, that is not converged at step j, will be labeled as converged in the new recursion.

After the shift change, those Ritz values that correspond to eigenvalues close to the new shift σ_{j+1} will be the first to converge.

3 Model order reduction.

3.1 Approximate solution.

When using Rational Krylov to find a reduced order model of the linear time invariant system (1.1) replace the transfer function (1.2) by

$$\hat{H}(s) = c\hat{X}(s)$$

where $\hat{X}$ is an approximate solution to the system

$$(A - sE)x = b.$$

Do a series of rational Krylov steps starting at

$$\beta_1 v_1 = (A - \sigma_1 E)^{-1}b$$

where β_1 is a normalization factor. Use appropriately chosen shifts σ_j.

When we are finished at step j, formulate the recursion as

$$(A - \sigma_1 E)^{-1} E V_j = V_{j+1} H_{j+1,j}$$

with the first shift σ_1 but with an updated basis V_{j+1}. The approximate solution is

$$\hat{X}(s) = V_j y$$

where y is

$$y = \beta_1 K_{j,j}^{-1} z^{(j)} \quad \text{with } K_{j+1,j} = I_{j+1,j} - (s - \sigma_1) H_{j+1,j}$$

with $z^{(j)}$ is a unit length vector obtained by letting the basis changes (2.10) affect the first basis vector e^1.

3.2 Error estimate.

The error in the transfer function is

$$\begin{aligned}
e(s) &= c(\widehat{X}(s) - X(s)) \\
&= c(V_j y - (A - sE)^{-1} b) \\
&= c(A - sE)^{-1} \big\{ (A - sE) V_j y - b \big\} \\
&= c(A - sE)^{-1} (A - \sigma_1 E) \big\{ (A - \sigma_1 E)^{-1} (A - sE) V_j y - (A - \sigma_1 E)^{-1} b \big\} \\
&= c(A - sE)^{-1} (A - \sigma_1 E) \\
&\quad \times \big\{ (A - \sigma_1 E)^{-1} (A - \sigma_1 E - (s - \sigma_1) E) V_j y - V_j z^{(j)} \beta_1 \big\} \\
&= c(A - sE)^{-1} (A - \sigma_1 E) V_{j+1} \big\{ (I_{j+1,j} - (s - \sigma_1) H_{j+1,j}) y - z^{(j)} \beta_1 \big\} \\
&= c(A - sE)^{-1} (A - \sigma_1 E) v_{j+1} \big\{ - (s - \sigma_1) h_{j+1,j} e_j^T y \big\}.
\end{aligned}$$

This means that we can split the error into two factors, $e(s) = e_1(s) e_2(s)$, where

$$e_1(s) \equiv c(A - sE)^{-1} (A - \sigma_1 E) v_{j+1} \tag{3.1}$$

involves the large matrices and

$$e_2(s) \equiv -(s - \sigma_1) h_{j+1,j} e_j^T y \tag{3.2}$$

only contains quantities from the small system. It is cheap to compute for several values of s.

This kind of splitting of the error was done for the rational Lanczos method in [3]. In our experiments, we use the factor $e_2(s)$ as an error estimate for deciding where to put the next shift.

4 Numerical results.

We apply our algorithm to one illustrative toy example and one test problems from the Oberwolfach Model Reduction Benchmark Collection [5]. All computations are made in MATLAB 7, with the usual double precision where the floating point relative accuracy is 2.22×10^{-16}. This was run under Linux on a PC with 1 GB memory and a Pentium 4 3.00 GHz processor with 1024 kB cache.

4.1 Mass-Spring-Damper system.

To clearly see some properties of mechanical systems and our algorithms, we want a small system where we can change parameters at will and analyse the effects.

We study a mass-spring-damper system described in [1], where n masses are connected by springs and dampers in a one dimensional array. This gives a second order system

$$\begin{aligned} M\ddot{x}(t) + D\dot{x}(t) + Kx(t) &= bu(t), \\ y(t) &= cx(t), \end{aligned}$$

where x_j is the position of the jth mass, M is diagonal containing mass values, D is diagonal containing damping constants, and K is tridiagonal constructed from spring constants. We choose $b = e_{n-1}$ and $c = e_2^H$, meaning that we apply a force to the penultimate mass and observe the displacement of the second mass.

With $n = 20$ masses, all of equal unit value, and all spring constants equal to one fourth, we choose all the damping constants to be 0.01 to get a lightly damped system. Note that we get a Rayleigh damping, with $D = 0.01M$. We see the spectrum in Figure 4.1, where it can be compared with that for another Rayleigh damping with $D = 0.01(K + M)$. In the latter case, we see the typical effect of increased damping for higher frequencies.

We know that the interesting frequencies are in the interval $(0, 1)$. Using our algorithm on a linearised system of the form (1.1) of size $N = 40$, generating a few vectors with the shift $\sigma = 0.8i$ and then moving to $\sigma = 0.2i$ for another few vectors, we get a reduced order model matching moments in each of these shifts. After using these two shifts – in the upper and lower part of the frequency

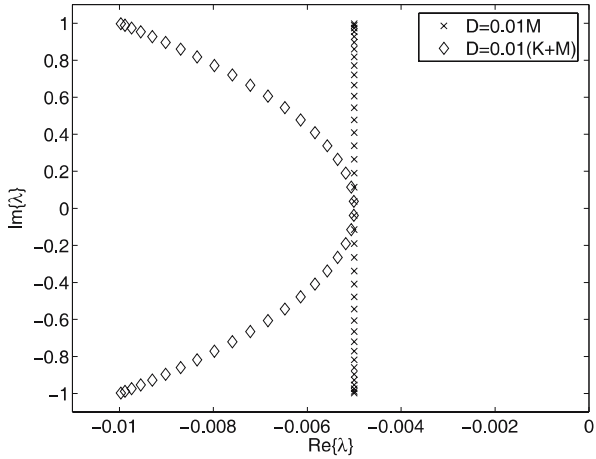
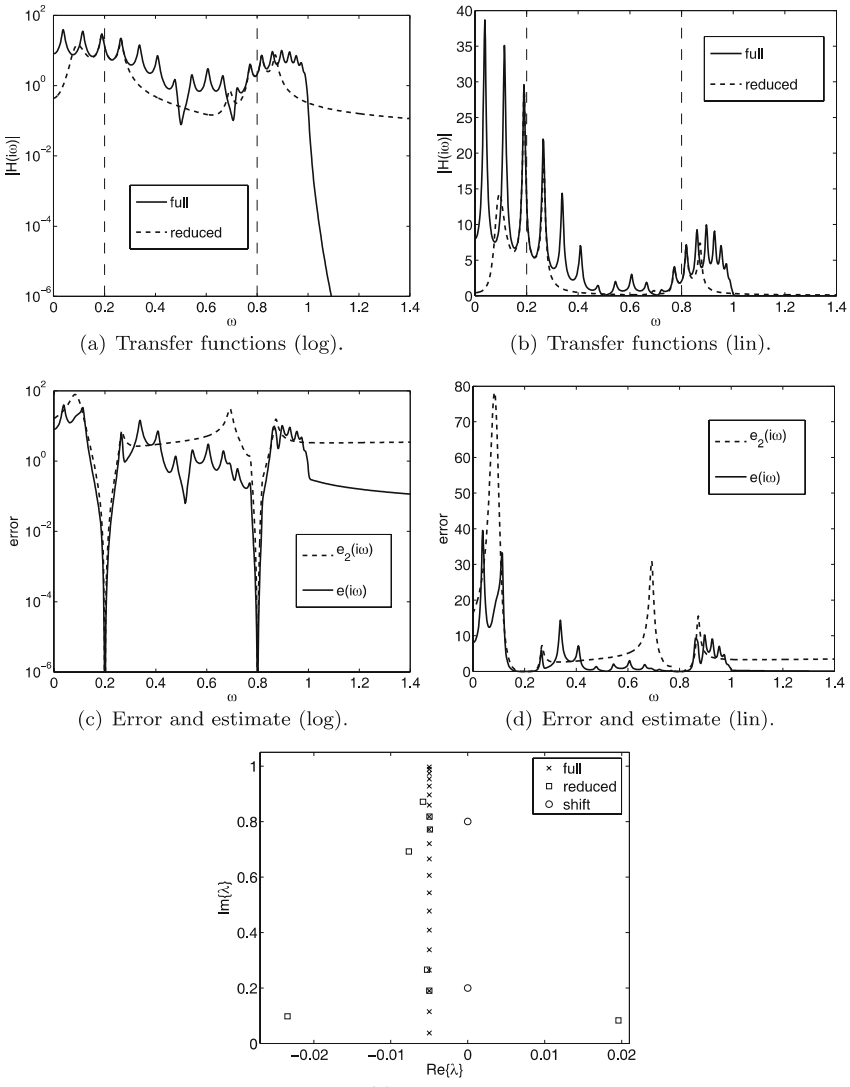


Figure 4.1: Eigenvalues of the mass-spring-damper system, for both the given damping and another Rayleigh damping. Note the different scales for the real and imaginary axes.

interval, respectively – we would like to use the error estimate derived in Section 3.2 for finding where to put the next shift. We see the transfer functions and eigenvalues of the full and reduced order systems in Figure 4.2.

We see that we get moment matching as expected, and that the error estimate $e_2(i\omega)$ indicates large errors when an eigenvalue of the reduced system is on its



(a) Transfer functions (log).

(b) Transfer functions (lin).

(c) Error and estimate (log).

(d) Error and estimate (lin).

(e) Eigenvalues.

Figure 4.2: Mass-spring-damper system: Transfer functions for full and reduced order model, error e and error estimate e_2 for the transfer function matching, and eigenvalues of full and reduced order model. Reduced order model of size 8 from matching 4 moments at each of the shifts $\sigma = 0.2i$ and $\sigma = 0.8i$.

way towards the corresponding eigenvalue of the full system. We also see that the reduced order model of size 8 is unstable, this is a similar situation as when we get spurious Ritz values in the Lanczos algorithm, see [1].

We remark that this is a simple example, in the sense that the full order system is of low order, but a rather difficult one, in the sense that it has a large number of lightly damped poles.

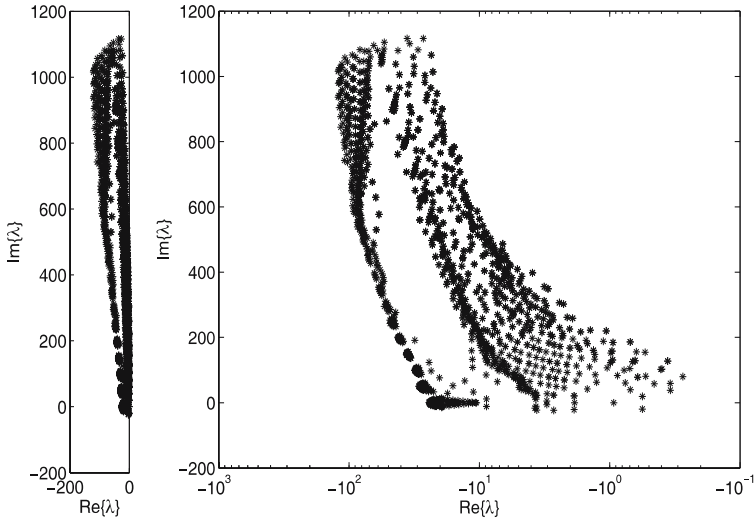
4.2 Supersonic engine inlet.

The second system we study here comes from *Active Control of a Supersonic Engine Inlet* by Willcox, Lassaux, and Gratton [5]. It is a descriptor system

$$\begin{aligned} E\dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t), \end{aligned}$$

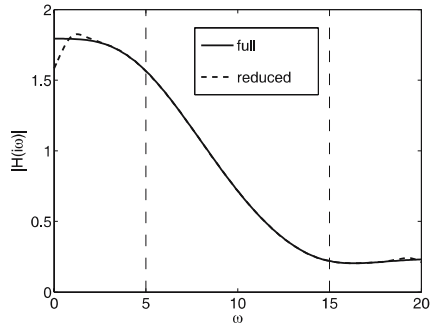
of size $N = 11730$ coming from a linearisation of two-dimensional Euler equations describing unsteady flow through a supersonic diffuser. The descriptor matrix E is singular. There are two inputs and one output, which means that $B \in \mathbb{R}^{N\times 2}$ and $C \in \mathbb{R}^{1\times N}$. This is an interesting test problem to us, since it is reasonably large and has quite a lot of eigenvalues along the imaginary axis; the transfer function is also rather smooth in the frequency region said to be relevant [5]. Some of the eigenvalues in the second quadrant are seen in Figure 4.3.

Looking for a reduced order model that is a good approximation in the frequency range $(0, 20)$, we generate six vectors at each of the shifts $\sigma_1 = 5i$ and $\sigma_2 = 15i$. The resulting model of size 12 has the approximation properties seen in Figure 4.4.

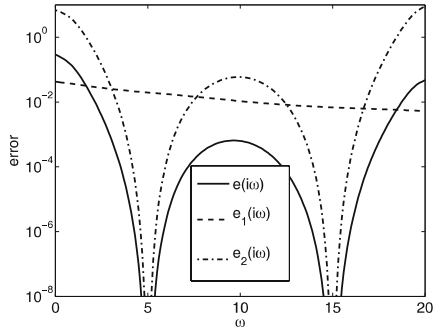


(a) Equal axes. (b) Logarithmic x-axis.

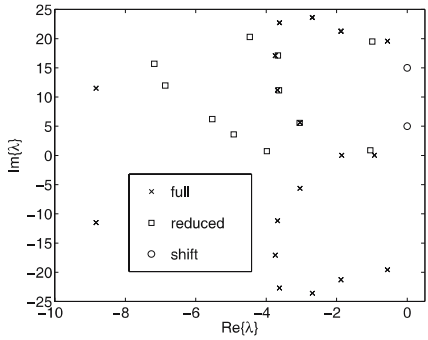
Figure 4.3: Supersonic engine inlet: Some eigenvalues.



(a) Transfer functions.



(b) Error and estimate.



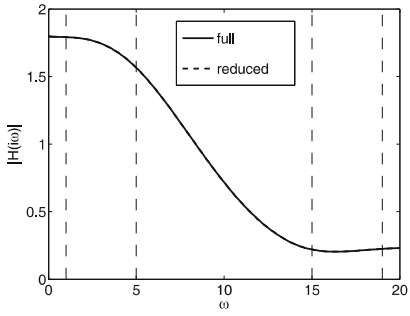
(c) Eigenvalues.

Figure 4.4: Supersonic engine inlet: Transfer functions for full and reduced order model, error e and error estimate e_2 for the transfer function matching, and eigenvalues of full and reduced order model. Reduced order model of size 12 from matching 6 moments at each of the two shifts $5i$ and $15i$.

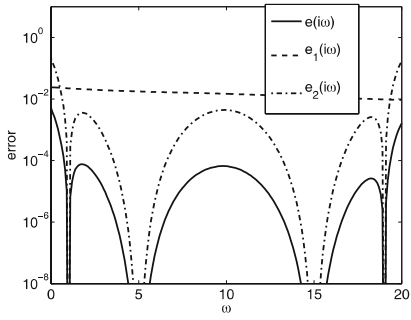
We see that the error estimate $e_2(i\omega)$ (3.2) is useful here – indicating that the error is larger at the ends of the interval – and that the other error factor $e_1(i\omega)$ (3.1) is slowly varying. Adding two vectors at each of the shifts $\sigma_3 = 19i$

and $\sigma_4 = i$ to decrease the error at the ends we get the approximation seen in Figure 4.5.

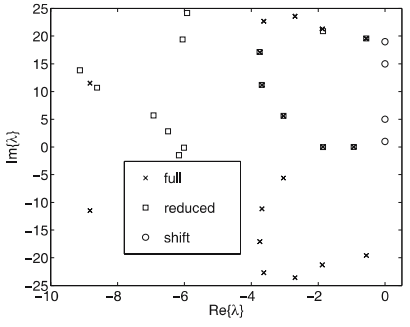
For computing eigenvalues close to the imaginary axis, we start from $\sigma = 10i$ and move upward. Computing a hundred eigenvalues with an active recursion of size $J = 50$, changing shifts when 10 eigenvalues have converged and keeping 5



(a) Transfer functions.



(b) Error and estimate.



(c) Eigenvalues.

Figure 4.5: Supersonic engine inlet: Transfer functions for full and reduced order model, error e and error estimate e_2 for the transfer function matching, and eigenvalues of full and reduced order model. Reduced order model of size 16 from matching 6 moments at $5i$ and $15i$ and 2 moments at i and $19i$.

unconverged vectors when purging, we get the result seen in Figure 4.6. In the shift changes in this computation, the norm of the inverse of R was never larger than 14.

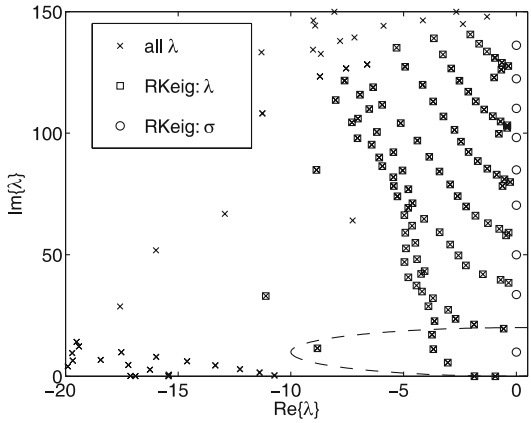


Figure 4.6: Supersonic engine inlet: Some eigenvalues computed with Algorithm RKeig, compared with the spectrum found with eigs. Also included is a (half-)circle with radius 10 centered at $10i$ for indicating the axis scales and the shape of a typical convergence region of a shift-and-invert method.

REFERENCES

1. Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, eds., *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, SIAM, Philadelphia, 2000.
2. D. Bindel, J. Demmel, W. Kahan, and O. Marques, *On computing Givens rotations reliably and efficiently*, ACM Trans. Math. Softw., 28 (2002), pp. 206–238.
3. K. Gallivan, E. Grimme, and P. V. Dooren, *A rational Lanczos algorithm for model reduction*, Numer. Algorithms, 12 (1995), pp. 33–63.
4. G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, Maryland, 3rd edn., 1996.
5. *Oberwolfach model reduction benchmark collection*, `http://www.imtek.uni-freiburg.de/simulation/benchmark/`.
6. K. H. A. Olsson and A. Ruhe, *Rational Krylov for model order reduction and eigenvalue computation*, technical report TRITA-NA-0520, NADA, Royal Institute of Technology, Stockholm, 2005.
7. A. Ruhe, *An algorithm for numerical determination of the structure of a general matrix*, BIT, 10 (1970), pp. 196–216.
8. A. Ruhe, *Eigenvalue algorithms with several factorizations – a unified theory yet?*, technical report 1998:11, Department of Mathematics, Chalmers University of Technology, Göteborg, Sweden, 1998.
9. A. Ruhe, *Rational Krylov, a practical algorithm for large sparse nonsymmetric matrix pencils*, SIAM J. Sci. Comput., 19 (1998), pp. 1535–1551.
10. D. C. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 357–385.
11. G. W. Stewart, *A Krylov–Schur algorithm for large eigenproblems*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 601–614.