

Schur Decomposition Methods for the Computation of Rational Matrix Functions

T. Politi¹ and M. Popolizio²

¹ Dipartimento di Matematica, Politecnico di Bari,
Via Amendola 126/B, I-70126 Bari (Italy)
politit@poliba.it

² Dipartimento di Matematica, Università degli Studi di Bari,
Via Orabona 4, I-70125 Bari (Italy)
popolizio@dm.uniba.it

Abstract. In this work we consider the problem to compute the vector $\mathbf{y} = \Phi_{m,n}(A)\mathbf{x}$ where $\Phi_{m,n}(z)$ is a rational function, \mathbf{x} is a vector and A is a matrix of order N , usually nonsymmetric. The problem arises when we need to compute the matrix function $f(A)$, being $f(z)$ a complex analytic function and $\Phi_{m,n}(z)$ a rational approximation of f . Hence $\Phi_{m,n}(A)$ is a approximation for $f(A)$ cheaper to compute. We consider the problem to compute first the Schur decomposition of A then the matrix rational function exploiting the partial fractions expansion. In this case it is necessary to solve a sequence of linear systems with the shifted coefficient matrix $(A - z_j I)\mathbf{y} = \mathbf{b}$.

1 Introduction

Matrix functions arise in different fields of Mathematical Sciences and Engineering, in particular in connection with the solution of ordinary differential systems, with applications in control theory (see [1]), nuclear magnetic resonance, Lie group methods for geometric integration (see [8]), and in the numerical solution of stiff differential equations (see [11], and references therein). Usually the matrix function can be defined via power series or as the solution of non-linear systems. Several methods have been proposed in past to solve one of the following problems:

- Given a complex matrix $A \in \mathbb{C}^{n \times n}$ compute $f(A)$, where f is an analytic function on a region of the complex plane;
- Given a matrix A with a given geometric structure compute the matrix function $f(A)$, having a special structure too;
- Given a matrix A and a vector \mathbf{x} compute $f(A)\mathbf{x}$.

The second problem has a great interest in very important cases when $f(z)$ is the exponential function. In fact it is well known that the exponential map is a function defined from a Lie Algebra to its related Lie Group, i.e. if A is skew-symmetric (i.e. $A^T = -A$) then $Q = \exp(A)$ is orthogonal. In these cases it is very important in the numerical approximation to preserve the property.

Recently this problem has been considered using approaches based on the Krylov subspaces technique but also exploiting the Schur decomposition of Hamiltonian skew-symmetric matrices ([5, 9]).

In past many algorithms have been proposed for the approximation of matrix $f(A)$ (see [7]) and most of these are based on the rational approximation of f ,

$$\Phi_{m,n}(z) = \frac{R_m(z)}{Q_n(z)},$$

being $R_m(z)$ and $Q_n(z)$ polynomials of degree m and n , respectively. Hence

$$f(A) \simeq \Phi_{m,n}(A) = R_m(A) [Q_n(A)]^{-1}.$$

A classical way to obtain the function $\Phi_{m,n}(z)$ is to use the Padé approximation (usually with $m = n$) together with a pre-processing technique for matrix A (usually the scaling and squaring technique, see [6]).

In this work we describe briefly some numerical methods based on the Schur decomposition of the matrix, in order to compute the rational function approximation of the exponential matrix function, then we consider the use of the partial fraction expansion. In Section 2 we describe the numerical methods based on the Schur decomposition, while in Section 3 we describe the algorithms to compute rational matrix functions based on the partial fractions expansion technique. Finally in Section 4 some numerical issues are shown.

2 Numerical Methods Based on the Schur Decomposition

A general approach to compute $f(A)$, with $A \in \mathbb{C}^{n \times n}$ is to employ similarity transformation

$$A = SDS^{-1} \tag{1}$$

where $f(D)$ is easily computable. In fact

$$f(A) = Sf(D)S^{-1}. \tag{2}$$

If A is diagonalizable then we could take D diagonal hence the computation of $f(D)$ is trivial since it is a diagonal matrix. For stability problems it is a good choice to take well conditioned matrix S in (1). The better way to take a well conditioned matrix is S to be orthogonal, so that the decomposition (1) becomes the Schur decomposition of A :

$$A = QTQ^*,$$

where T is upper triangular or block upper triangular. The computation of the Schur factors is achieved with perfect backward stability by the QR algorithm (see [6]). Once the decomposition has been performed the problem is the computation of $f(T)$. If T is upper triangular also

$$F = f(T)$$

is upper triangular, so that Parlett (see [10]) proposed the following recurrence relation, which comes equating the (i, j) elements ($i < j$) in the commutativity relation $FT = TF$:

$$f_{ij} = t_{ij} \frac{f_{ii} - f_{jj}}{t_{ij} - t_{ji}} + \sum_{k=i+1}^{j-1} \frac{f_{ik}t_{kj} - t_{ik}f_{kj}}{t_{ii} - t_{jj}} \tag{3}$$

The problem with this approach is that the recurrence breaks in when $t_{ii} = t_{jj}$ for some $i \neq j$, that is when T has repeated eigenvalues, and it can give inaccurate results in floating point arithmetic when T has close eigenvalues. Parlett observed that if $T = (T_{ij})$ is block upper triangular then $F = (F_{ij})$ has the same block structure and, for $i < j$,

$$T_{ii}F_{ij} - F_{ij}T_{jj} = F_{ii}T_{ij} - T_{ij}F_{jj} + \sum_{k=i+1}^{j-1} (F_{ik}T_{kj} - T_{ik}F_{kj}). \tag{4}$$

The recurrence can be used to compute F a superdiagonal at a time, provided we can evaluate the blocks $F_{ii} = f(T_{ii})$ and solve the Sylvester equation (4) for the F_{ij} . For the equation (4) to be nonsingular we need that T_{ii} and T_{jj} have no common eigenvalue. For the Sylvester equations to be well conditioned a necessary condition is that the eigenvalues of T_{ii} and T_{jj} are well separated. An alternative approach is first to compute

$$A = XDX^{-1}$$

where X is well conditioned and D is block diagonal. Then

$$f(A) = Xf(D)X^{-1}$$

and the problem reduces to compute $f(D)$. The usual way to compute a block diagonalization is first to compute the Schur form and then to eliminate off-diagonal blocks by solving Sylvester equations (see [6]). In order to guarantee a well conditioned matrix X a bound must be imposed on the condition of the individual transformations, this bound will be a parameter of the algorithm (see [4]).

Computing $f(D)$ reduces to compute $f(D_{ii})$ for each diagonal block D_{ii} . The D_{ii} are triangular but, unlike the Schur method, no particular eigenvalue distribution is guaranteed, because of limitations on the condition of the transformations; therefore $f(D_{ii})$ is still a nontrivial calculation.

3 Partial Fractions Expansions

A different approach in the computation of the matrix function $f(A)$ is to use a rational approximation, $\Phi_{m,n}(z)$:

$$\Phi_{m,n}(z) = \frac{R_m(z)}{Q_n(z)} \tag{5}$$

where $R_m(z)$ and $Q_n(z)$ are polynomials of degree m and n respectively. Usually function $\Phi_{m,n}(z)$ could be the (m, n) Padé approximation of the exponential function or the Chebyshev approximation but also another approaches are possible. The problem becomes the computation of the vector $\Phi_{m,n}(A)\mathbf{y}$. Supposing all the roots of polynomial $Q_n(z)$ to be distinct and $n \geq m$, the rational function $\Phi_{m,n}(z)$ can be represented using the partial fractions expansion:

$$\frac{R_m(z)}{Q_n(z)} = \alpha_0 + \sum_{j=1}^n \frac{\alpha_j}{z - z_j} \quad (6)$$

where

$$\alpha_0 = \lim_{z \rightarrow \infty} \frac{R_m(z)}{Q_n(z)}$$

and

$$\alpha_j = \frac{R_m(z)}{Q'_n(z)} \quad (7)$$

where z_j are the roots of $Q_n(z)$. Hence

$$\Phi_{m,n}(A) = R_m(A) [Q_n(A)]^{-1}$$

and, from (6):

$$\Phi_{m,n}(A)\mathbf{x} = \alpha_0\mathbf{x} + \sum_{j=1}^n \alpha_j (A - z_j I)^{-1} \mathbf{x}.$$

The vector \mathbf{y} can be computed using the following algorithm:

1. Compute

$$\alpha_0 = \lim_{z \rightarrow \infty} \frac{R_m(z)}{Q_n(z)}, \quad \alpha_j = \frac{R_m(z)}{Q'_n(z)}$$

for $j = 1, \dots, n$.

2. For $j = 1, \dots, n$, compute the vector \mathbf{x}_j solving

$$(A - z_j I)\mathbf{x}_j = \mathbf{b}$$

3. Set

$$\mathbf{y} = \alpha_0 \mathbf{b} + \sum_{j=1}^n \alpha_j \mathbf{x}_j.$$

The main problem is the solution of the linear systems at step 2. A good choice for this solution is to use Krylov-subspace iterations, since they are invariant under the shifts z_j . Hence the work required to solve n linear systems is the same to solve one system. Another advantage of this technique is that it can easily implemented on a parallel architecture since the linear systems to be solved in step 2 are independent (see [2]). Moreover there is a second level parallelism since also each system could be solved using a parallel algorithm. As last remark

we consider that, when the rational function has poles and residuals as complex conjugate pairs, it is possible to exploit the following property (see [12]):

$$\alpha_j(A - z_j I)^{-1} \mathbf{x}_j + \bar{\alpha}_j(A - \bar{z}_j I)^{-1} \mathbf{x}_j = 2\Re(\alpha_j(A - z_j I)^{-1} \mathbf{x}_j). \tag{8}$$

In [3] it is shown that the algorithm based on partial fractions representation can be very sensitive to perturbations. In fact the coefficients (7) are

$$\alpha_j = \frac{R_m(z_j)}{a_0 \prod_{k=1, k \neq j}^n (z_j - z_k)}$$

and it follows that the presence of closed poles z_k may cause some coefficient α_j to be very large. In this case the error on the computation of vector \mathbf{x}_j is amplified. In ([3]) these difficulties are remedied using an incomplete partial fraction representation of the rational function. The representation consists in writing the function as

$$\frac{R_m(z)}{Q_n(z)} = \prod_{k=1}^t \frac{r_l(z)}{q_l(z)} \tag{9}$$

and using a partial fraction representation for each of the factors $r_l(z)/q_l(z)$. The functions $r_l(z)$ and $q_l(z)$ are polynomials such that the degree of $q_l(z)$ is not greater than the degree of $p_l(z)$ for each l . The polynomials $r_l(z)$ and $q_l(z)$ are chosen so that the partial fraction coefficient α_{jl} of the representation

$$\frac{r_l(z)}{q_l(z)} = \alpha_{0l} + \sum_{j=1}^{n_l} \frac{\alpha_{jl}}{z - z_{jl}}$$

where n_l is the degree of $q_l(z)$, and $j = 1, \dots, n_l$, and $1 \leq l \leq t$, are not so large and the number of factors t is small. The incomplete partial representation algorithm can be described through the following steps:

1. Given the sets z_{jl} and α_{jl} with $j = 1, \dots, n_l$, and $l = 1, \dots, t$;
2. Put $\mathbf{x} = \mathbf{b}$;
3. For $l = 1, \dots, t$:
 - 3.1 Solve the systems

$$(A - z_{jl} I) \mathbf{x}_j = \mathbf{x}, \quad j = 1, \dots, n_l$$

3.2 Compute the vector

$$\mathbf{x} = \alpha_{0l} \mathbf{x} + \sum_{j=1}^{n_l} \alpha_{jl} \mathbf{x}_j.$$

The methods described previously could be used together computing first the Schur decomposition of matrix A :

$$A = QTQ^*$$

and then using the partial fractions expansions of the rational approximation of function $f(z)$ applied to matrix T . Hence given the vector \mathbf{x} we have the approximation

$$f(A)\mathbf{x} \simeq Q\Phi_{m,n}(T)\mathbf{y}$$

where $\mathbf{y} = Q^*\mathbf{x}$.

4 Numerical Examples

In this section we show an example of the application of the algorithm described in the previous section to the computation of the exponential matrix applied to a real vector. Given the unsymmetric matrix $A \in \mathbb{R}^{n \times n}$ and the vector $\mathbf{x} \in \mathbb{R}^n$ our aim is to approximate the vector $e^A\mathbf{x}$. We recall that the method is based on the following steps:

1. Computation of the real Schur decomposition of the real matrix $A = QTQ^T$;
2. Computation of the Chebyshev rational approximation for the exponential function $C_{n,n}(z)$;
3. Computation of the poles z_j and the residuals α_j of rational function $C_{n,n}(z)$;
4. Solve the linear systems $(T - z_j I)\mathbf{x}_j = \mathbf{y}$, where $\mathbf{y} = Q^T\mathbf{x}$;
5. Compute the vector

$$\mathbf{w} = \alpha_0\mathbf{y} + \sum_{j=1}^n \alpha_j\mathbf{x}_j;$$

6. Compute the vector $\mathbf{z} = Q\mathbf{w}$.

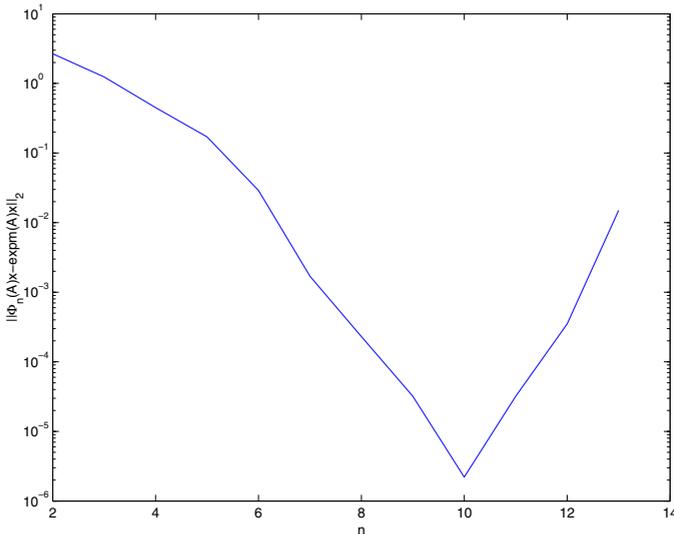


Fig. 1. Estimate of the error

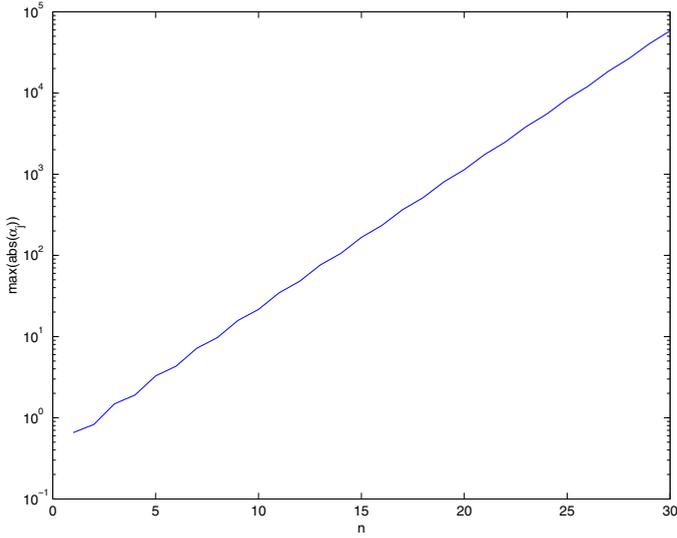


Fig. 2. Maximum Residuals of the Partial Fractions Expansion for Chebyshev Approximations of degree n

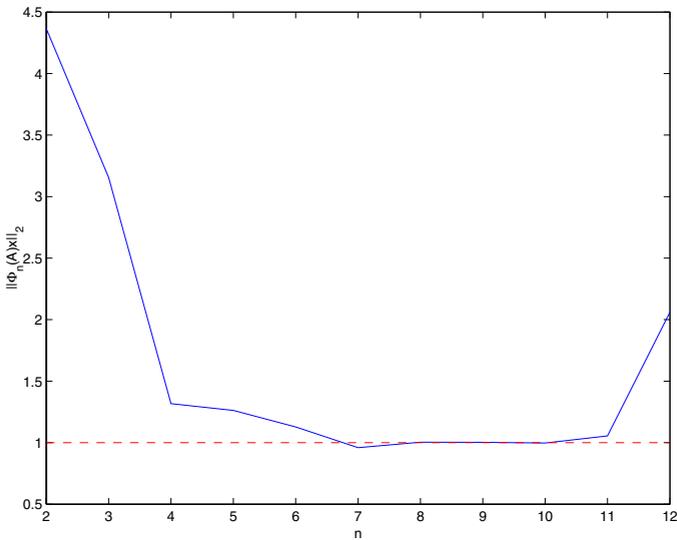


Fig. 3. Error respect to unitary vector solution

It is obvious that the steps 2 and 3 of the algorithm do not require any computation since the coefficients of the rational function $C_{n,n}(z)$, the poles and the residuals are tabbed. For the solution of these systems here we have considered just direct methods neglecting their shifted structure, in future we shall consider the application of iterative methods based on Krylov subspaces. A further

remark is related to the use of Chebyshev approximation in step 2: we recall that using the property (8) the number of linear systems can be reduced. In Figure 3 we show the difference between the vector computed by MatLab function `expm` times a random vector \mathbf{x} taking a random matrix $A \in \mathbb{R}^{20 \times 20}$ and the vector computed by the algorithm described in the present section. We observe that taking the result computed by MatLab routine as correct the error decreases when the degree of the Chebyshev approximation is growing until $n = 10$, then the error increases. The reason of this behaviour is explained in Figure 3 where we show that the maximum of the residuals $|\alpha_j|$ increases with n , and as observed in the Section 3 the error in the solution of the linear systems is amplified. Finally we have considered a random skew-symmetric matrix A and a unitary 2-norm vector \mathbf{x} . In this case also the vector $e^A \mathbf{x}$ has unitary 2-norm. We observe that the behaviour of the error is exactly the same observed in the first example.

References

1. Åström K.J., Wittenmark B.: *Computer-Controlled Systems: Theory and Design*. Prentice-Hall, Englewoods Cliffs, NJ, (1997)
2. Baldwin C., Freund R.W., Gallopoulos E.: A Parallel Iterative Method for Exponential Propagation. *Proceedings of the Seventh SIAM Conference on Parallel Processing for Scientific Computing*. D.H. Bailey et al. ed. SIAM (1995) 534–539
3. Calvetti D., Gallopoulos E., Reichel L.: Incomplete Partial Fractions for Parallel Evaluation of Rational Matrix Functions. *J. Comp. Appl. Math.* **59** (1995) 349–380
4. Davies P.J., Higham N.J.: A Schur-Parlett Algorithm for Computing Matrix Functions. *SIAM J. Matr. Anal. Appl.* **25** (2) (2003) 464–485
5. Del Buono N., Lopez L., Politi T.: Computation of functions of Hamiltonian and skew-symmetric matrices. Preprint (2006)
6. Golub G.H., Van Loan C.F.: *Matrix Computation*. The John Hopkins Univ. Press, Baltimore, (1996)
7. Higham N.J.: *Functions of Matrices*. MIMS EPrint 2005.21 The University of Manchester
8. Iserles A., Munthe-Kaas H., Nørsett S., Zanna A.: Lie-Group Methods. *Acta Numerica* **9** (2000) 215–365
9. Lopez L., Simoncini V.: Analysis of projection methods for rational function approximation to the matrix exponential. *SIAM J. Numer. Anal.* (to appear)
10. Parlett B.N.: A Recurrence among the Elements of Functions of Triangular Matrices. *Lin. Alg. Appl.* **14** (1976) 117–121
11. Saad Y.: Analysis of some Krylov subspace approximation to the matrix exponential operator, *SIAM J. Numer. Anal.* **29** (1) (1992) 209–228
12. Schmelzer T.: *Rational approximations in scientific computing*. Computing Laboratory, Oxford University, U.K. (2005)