Записки научных
семинаров ПОМИ
Том 268, 2000 г.

Axel Ruhe

A RATIONAL KRYLOV ALGORITHM FOR NONLINEAR MATRIX EIGENVALUE PROBLEMS

Abstract. It is shown how the rational Krylov algorithm can be applied to a matrix eigenvalue problem that is nonlinear in the eigenvalue parameter.

Dedicated to Vera Nikolaevna Kublanovskaya on the occasion of her 80th birthday

1. Introduction and background

In the present contribution we will describe an algorithm for the numerical computation of solutions to the eigenvalue problem

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

linear in the vector x but nonlinear in the eigenvalue parameter λ. Typical applications are mechanical systems with friction, viscous damping or nonlinear materials.

The Leningrad group with Faddeeva and Kublanovskaya did devote quite some effort to these problems back in 1969, when I was visiting at LOMI. In [3, 4] Kublanovskaya described an ingenious way to use QR factorizations in a Newton type algorithm to find singularities of $A(\lambda)$. It inspired me to write [5] where an algorithm of successive linear problems is developed as a closely related Newton approach.

Now we are interested in large matrices, where it is possible but costly to do a Gaussian elimination factorization and even more a QR factorization. We are then confined to using a fixed factorization point (pole) σ but evaluate $A(\mu)x$ for many different values (shifts) μ and vectors x. This way we get a kind of secant, or more properly *Regula falsi* iteration, which we combine with rational Krylov (see [6, 1, Section 8.5]) to get an expanding subspace from which we find eigenvectors.

For further numerical results see [2] and the forthcoming thesis of Patrik Hager.

2. Predicting singularities

We seek values of λ where $A(\lambda)$ is singular. Take two points σ and μ and use the Lagrange interpolation formula,

$$A(\lambda) = \frac{\lambda-\mu}{\sigma-\mu}A(\sigma) + \frac{\lambda-\sigma}{\mu-\sigma}A(\mu) + (\lambda-\sigma)(\lambda-\mu)R(\lambda). \tag{2}$$

Use the terms linear in λ to predict a singularity of $A(\lambda)$ at a solution of the eigenvalue problem,

$$[(\lambda-\mu)A(\sigma) - (\lambda-\sigma)A(\mu)]\, w = 0, \tag{3}$$

$$A(\sigma)^{-1}A(\mu)w = w\theta, \quad \text{with} \quad \theta = (\lambda-\mu)/(\lambda-\sigma),$$

which predicts singularity at $\lambda = \mu + \frac{\theta}{1-\theta}(\mu-\sigma)$. The extreme eigenvalues θ correspond to the interesting singularities that are close to the pole σ. There will be many eigenvalues close to $\theta = 1$ corresponding to λ values far away, while the singularities close to the shift μ will correspond to θ close to zero.

Use the Arnoldi algorithm on the generalized eigenvalue problem (3), with a Gaussian elimination LU-factorization of $A(\sigma)$ regarded as a preconditioner,

$$A(\sigma)^{-1}A(\mu)V_j = V_j H_{j,j} + R_j, \tag{4}$$

where V_j is the computed orthogonal basis with j columns, $H_{j,j}$ is a $j \times j$ Hessenberg matrix and the $n \times j$ residual matrix has only its last column filled with a multiple of the next basis vector v_{j+1}.

In the linear case any choice of the shift μ is equally good, and most often we choose a simple shift invert variant which corresponds to μ at infinity. In the nonlinear case $A(\mu)$ will vary, and it is natural to use a rational Krylov approach, where μ is updated during the recursion. We may formulate this as an updating Arnoldi algorithm in the same way, as described in Eigentemplates [1, Section 8.5], and get the following algorithm:

Algorithm NLRKS

1. Start with pole σ, approximation μ, starting vector v_1, $j = 1$.
2. Set $h_j = 0_j$; $s = e_j$; $x = v_j$.
3. Compute $r = A(\sigma)^{-1}A(\mu)x$ (Operate).
4. Compute $k_j = V_j^* r$, (Gram Schmidt).
5. If $\|k_j\|_2 < toln$, go to step 13! (residual orthogonal).
6. Orthogonalize $r = r - V_j k_j$.

7. Accumulate $h_j = h_j + k_j s_j^{-1}$.
8. Compute eigensystem $H_{j,j}S = S\,\mathrm{diag}\,(\theta_i)$.
9. Choose $\theta = \theta_i$ close to 0, $s = s_i$, (Ritz value correction).
10. Compute $x = V_j s$, (Ritz vector)
11. Update $\mu := \mu + \frac{\theta}{1-\theta}(\mu - \sigma) = \frac{1}{1-\theta}\mu - \frac{\theta}{1-\theta}\sigma$.
12. Update $H_{j,j} := \frac{1}{1-\theta}H_{j,j} - \frac{\theta}{1-\theta}I$, go to step 3!
13. Compute $h_{j+1,j} = \|r\|_2$.
14. $|h_{j+1,j}s_j| > tolc$, set $v_{j+1} = r/h_{j+1,j}$; $j = j+1$, go to step 2!
15. Else return eigenvalue $\lambda_i = \mu$ and eigenvector $x_i = x$.
16. Lock and purge $H_{j,j}$, decrease j and update V_j
17. If more eigenvalues wanted, choose next $\theta = \theta_i$; $s = s_i$, go to step 10!

END

The logic of this algorithm is not entirely simple, it is two iterations knit together. One is a Krylov iteration, steps 2 to 14, steered by the counter j, which stands for the dimension of the Krylov basis. It grows one dimension at a time, but may shrink when locking and purging is performed. The other iteration is a nonlinear secant type iteration in steps 3 to 12, which makes sure that the residual of the nonlinear problem (1) is orthogonal to the Krylov space spanned by the basis V_j. The last steps, 15 to 17, return converged eigenvalues, purges uninteresting directions from the basis, decreasing the j counter, and starts iteration for another eigenvalue if that is asked for.

Let us comment on details step by step:

Step 1: The pole σ is where the matrix is factorized, and should be chosen as close as possible, but not exactly on the eigenvalues wanted. The shift μ is a guess on the first eigenvalue to be computed.

Step 2: The vector h_j is the last column of the Hessenberg matrix $H_{j,j}$, it is initialized as a zero j dimensional vector.

Step 3: This is the heavy work. We save a Gaussian elimination sparse LU factorization of $A(\sigma)$ on start, and use it to operate here. Note that μ, on the other hand, is changed each time we come to this step.

Step 5: Note that we start a new sequence of updates of H and μ, without changing the basis size j, until we have made sure that the residual for the new eigenvalue approximation is orthogonal to the basis V_j. For a linear eigenproblem this should always happen the second time around. Possibly a larger value of the orthogonality tolerance *toln* could be used to advantage in early iterations.

Step 7: This is not a simple Gram Schmidt reorthogonalization. The formula supposes that s is an eigenvector to a zero eigenvalue of the matrix $H_{j,j}$, which is a consequence of the updating in step 12. The first time we reach step 7 for each j, $s_j = 1$ and $h_j = 0$ and we fill the last column of $H_{j,j}$ with the Gram Schmidt coefficients, precisely as in Arnoldi.

At later inner iterations, we have the basic recursion,

$$A(\sigma)^{-1}A(\mu)V_j \begin{bmatrix} I_{j-1} & s_1 \\ 0 & s_j \end{bmatrix} = V_j \,[\, H_{j,j-1} \quad k_j \,] + re_j^T\,,$$

where s_1 stands for the leading $j-1$ vector of s. Multiply by the inverse of the bracketed matrix on the left hand side and get the matrix on the right hand side,

$$\begin{aligned} H'_{j,j} &= [\, H_{j,j-1} \quad k_j \,] \begin{bmatrix} I_{j-1} & -s_1 s_j^{-1} \\ 0 & s_j^{-1} \end{bmatrix} = \\ &= [\, H_{j,j-1} \quad -H_{j,j-1}s_1 s_j^{-1} + k_j s_j^{-1} \,]\,. \end{aligned}$$

Now $H_{j,j-1}s_1 + h_j s_j = 0$, since s is a zero eigenvector of $H_{j,j}$, and then the last column of the updated matrix $H'_{j,j}$ is $h_j + k_j s_j^{-1}$.

Step 12: This recursion is derived from the linear interpolation (2), and makes sure that the updated $H_{j,j}$ gets zero as an eigenvalue with s as eigenvector.

Step 15: When you come here the residual estimate signals convergence, and the eigenvalue and eigenvector is returned. Please note that this cannot be deferred to the very end, because when looking for further eigenvalues, H will be modified and the eigenvalue corresponding to this μ will move.

Step 16: This is very similar to what is done in implicitly restarted Arnoldi, see [1, Section 7.6]. We do locking of those eigenvalues θ that correspond to eigenvalues λ closer to the pole σ than the just converged value μ, i.e., those outside the circle $|\theta - 1| = 1$, and keep just one of those inside the circle to act as a starting guess $\theta = \theta_i$ for the next eigenvalue in turn to be computed.

Step 17: This way we deflate the problem and will get good approximations to further eigenvalues.

In the present version, there are no precautions taken for bad starting guesses, and it is assumed that the problem lets itself be linearized by Lagrange interpolation (2) to at least some accuracy.

Convergence is reported when the residual,

$$\|A(\sigma)^{-1}A(\lambda)x\| < tolc,$$

and that is an indication, but no guarantee, that λ is close to an eigenvalue and x to an eigenvector of the nonlinear eigenproblem (1).

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. P. Hager and N.-E. Wiberg, *The rational Krylov algorithm for nonlinear eigenvalue problems.* submitted (2000).
3. V. N. Kublanovskaja, *On an application of Newton's method to the determination of eigenvalues of λ-matrices.* — Dokl. Akad. Nauk SSSR, transl Soviet. Math. Dokl. **10** (1969), 1240–1241.
4. V. N. Kublanovskaya, *On an approach to the solution of the generalized latent value problem for λ-matrices.* — SIAM J. on Num. Anal. **7** (1970), 532–537.
5. A. Ruhe, *Algorithms for the nonlinear algebraic eigenvalue problem.* — SIAM J. Num. Anal **10** (1973), 674–689.
6. A. Ruhe, *Rational Krylov, a practical algorithm for large sparse nonsymmetric matrix pencils.* — SIAM J. Sci. Comp. **19** (1998), 1535–1551.

Department of Mathematics,
Chalmers Institute of Technology
and the University of Göteborg,
S-41296 Göteborg, Sweden
(ruhe@math.chalmers.se)

Поступило 25 сентября 2000 г.