Computing $f(A)b$ for Matrix Functions f

Davies, Philip I. and Higham, Nicholas J.

2005

ISSN 1749-9097

Computing $f(A)b$ for Matrix Functions f [⋆]

Philip I. Davies and Nicholas J. Higham[⋆⋆]

School of Mathematics, University of Manchester, Manchester, M13 9PL, England.
pdavies@ma.man.ac.uk, higham@ma.man.ac.uk.

Summary. For matrix functions f we investigate how to compute a matrix-vector product $f(A)b$ without explicitly computing $f(A)$. A general method is described that applies quadrature to the matrix version of the Cauchy integral theorem. Methods specific to the logarithm, based on quadrature, and fractional matrix powers, based on solution of an ordinary differential equation initial value problem, are also presented

1 Introduction

A large literature exists on methods for computing functions $f(A)$ of a square matrix A, ranging from methods for general f to those that exploit properties of particular functions. In this work we consider the problem of computing $y = f(A)b$, for a given matrix A and vector b. Our aim is to develop methods that require less computation than forming $f(A)$ and then multiplying into b.

Motivation for this problem comes from various sources, but particularly from lattice quantum chromodynamics (QCD) computations in chemistry and physics; see [6], [17] and elsewhere in this proceedings. Here, $f(A)b$ must be computed for functions such as $f(A) = A(A^*A)^{-1/2}$, with A very large, sparse, complex and Hermitian. Applications arising in the numerical solution of stochastic differential equations are described in [1], with $f(A) = A^{1/2}$ and A symmetric positive definite. More generally, it might be desired to compute just a single column of $f(A)$, in which case b can be taken to be a unit vector e_i. We mention that Bai, Fahey and Golub [2] treat the problem of computing upper and lower bounds for a quadratic form $u^T f(A)v$, principally for symmetric positive definite A.

We treat general nonsymmetric A and assume that factorization of A is feasible. While our methods are not directly applicable to very large, sparse

[⋆] This work was supported by Engineering and Physical Sciences Research Council grant GR/R22612.

[⋆⋆] Supported by a Royal Society-Wolfson Research Merit Award.

A, they should be useful in implementing methods specialized to such A. For example, in the QCD application the Lanczos-based method of [17, Sec. 4.6] requires the computation of $T^{-1/2}e_1$, where T is symmetric tridiagonal, while techniques applying to general f and sparse A and leading to dense subproblems are described by Van Der Vorst [18].

2 Rational approximations

A rational approximation $r(A) = q(A)^{-1}p(A) \approx f(A)$, where p and q are polynomials, can be applied directly to the $f(A)b$ problem to give $y = f(A)b \approx r(A)b$ as the solution of $q(A)y = p(A)b$. Forming $q(A)$ is undesirable, so if this formulation is used then iterative methods requiring only matrix-vector products must be used to solve the linear system [18]. It may also be possible to express $r(A)$ in linear partial fraction form, so that y can be computed by solving a sequence of linear systems involving A but not higher powers (an example is given in the next section). The issues here are largely in construction of the approximation $r(A)$, and hence are not particular to the $f(A)b$ problem. See Golub and Van Loan [8, Chap. 11] for a summary of various rational approximation methods.

3 Matrix logarithm

We consider first the principal logarithm of a matrix $A \in \mathbb{C}^{n\times n}$ with no eigenvalues on $\mathbb{R}^-$ (the closed negative real axis). This logarithm is denoted by $\log A$ and is the unique matrix Y such that $\exp(Y) = A$ and the eigenvalues of Y have imaginary parts lying strictly between $-\pi$ and π. We will exploit the following integral representation, which is given, for example, by Wouk [19].

Theorem 1. *For $A \in \mathbb{C}^{n\times n}$ with no eigenvalues on $\mathbb{R}^-$,*

$$\log\bigl(s(A-I)+I\bigr) = \int_0^s (A-I)\bigl[t(A-I)+I\bigr]^{-1}\,dt,$$

and hence

$$\log A = \int_0^1 (A-I)\bigl[t(A-I)+I\bigr]^{-1}\,dt. \tag{1}$$

Proof. It suffices to prove the result for diagonalizable A [11, Thm. 6.2.27 (2)], and hence it suffices to show that

$$\log(s(x-1)+1) = \int_0^s (x-1)\bigl[t(x-1)+1\bigr]^{-1}\,dt$$

for $x \in \mathbb{C}$ lying off $\mathbb{R}^-$; this latter equality is immediate. □

The use of quadrature to approximate the integral (1) is investigated by Dieci, Morini and Papini [5]. Quadrature is also directly applicable to our $f(A)b$ problem. We can apply a quadrature rule

$$\int_0^1 g(t)\,dt \approx \sum_{k=1}^{m} c_k g(t_k) \tag{2}$$

to (1) to obtain

$$(\log A)\,b \approx \left(\sum_{k=1}^{m} c_k \big[t_k(A-I)+I\big]^{-1}\right)(A-I)b. \tag{3}$$

Unlike when quadrature is used to approximate $\log A$ itself, computational savings accrue from reducing A to a simpler form prior to the evaluation. Since A is a general matrix we compute the Hessenberg reduction

$$A = QHQ^T, \tag{4}$$

where Q is orthogonal and H is upper Hessenberg, and evaluate

$$(\log A)\,b \approx Q\sum_{k=1}^{m} c_k \big[t_k(H-I)+I\big]^{-1} d, \qquad d = Q^T(A-I)b, \tag{5}$$

where the Hessenberg linear systems are solved by Gaussian elimination with partial pivoting (GEPP). The computation of (4) (with Q maintained in factored form) and the evaluation of (5) cost $(10/3)n^3 + 2mn^2$ flops, whereas evaluation from (3) using GEPP to solve the linear systems costs $(2/3)mn^3$ flops; thus unless $m \lesssim 5$ the Hessenberg reduction approach is the more efficient for large n. If m is so large that $m \gtrsim 32n$ then it is more efficient to employ a (real) Schur decomposition.

Gaussian quadrature is a particularly interesting possibility in (2). It is shown by Dieci, Morini and Papini [5, Thm. 4.3] that applying the m point Gauss-Legendre quadrature rule to (1) produces the rational function $r_m(A-I)$, where $r_m(x)$ is the $[m/m]$ Padé approximant to $\log(1+x)$, the numerator and denominator of which are polynomials in x of degree m. Padé approximants to $\log(I+X)$ are a powerful tool whose use is explored in detail in [3], [10]. These approximations are normally used only for $\|X\| < 1$, and under this condition Kenney and Laub [13] show that the error in the matrix approximation is bounded by the error in a corresponding scalar approximation:

$$\|r_m(X) - \log(I+X)\| \le |r_m(-\|X\|) - \log(1-\|X\|)|; \tag{6}$$

the norm here is any subordinate matrix norm. The well known formula for the error in Gaussian quadrature provides an exact expression for the error $r_m(X) - \log(I+X)$. As shown in [5, Cor. 4.4] this expression can be written as a power series in X when $\|X\| < 1$. Both these approaches provide error bounds for the approximation of $(\log A)\, b$.

We note that in the case where A is symmetric positive definite, the method of Lu [14] for computing $\log(A)$ that uses Padé approximants is readily adapted to compute $(\log A)b$. That method places no restrictions on $\|I - A\|_2$ but it strongly relies on the symmetry and definiteness of A.

In the inverse scaling and squaring method for computing $\log A$ [3], repeated square roots are used to bring A close to I, with subsequent use of the identity $\log A = 2^k \log A^{1/2^k}$. Unfortunately, since each square root requires $O(n^3)$ flops, this approach is not attractive in the context of computing $(\log A)\, b$.

When $\|I - A\| > 1$, we do not have a convenient bound for the error in the m-point Gauss-Legendre approximation to $(\log A)\, b$. While it follows from standard results on the convergence of Gaussian quadrature [4] that the error in our approximation converges to zero as $m \to \infty$, we cannot predict in advance the value of m needed. Therefore for $\|I - A\| > 1$ adaptive quadrature is the most attractive option.

We report numerical experiments for four problems:

$$\begin{aligned}
A &= \texttt{eye(64)} + 0.5\,\texttt{urandn(64)}, & b &= \texttt{urandn(64,1)},\\
A &= \texttt{eye(64)} + 0.9\,\texttt{urandn(64)}, & b &= \texttt{urandn(64,1)},\\
A &= \texttt{gallery('parter',64)}, & b &= \texttt{urandn(64,1)},\\
A &= \texttt{gallery('pascal',8)}, & b &= \texttt{urandn(8,1)},
\end{aligned}$$

where we have used MATLAB notation. In addition, `urandn(m,n)` denotes an $m \times n$ matrix formed by first drawing the entries from the normal $N(0, 1)$ distribution and then scaling the matrix to have unit 2-norm. The Parter matrix is mildly nonnormal, with eigenvalues lying in the right half-plane on a curve shaped like a "U" rotated anticlockwise through 90 degrees. The Pascal matrix is symmetric positive definite, with eigenvalues ranging in magnitude from 10^{-4} to 10^3.

We computed $y = \log(A)b$ using a modification of the MATLAB adaptive quadrature routine `quadl`, which is based on a 4-point Gauss-Lobatto rule together with a 7-point Kronrod extension [7]. Our modification allows the integration of vector functions. We employ the Hessenberg reduction as in (5). We also computed the m-point Gauss-Legendre approximation, using the smallest m such that the upper bound in (6) was less than the tolerance when $\|I - A\|_2 < 1$, or else by trying $m = 1, 2, 3 \ldots$ successively until the absolute error $\|\log(A)b - \widehat{y}\|_2$ was no larger than the tolerance. Three different absolute error tolerances tol were used. The results are reported in Tables 1–4, in which "g evals" denotes the value of m in (5) that `quadl` effectively uses.

We see from the results that Gauss-Legendre quadrature with an appropriate choice of m is more efficient than the modified `quadl` in every case. This is not surprising in view of the optimality properties of Gaussian quadrature, and also because adaptive quadrature incurs an overhead in ensuring that an error criterion is satisfied [15]. The inefficiency of adaptive quadrature is particularly notable in Table 1, where at least 18 function evaluations are always required and a much more accurate result than necessary is returned. Recall, however that unless $\|I - A\|_2 > 1$ we have no way of choosing m for the Gauss-Legendre approximation automatically. We also observe that the error bound (6) provides a rather pessimistic bound for the $(\log A)\, b$ error in Table 2.

Table 1. Results for $A =$ `eye(64)` $+ 0.5$ `urandn(64)`, $b =$ `urandn(64,1)`. $\|I - A\|_2 = 0.5$, $\|\log(A)\|_2 = 5.3\text{e-}1$.

	Adaptive quadrature		Gauss-Legendre		
tol	g evals	Abs. err.	m	Abs. err.	Upper bound in (6)
1e-3	18	1.2e-13	2	7.9e-6	8.4e-4
1e-6	18	1.2e-13	4	1.1e-10	7.6e-7
1e-9	18	1.2e-13	6	4.1e-15	6.7e-10

Table 2. Results for $A =$ `eye(64)` $+ 0.9$ `urandn(64)`, $b =$ `urandn(64,1)`. $\|I - A\|_2 = 0.9$, $\|\log(A)\|_2 = 1.0$.

	Adaptive quadrature		Gauss-Legendre		
tol	g evals	Abs. err.	m	Abs. err.	Upper bound in (6)
1e-3	18	5.6e-10	7	9.7e-13	3.2e-4
1e-6	18	5.6e-10	12	3.1e-15	4.7e-7
1e-9	18	5.6e-10	17	3.1e-15	6.8e-10

Table 3. Results for $A =$ `gallery('parter',64)`, $b =$ `urandn(64,1)`. $\|I - A\|_2 = 3.2$, $\|\log(A)\|_2 = 1.9$.

	Adaptive quadrature		Gauss-Legendre	
tol	g evals	Abs. err.	m	Abs. err.
1e-3	48	1.6e-4	8	5.2e-6
1e-6	48	1.4e-10	10	2.5e-7
1e-9	138	1.6e-13	14	5.2e-10

4 Matrix powers

To compute the action of an arbitrary matrix power on a vector we identify an initial value ODE problem whose solution is the required vector.

Table 4. Results for A = `gallery('pascal',8)`, b = `urandn(64,1)`. $\|I-A\|_2 =$ 4.5e3, $\|\log(A)\|_2 = 8.4$.

	Adaptive quadrature		Gauss-Legendre	
tol	g evals	Abs. err.	m	Abs. err.
1e-3	198	1.1e-5	128	1.0e-3
1e-6	468	2.8e-10	245	1.0e-6
1e-9	1158	5.7e-14	362	9.8e-10

Note that for a positive integer p and A having no eigenvalues on $\mathbb{R}^-$, $A^{1/p}$ denotes the principal pth root: the pth root whose eigenvalues lie in the segment $\{\, z : -\pi/p < \arg(z) < \pi/p \,\}$. For other fractional α, A^α can be defined as $\exp(\alpha \log A)$.

Theorem 2. *For $A \in \mathbb{C}^{n\times n}$ with no eigenvalues on $\mathbb{R}^-$ and $\alpha \in \mathbb{R}$, the initial value ODE problem.*

$$\frac{dy}{dt} = \alpha(A-I)\big[t(A-I)+I\big]^{-1} y, \qquad y(0) = b, \qquad 0 \le t \le 1, \tag{7}$$

has a unique solution $y(t) = \big[t(A-I)+I\big]^\alpha b$, and hence $y(1) = A^\alpha b$.

Proof. The existence of a unique solution follows from the fact that the ODE satisfies a Lipschitz condition with Lipschitz constant $\sup_{0\le t\le 1} \|(A-I)\big[t(A-I)+I\big]^{-1}\| < \infty$. It is easy to check that $y(t)$ is this solution. □

This result is obtained by Allen, Baglama and Boyd [1] in the case $\alpha = 1/2$ and A symmetric positive definite. They propose using an ODE initial value solver to compute $x(1) = A^{1/2}b$.

Applying an ODE solver is the approach we consider here also. The initial value problem can potentially be stiff, depending on α, the matrix A, and the requested accuracy, so some care is needed in choosing a solver. Again, a Hessenberg reduction of A can be used to reduce the cost of evaluating the differential equation.

We report an experiment with the data A = `gallery('parter',64)`, b = `urandn(64,1)`, as used in the previous section, with $\alpha = -1/2$ and $\alpha = 2/5$. We solved the ODE initial value problem with MATLAB's `ode45` function [9, Chap. 12], obtaining the results shown in Table 5; here, tol is the relative error tolerance, and the absolute error tolerance in the function's mixed absolute/relative error test was set to 10^{-3}tol. It is clear from the displayed numbers of successful steps and failed attempts to make a step that `ode45` found the problems relatively easy.

5 General f: Cauchy integral theorem

For general f we can represent $y = f(A)b$ using the matrix version of the Cauchy integral theorem:

Table 5. Results for $A =$ `gallery('parter',64)`, $b =$ `randn(64,1)`.

$f(A)$	tol	Succ. steps	Fail. atts	ODE evals	Rel. err
$A^{-1/2}$	1e-3	12	0	73	3.5e-8
	1e-6	14	0	85	6.0e-9
	1e-9	40	0	241	7.7e-12
$A^{2/5}$	1e-3	15	0	79	2.8e-8
	1e-6	16	0	91	2.4e-9
	1e-9	54	0	325	1.8e-12

$$y = \frac{1}{2\pi i}\int_{\Gamma} f(z)(zI - A)^{-1}b\; dz, \tag{8}$$

where f is analytic inside a closed contour Γ that encloses the eigenvalues of A. We take for the contour Γ a circle with centre α and radius β,

$$\Gamma\colon\; z - \alpha = \beta e^{i\theta}, \quad 0 \le \theta \le 2\pi, \tag{9}$$

and then approximate the integral using the repeated trapezium rule. Using $dz = i\beta e^{i\theta} d\theta = i d\theta (z(\theta) - \alpha)$, and writing the integrand in (8) as $g(z)$, we obtain

$$\int_{\Gamma} g(z) dz = i\int_0^{2\pi} (z(\theta) - \alpha) g(z(\theta))\, d\theta. \tag{10}$$

The integral in (10) is a periodic function of θ with period 2π. Applying the n-point repeated trapezium rule to (10) gives

$$\int_{\Gamma} g(z)\; dz \approx \frac{2\pi i}{n}\sum_{k=0}^{n-1}(z_k - \alpha) g(z_k),$$

where $z_k - \alpha = \beta e^{2\pi k i/n}$, that is, $z_0, \dots, z_n$ are equally spaced points on the contour Γ (note that since Γ is a circle we have $z_0 = z_n$). When A is real and we take α real it suffices to use just the z_k in the upper half-plane and then take the real part of the result. When applied to periodic functions the repeated trapezium rule can produce far more accurate results than might be expected from the traditional error estimate [4]

$$\int_a^b f(x) dx - T_n(f) = -\frac{(b-a)^3}{12n^2} f''(\xi), \quad a < \xi < b,$$

where $T_n(f)$ denotes the n-point repeated trapezium rule for the function f. The following theorem can be shown using the Euler-Maclaurin formula.

Theorem 3 ([4, p. 137]). *Let $f(x)$ have period 2π and be of class $C^{2k+1}(-\infty, \infty)$ with $|f^{(2k+1)}(x)| \le M$. Then*

$$\left|\int_0^{2\pi} f(x)\; dx - T_n(f)\right| \le \frac{4\pi M\,\zeta(2k+1)}{n^{2k+1}},$$

where $\zeta(k) = \sum_{j=1}^{\infty} j^{-k}$ *is the Riemann zeta function.*

We want to apply Theorem 3 to the integral (10)

$$\int_0^{2\pi} h(\theta)\, d\theta := \frac{1}{2\pi} \int_0^{2\pi} (z(\theta) - \alpha) f\bigl(z(\theta)\bigr) \bigl(z(\theta)I - A\bigr)^{-1} b \, d\theta \qquad (11)$$

where $z(\theta) = \alpha + \beta e^{i\theta}$. The integrand is continuously differentiable so we can choose any k in Theorem 3. We need to consider the derivatives of the integrand in (11), which have the form

$$h^{(k)}(\theta) = \frac{i^k}{2\pi} \sum_{j=0}^{k} (z(\theta) - \alpha)^{j+1} \sum_{i=0}^{j} c_{ijk} f^{(j-i)}(z(\theta))(z(\theta)I - A)^{-(1+i)} b, \qquad (12)$$

for certain constants c_{ijk}.

Several terms in (12) can make $|h^{(2k+1)}(\theta)|$ large and therefore make the error bound in Theorem 3 large. First, we have the term $(z(\theta) - \alpha)^{j+1}$ where $0 \le j \le 2k + 1$. The term $z(\theta) - \alpha$ has absolute value equal to the radius of the contour, β. Therefore $|h^{(2k+1)}(\theta)|$ is proportional to β^{2k+2} and the error bound for the repeated trapezium rule will be proportional to $\beta(\beta/n)^{2k+1}$. As the contour needs to enclose all the eigenvalues of A, β needs to be large for a matrix with a large spread of eigenvalues. Therefore a large number of points are required to make the error bound small. Second, we have the powers of the resolvent, $(z(\theta)I - A)^{-(1+i)}$, where $0 \le i \le 2k + 1$. These powers will have a similar effect to β on $|h^{(2k+1)}(\theta)|$ and therefore on the error bound. These terms can be large if the contour passes too close to the eigenvalues of A; even if the contour keeps well away from the eigenvalues, the terms can be large for a highly nonnormal matrix, as is clear from the theory of pseudospectra [16]. A large resolvent can also make rounding errors in the evaluation of the integrand degrade the computed result, depending on the required accuracy. Third, the derivatives of $f(z)$ can be large: for example, for the square root function near $z = 0$. Finally, the constants c_{ijk} in (12) grow quickly with k. In summary, despite the attractive form of the bound in Theorem 3, rapid decay of the error with n is not guaranteed.

We give two examples to illustrate the performance of the repeated trapezium rule applied to (10). As in the previous sections, we use a Hessenberg reduction of A to reduce the cost of the function evaluations. We consider the computation of $y = A^{1/2}b$. Our first example is generated in MATLAB by

```
A = randn(20)/sqrt(20) + 2*eye(20); b = randn(20,1)
```

We took $\alpha = 2$ and $\beta = 1.4$ in (9), so that the contour does not closely approach any eigenvalue. From Table 6 we can see that the trapezium rule converges rapidly as the number of points increases. This is predicted by the theory since

- β, the radius of the contour, and $\|(z(\theta)I - A)^{-1}\|$ are small,

- Γ does not go near the origin, and therefore $f^{(k)}(z)$ remains of moderate size.

Our second example involves the Pascal matrix of dimensions 4 and 5. The Pascal matrix is symmetric positive definite and has a mix of small and large eigenvalues. As noted above we would like to choose the contour so that it does not go too near the eigenvalues of A and also does not go near the negative real axis, on which the principal square root function is not defined. As a compromise between these conflicting requirements we choose for Γ the circle with centre $(\lambda_{\min} + \lambda_{\max})/2$ and radius $\lambda_{\max}/2$. The results in Table 7 show that the increases in β and $\|(z(\theta)I - A)^{-1}\|$ and the proximity of Γ to the origin cause a big increase in the number of points required for convergence of the repeated trapezium rule. When we repeated the same experiment using the 6×6 Pascal matrix we found that we required over 1 million points to achieve a relative error of 7.0×10^{-5}.

Our conclusion is that the repeated trapezium rule applied to the Cauchy integral formula can be an efficient way to compute $f(A)b$, but the technique is restricted to matrices that are not too nonnormal and whose eigenvalues can be enclosed within a circle of relatively small radius that does not approach singularities of the derivatives of f too closely. We note that Kassam and Trefethen [12] successfully apply the repeated trapezium rule to the Cauchy integral formula to compute certain matrix coefficients in a numerical integration scheme for PDEs, their motivation being accuracy (through avoidance of cancellation) rather than efficiency.

Table 6. Results for `A = randn(20)/sqrt(20) + 2*eye(20)`, `b = randn(20,1)`, $\alpha = 2$, $\beta = 1.4$.

No. points	8	16	32	64	128
Rel. err.	3.0e-2	1.0e-3	1.4e-6	3.0e-12	2.1e-15

Table 7. Results for $A =$ `gallery('pascal',n)`, $b =$ `randn(n,1)`.

$n = 4$, $\alpha = 13.17$, $\beta = 13.15$		$n = 5$, $\alpha = 46.15$, $\beta = 46.14$	
No. points	Rel. err.	No. points	Rel. err.
2^{9}	1.5e-1	2^{12}	1.6e+0
2^{10}	5.0e-2	2^{13}	6.1e-1
2^{11}	9.4e-3	2^{14}	1.7e-1
2^{12}	4.6e-4	2^{15}	2.2e-2
2^{13}	1.2e-6	2^{16}	4.5e-4
2^{14}	8.9e-12	2^{17}	2.0e-7
2^{14}	3.8e-15	2^{18}	9.9e-14

References

1. E. J. Allen, J. Baglama, and S. K. Boyd. Numerical approximation of the product of the square root of a matrix with a vector. *Linear Algebra Appl.*, 310: 167–181, 2000.
2. Z. Bai, M. Fahey, and G. H. Golub. Some large-scale matrix computation problems. *J. Comput. Appl. Math.*, 74:71–89, 1996.
3. S. H. Cheng, N. J. Higham, C. S. Kenney, and A. J. Laub. Approximating the logarithm of a matrix to specified accuracy. *SIAM J. Matrix Anal. Appl.*, 22 (4):1112–1125, 2001.
4. P. J. Davis and P. Rabinowitz. *Methods of Numerical Integration.* Academic Press, London, second edition, 1984. ISBN 0-12-206360-0. xiv+612 pp.
5. L. Dieci, B. Morini, and A. Papini. Computational techniques for real logarithms of matrices. *SIAM J. Matrix Anal. Appl.*, 17(3):570–593, 1996.
6. R. G. Edwards, U. M. Heller, and R. Narayanan. Chiral fermions on the lattice. *Parallel Comput.*, 25:1395–1407, 1999.
7. W. Gander and W. Gautschi. Adaptive quadrature—revisited. *BIT*, 40(1): 84–101, 2000.
8. G. H. Golub and C. F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, MD, USA, third edition, 1996. ISBN 0-8018-5413-X (hardback), 0-8018-5414-8 (paperback). xxvii+694 pp.
9. D. J. Higham and N. J. Higham. *MATLAB Guide.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, second edition, 2005.
10. N. J. Higham. Evaluating Padé approximants of the matrix logarithm. *SIAM J. Matrix Anal. Appl.*, 22(4):1126–1135, 2001.
11. R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis.* Cambridge University Press, 1991. ISBN 0-521-30587-X. viii+607 pp.
12. A.-K. Kassam and L. N. Trefethen. Fourth-order time stepping for stiff PDEs. To appear in SIAM J. Sci. Comput., 2005.
13. C. S. Kenney and A. J. Laub. Padé error estimates for the logarithm of a matrix. *Internat. J. Control*, 50(3):707–730, 1989.
14. Y. Y. Lu. Computing the logarithm of a symmetric positive definite matrix. *Appl. Numer. Math.*, 26:483–496, 1998.
15. J. N. Lyness. When not to use an automatic quadrature routine. *SIAM Rev.*, 25(1):63–87, 1983.
16. L. N. Trefethen and M. Embree. *Spectra and Pseudospectra: The Behavior of Non-Normal Matrices and Operators.* Princeton University Press, Princeton, NJ, USA, 2005.
17. J. van den Eshof, A. Frommer, T. Lippert, K. Schilling, and H. A. van der Vorst. Numerical methods for the QCD overlap operator. I. Sign-function and error bounds. *Computer Physics Communications*, 146:203–224, 2002.
18. H. A. Van Der Vorst. Solution of $f(A)x = b$ with projection methods for the matrix A. In A. Frommer, T. Lippert, B. Medeke, and K. Schilling, editors, *Numerical Challenges in Lattice Quantum Chromodynamics*, volume 15 of *Lecture Notes in Computational Science and Engineering*, pages 18–28. Springer-Verlag, Berlin, 2000.
19. A. Wouk. Integral representation of the logarithm of matrices and operators. *J. Math. Anal. and Appl.*, 11:131–138, 1965.