The Complex Step Approximation to the Fréchet Derivative of a Matrix Function

Al-Mohy, Awad H. and Higham, Nicholas J.

2009

MIMS EPrint: **2009.31**

Manchester Institute for Mathematical Sciences
School of Mathematics

The University of Manchester

Reports available from: http://eprints.maths.manchester.ac.uk/
And by contacting: The MIMS Secretary
School of Mathematics
The University of Manchester
Manchester, M13 9PL, UK

ISSN 1749-9097

The Complex Step Approximation to the Fréchet Derivative of a Matrix Function

Awad H. Al-Mohy · Nicholas J. Higham

Abstract We show that the Fréchet derivative of a matrix function f at A in the direction E, where A and E are real matrices, can be approximated by $\operatorname{Im} f(A+ihE)/h$ for some suitably small h. This approximation, requiring a single function evaluation at a complex argument, generalizes the complex step approximation known in the scalar case. The approximation is proved to be of second order in h for analytic functions f and also for the matrix sign function. It is shown that it does not suffer the inherent cancellation that limits the accuracy of finite difference approximations in floating point arithmetic. However, cancellation does nevertheless vitiate the approximation when the underlying method for evaluating f employs complex arithmetic. The ease of implementation of the approximation, and its superiority over finite differences, make it attractive when specialized methods for evaluating the Fréchet derivative are not available, and in particular for condition number estimation when used in conjunction with a block 1-norm estimation algorithm.

Keywords Fréchet derivative, matrix function, complex step approximation, complex arithmetic, finite difference, matrix sign function, condition number estimation, block 1-norm estimator

Mathematics Subject Classification (2000) 15A60, 65F30

1 Introduction

The Fréchet derivative of a matrix function $f : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ determines the sensitivity of f to small perturbations in the input matrix. The Fréchet derivative of f at $A \in \mathbb{C}^{n\times n}$ is a linear mapping

$$\begin{array}{ccc} \mathbb{C}^{n\times n} & \xrightarrow{L_f(A)} & \mathbb{C}^{n\times n} \\ E & \longmapsto & L_f(A,E) \end{array}$$

such that

$$f(A+E) - f(A) - L_f(A,E) = o(\|E\|) \tag{1.1}$$

Version of April 25, 2009.

School of Mathematics, The University of Manchester, Manchester, M13 9PL, UK (almohy@maths.manchester.ac.uk, http://www.maths.manchester.ac.uk/~almohy, higham@ma.man.ac.uk, http://www.ma.man.ac.uk/~higham). The work of the second author was supported by a Royal Society-Wolfson Research Merit Award and by Engineering and Physical Sciences Research Council grant EP/D079403.

for all $E \in \mathbb{C}^{n\times n}$. The norm of the Fréchet derivative yields a condition number of the matrix function f at A [12, Thm. 3.1]:

$$\operatorname{cond}(f,A) := \lim_{\epsilon\to 0} \sup_{\|E\|\le\epsilon\|A\|} \frac{\|f(A+E)-f(A)\|}{\epsilon\|f(A)\|} = \frac{\|L_f(A)\|\,\|A\|}{\|f(A)\|}, \tag{1.2}$$

where

$$\|L_f(A)\| := \max_{\|Z\|=1} \|L_f(A,Z)\| \tag{1.3}$$

and the norm is any matrix norm. When calculating $f(A)$, it is desirable to be able to efficiently estimate $\operatorname{cond}(f,A)$, and from (1.2) and (1.3) we see that this will in general required the evaluation of $L_f(A,Z)$ for certain Z. Thus it is important to be able to compute or estimate the Fréchet derivative reliably and efficiently. A natural approach is to approximate the Fréchet derivative by the finite difference

$$L_f(A,E) \approx \frac{f(A+hE)-f(A)}{h}, \tag{1.4}$$

for a suitably chosen h. This approach has the drawback that h needs to be selected to balance truncation errors with errors due to subtractive cancellation in floating point arithmetic, and as a result the smallest relative error that can be obtained is of order $u^{1/2}$, where u is the unit roundoff [12, Sec. 3.4].

In this work we pursue a completely different approach. Like (1.4), it requires one additional function evaluation, but now at a complex argument:

$$L_f(A,E) \approx \operatorname{Im} \frac{f(A+ihE)}{h}, \tag{1.5}$$

where $i = \sqrt{-1}$. This complex step (CS) approximation is known in the scalar case but has not, to our knowledge, been applied previously to matrix functions. The approximation requires A and E to be real and f to be real-valued at real arguments. The advantage of (1.5) over (1.4) is that in principal (1.5) allows h to be chosen as small as necessary to obtain an accurate approximation to $L_f(A,E)$, without cancellation errors contaminating the result in floating point arithmetic. It also provides an approximation to $f(A)$ from this single function evaluation. Unlike (1.4), however, it requires the use of complex arithmetic.

In Section 2 we review the complex step approximation for scalars. We extend the approximation to the matrix case in Section 3 and show that it is second order accurate for analytic functions f. The computational cost is considered in Section 4. In Section 5 we show that the CS approximation is also second order accurate for the matrix sign function, which is not analytic. In Section 6 we show that good accuracy can be expected in floating point arithmetic for sufficiently small h, but that if the method for evaluating f uses complex arithmetic then catastrophic cancellation is likely to vitiate the approximation. Finally, numerical experiments are given in Section 7 to illustrate the advantages of the CS approximation over finite differences, the role of the underlying method for evaluating f, and the application of the approximation to condition number estimation.

2 Complex step approximation: scalar case

For an analytic function $f : \mathbb{R} \to \mathbb{R}$, the use of complex arithmetic for the numerical approximation of derivatives of f was introduced by Lyness [19] and Lyness and Moler [20]. The earliest appearance of the CS approximation itself appears to be in Squire and Trapp [26]. Later uses of the formula appear in Kelley [15, Sec. 2.5.2] and Cox and Harris [3], while Martins, Sturdza, and Alonso [22] and Shampine [24] investigate the implementation of the approximation in high level languages.

The scalar approximation can be derived from the Taylor series expansion, with $x_0, h \in \mathbb{R}$,

$$f(x_0+ih) = \sum_{k=0}^{\infty} (ih)^k \frac{f^{(k)}(x_0)}{k!} = f(x_0) + ihf'(x_0) - h^2 \frac{f''(x_0)}{2!} + O(h^3). \quad (2.1)$$

Equating real and imaginary parts yields

$$f(x_0) = \operatorname{Re} f(x_0 + ih) + O(h^2), \qquad f'(x_0) = \operatorname{Im} \frac{f(x_0 + ih)}{h} + O(h^2). \quad (2.2)$$

Unlike in the finite difference approximation (1.4), subtractive cancellation is not intrinsic in the expression $\operatorname{Im} f(x_0 + ih)/h$, and this approximation therefore offers the promise of allowing h to be selected based solely on the need to make the truncation error sufficiently small. Practical experience reported in the papers cited above has indeed demonstrated the ability of (2.2) to produce accurate approximations, even with h as small as 10^{-100}, which is the value used in software at the National Physical Laboratory according to [3].

In next section we generalize the complex step approximation to real-valued matrix functions over $\mathbb{R}^{n\times n}$.

3 Complex step approximation: matrix case

Assuming that the Fréchet derivative is defined, replacing E by ihE in (1.1), where E is independent of h, and using the linearity of L_f, we obtain

$$f(A + ihE) - f(A) - ihL_f(A, E) = o(h).$$

Thus if $f : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ and $A, E \in \mathbb{R}^{n\times n}$ then

$$\lim_{h\to 0} \operatorname{Im} \frac{f(A + ihE)}{h} = L_f(A, E),$$

which justifies the CS approximation (1.5). However, this analysis does not reveal the rate of convergence of the approximation as $h \to 0$. To determine the rate we need a more careful analysis with stronger assumptions on f.

Denote by $L_f^{[j]}(A, E)$ the jth Fréchet derivative of f at A in the direction E, given by

$$L_f^{[j]}(A, E) = \frac{d^j}{dt^j} f(A + tE) \Big|_{t=0},$$

with $L_f^{[0]}(A,E) = f(A)$ and $L_f^{[1]} \equiv L_f$. The next result provides a Taylor expansion of f in terms of the Fréchet derivatives.

Theorem 3.1 *Let $f : \mathbb{C} \to \mathbb{C}$ have the power series expansion $f(x) = \sum_{k=0}^{\infty} a_k x^k$ with radius of convergence r. Let $A, E \in \mathbb{C}^{n\times n}$ such that $\rho(A+\mu E) < r$, where $\mu \in \mathbb{C}$. Then*

$$f(A+\mu E) = \sum_{k=0}^{\infty} \frac{\mu^k}{k!} L_f^{[k]}(A,E),$$

where

$$L_f^{[k]}(A,E) = \sum_{j=k}^{\infty} a_j L_{x^j}^{[k]}(A,E).$$

The Fréchet derivatives of the monomials satisfy the recurrence

$$L_{x^j}^{[k]}(A,E) = A L_{x^{j-1}}^{[k]}(A,E) + kE L_{x^{j-1}}^{[k-1]}(A,E). \tag{3.1}$$

Proof Najfeld and Havel [23, pp. 349–350] show that

$$(A+\mu E)^j = \sum_{k=0}^{j} \frac{\mu^k}{k!} L_{x^j}^{[k]}(A,E)$$

and that the $L_{x^j}^{[k]}$ satisfy the recurrence (3.1). By the assumption on the spectral radius, we have

$$\begin{aligned} f(A+\mu E) &= \sum_{j=0}^{\infty} a_j (A+\mu E)^j \\ &= \sum_{j=0}^{\infty} a_j \left(\sum_{k=0}^{j} \frac{\mu^k}{k!} L_{x^j}^{[k]}(A,E) \right) \\ &= \sum_{k=0}^{\infty} \frac{\mu^k}{k!} \sum_{j=k}^{\infty} a_j L_{x^j}^{[k]}(A,E). \end{aligned}$$

By the sum rule for Fréchet derivatives [12, Thm. 3.2], the inner summation in the last expression is $L_f^{[k]}(A,E)$. □

Replacing μ in Theorem 3.1 by ih, where $h \in \mathbb{R}$, we obtain

$$f(A+ihE) = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k)!} h^{2k} L_f^{[2k]}(A,E) + i \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)!} h^{2k+1} L_f^{[2k+1]}(A,E).$$

To be able to extract the desired terms from this expansion we need $f : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ and $A, E \in \mathbb{R}^{n\times n}$. Then

$$f(A) = \operatorname{Re} f(A+ihE) + O(h^2), \tag{3.2a}$$

$$L_f(A,E) = \operatorname{Im} \frac{f(A+ihE)}{h} + O(h^2). \tag{3.2b}$$

Theorem 3.1 can be used to develop approximations to higher Fréchet derivatives (cf. [18]), but we will not pursue this here.

The analyticity of f is sufficient to ensure a second order approximation, but it is not necessary. In Section 5 we consider the matrix sign function, which is not analytic, and show that the CS approximation error is nevertheless $O(h^2)$.

4 Cost analysis

The CS approximation has the major attraction that it is trivial to implement, as long as code is available for evaluating f at a complex argument. We now look at the computational cost, assuming that the cost of evaluating $f(A)$ is $O(n^3)$ flops, where a flop is a real scalar addition or multiplication.

First we note that the cost of multiplying two $n \times n$ real matrices is $2n^3$ flops. To multiply two $n \times n$ complex matrices requires $8n^3$ flops if complex scalar multiplications are done in the obvious way. However, by using a formula for multiplying two complex scalars in 3 real multiplications the cost can be reduced to $6n^3$ flops at the cost of weaker rounding error bounds [9], [10, Chap. 23]. For an algorithm for computing $f(A)$ whose cost is dominated by level 3 BLAS operations it follows [8], [14] that the cost of computing the CS approximation to $L_f(A, E)$ is 3–4 times that of the cost of computing $f(A)$ alone, though of course the CS approximation does yield approximations to both $f(A)$ and $L_f(A, E)$.

Next, we compare with another way of computing the Fréchet derivative, which is from the formula [12, Sec. 3.1]

$$f\left(\begin{bmatrix} A & E \\ 0 & A \end{bmatrix}\right) = \begin{bmatrix} f(A) & L_f(A,E) \\ 0 & f(A) \end{bmatrix}. \tag{4.1}$$

This formula requires the evaluation of f in real arithmetic at a $2n \times 2n$ matrix, which in principle is 8 times the cost of evaluating $f(A)$. However, it will usually be possible to reduce the cost by exploiting the block triangular, block Toeplitz structure of the argument. Hence this approach may be of similar cost to the CS approximation. Al-Mohy and Higham [1] note a drawback of (4.1) connected with the scaling of E. Since $L_f(A, \alpha E) = \alpha L_f(A, E)$ the norm of E can be chosen at will, but the choice may affect the accuracy of a particular algorithm based on (4.1) and it is difficult to know what is the optimal choice.

Another comparison can be made under the assumption that f is polynomial, which is relevant since a number of algorithms for evaluating $f(A)$ make use of polynomial or rational approximations. Let π_m be the number of matrix multiplications required to evaluate $f(A)$ by a particular scheme. Al-Mohy and Higham [1, Thm. 4.1] show that for a wide class of schemes the extra cost of computing $L_f(A, E)$ via the scheme obtained by differentiating the given scheme for $f(A)$ is at most $2\pi_m$ if terms formed during the evaluation of $f(A)$ are re-used. The CS approximation is therefore likely to be more costly, but it requires no extra coding effort and is not restricted to polynomials.

5 Sign function

The matrix sign function is an example of a function that is not analytic, so the analysis in Section 3 showing a second order error for the CS approximation is not applicable. We prove in this section that the CS approximation nevertheless has an error of second order in h for the sign function.

For $A \in \mathbb{C}^{n\times n}$ with no eigenvalues on the imaginary axis, $\operatorname{sign}(A)$ is the limit of the Newton iteration

$$X_{k+1} = \frac{1}{2}\left(X_k + X_k^{-1}\right), \qquad X_0 = A. \tag{5.1}$$

Moreover, the iterates E_k defined by

$$E_{k+1} = \frac{1}{2}\left(E_k - X_k^{-1}E_kX_k^{-1}\right), \qquad E_0 = E \tag{5.2}$$

converge to $L_{\operatorname{sign}}(A,E)$. Both iterations converge quadratically; see [12, Thms. 5.6, 5.7].

The next theorem uses these iterations to determine the order of the error of the CS approximation.

Theorem 5.1 *Let $A, E \in \mathbb{R}^{n\times n}$ and let A have no eigenvalues on the imaginary axis. In the iteration*

$$Z_{k+1} = \frac{1}{2}\left(Z_k + Z_k^{-1}\right), \qquad Z_0 = A + ihE, \qquad h \in \mathbb{R} \tag{5.3}$$

the Z_k are nonsingular for all h sufficiently small and

$$\begin{aligned}\operatorname{Re}\operatorname{sign}(A+ihE) &= \lim_{k\to\infty} \operatorname{Re} Z_k = \operatorname{sign}(A) + O(h^2),\\ \operatorname{Im}\operatorname{sign}(A+ihE) &= \lim_{k\to\infty} \operatorname{Im}\frac{Z_k}{h} = L_{\operatorname{sign}}(A,E) + O(h^2).\end{aligned}$$

Proof Write $Z_k = M_k + iN_k \equiv \operatorname{Re} Z_k + i\operatorname{Im} Z_k$. It suffices to show that

$$M_k = X_k + O(h^2), \qquad N_k = hE_k + O(h^3), \tag{5.4}$$

where X_k and E_k satisfy (5.1) and (5.2), which we prove by induction. First, set $k = 1$ and assume that $\rho(EA^{-1}) < 1/h$, which is true for sufficiently small h. Then we have the expansion

$$(A+ihE)^{-1} = A^{-1}\sum_{j=0}^{\infty}(-ih)^j(EA^{-1})^j.$$

Therefore the first iteration of (5.3) gives

$$M_1 = \frac{1}{2}\left(A + A^{-1}\right) + O(h^2), \qquad N_1 = \frac{h}{2}\left(E - A^{-1}EA^{-1}\right) + O(h^3),$$

so that (5.4) holds for $k = 1$. Suppose that (5.4) holds for k. Then we can write $M_k = X_k + h^2 R_k$, for some matrix $R_k \in \mathbb{R}^{n\times n}$. Assuming $\rho(R_k X_k^{-1}) < 1/h^2$, which again is true for sufficiently small h, we have

$$M_k^{-1} = X_k^{-1} \sum_{j=0}^{\infty} h^{2j} (-R_k X_k^{-1})^j = X_k^{-1} + O(h^2). \tag{5.5}$$

Now assume $\rho(N_k M_k^{-1}) < 1$, which is true for sufficiently small h since $N_k = O(h)$. Then, using (5.5) and $(M_k + iN_k)^{-1} = M_k^{-1} \sum_{j=0}^{\infty} (-i)^j (N_k M_k^{-1})^j$, we have

$$\begin{aligned}
M_{k+1} &= \frac{1}{2}\big(M_k + M_k^{-1} + M_k^{-1} \sum_{j=1}^{\infty} (-1)^j (N_k M_k^{-1})^{2j}\big) \\
&= \frac{1}{2}\left(X_k + X_k^{-1}\right) + O(h^2), \\
N_{k+1} &= \frac{1}{2}\big(N_k - M_k^{-1} N_k M_k^{-1} + M_k^{-1} \sum_{j=1}^{\infty} (-1)^{j+1} (N_k M_k^{-1})^{2j+1}\big) \\
&= \frac{h}{2}\left(E_k - X_k^{-1} E_k X_k^{-1}\right) + O(h^3),
\end{aligned}$$

which completes the induction. □

Note that another approach to proving Theorem 5.1 would be to use existing perturbation theory for the matrix sign function, such as that of Sun [27]. However, the perturbation expansions in [27] make use of the Schur and Jordan forms and do not readily permit the real and imaginary parts to be extracted.

The cost of evaluating the E_k in (5.2) is twice the cost of evaluating the X_k (assuming an LU factorization of X_k is computed for (5.1) and re-used). The CS approximation provides an approximation to $L_{\mathrm{sign}}(A, E)$ by iterating (5.1) with a complex starting matrix, so the cost is 3–4 times that for computing $\mathrm{sign}(A)$ alone. Given the ease of implementing (5.2) one would probably not use the CS approximation with the Newton iteration. However, with other methods for evaluating $\mathrm{sign}(A)$, of which there are many [12, Chap. 5], the economics may be different.

6 Accuracy

What accuracy can we expect from the CS approximation in floating point arithmetic? Equivalently, how accurately is the imaginary part of $f(A + ihE)$ computed when h is small, bearing in mind that the imaginary part is expected to be of order h, and hence much smaller than the real part? In order to obtain an accurate result it is necessary that the information contained in hE is accurately transmitted through to the imaginary part of $f(A + ihE)$, and this is most likely when the imaginary part does not undergo large growth and then reduction (due to subtractive cancellation) during the computation.

It is straightforward to show that the sum and product of two complex matrices with tiny imaginary part has tiny imaginary part and that the inverse of a matrix with tiny imaginary part has tiny imaginary part. It follows when a polynomial or rational function with real coefficients is evaluated at a matrix with tiny imaginary part the result has tiny imaginary part. Hence when we evaluate $f(A+ihE)$ using an algorithm for f based on polynomial or rational approximations with real coefficients there is no a priori reason to expect damaging cancellation within the imaginary part. In particular, there is no a priori lower bound on the accuracy that can be expected, unlike for the finite difference approximation (1.4), for which such a lower bound is of order $u^{1/2}$.

However, numerical instability is possible if the algorithm for $f(A)$ itself employs complex arithmetic, as we now show. Suppose we compute $C = \cos(A)$ by the simple algorithm [12, Alg 12.7]

$$X = e^{iA}, \tag{6.1a}$$

$$C = (X + X^{-1})/2. \tag{6.1b}$$

The CS approximation gives

$$L_{\cos}(A,E) \approx \operatorname{Im} \frac{\cos(A+ihE)}{h} = \operatorname{Im} \frac{e^{iA-hE} + e^{-iA+hE}}{2h}. \tag{6.2}$$

Making the simplifying assumption that A and E commute, in which case $L_{\cos}(A,E) = -E\sin(A)$ [12, Sec. 12.2], we have

$$\begin{aligned} e^{iA-hE} + e^{-iA+hE} &= e^{iA}e^{-hE} + e^{-iA}e^{hE} \\ &= \cos(A)(e^{-hE} + e^{hE}) + i\sin(A)(e^{-hE} - e^{hE}), \end{aligned}$$

and the CS approximation reduces to

$$L_{\cos}(A,E) \approx \frac{\sin(A)(e^{-hE} - e^{hE})}{2h}.$$

Thus $-E$ is being approximated by $(e^{-hE} - e^{hE})/(2h)$, and the latter expression suffers massive subtractive cancellation for small h. We illustrate in Figure 6.1 with A and E both the scalar 1. These computations, and those in the next section, were performed in MATLAB R2009a, with unit roundoff $u = 2^{-53} \approx 1.1 \times 10^{-16}$. Note that the CS approximation deteriorates once h decreases below 10^{-5}, yielding maximum accuracy of about 10^{-10}. This weakness of the CS approximation for scalar problems is noted by Martins, Sturdza, and Alonso [21].

The unwanted resonance between complex arithmetic in the underlying algorithm and the pure imaginary perturbation used by the CS approximation affects any algorithm based on the Schur form, such as those in [4], [6], [17]. Since $B := A + ihE$ is nonreal, the complex Schur form $B = QTQ^*$, with Q unitary and T upper triangular, must be used. In general, Q will have real and imaginary parts of similar norm (since A may have some nonreal eigenvalues), and likewise for T. The $O(h)$ imaginary part of $f(B) = Qf(T)Q^*$ is therefore the result of massive cancellation, which signals a serious loss of accuracy of the CS approximation in floating point arithmetic. The first experiment in the next section illustrates this phenomenon.

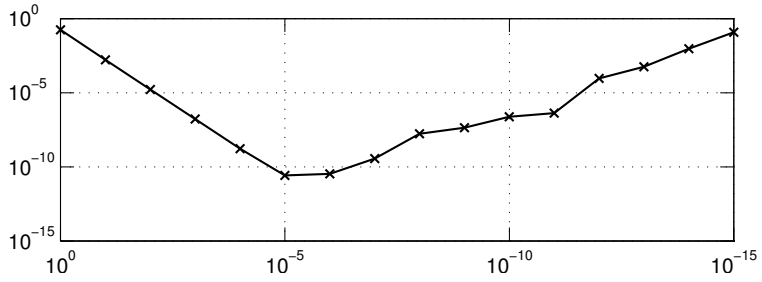

Fig. 6.1 Relative errors for approximating $L_{\cos}(A, E)$ for scalars $A = E = 1$ using the CS approximation with (6.2) and $h = 10^{-k}$, $k = 0\colon 15$.

7 Numerical experiments

We now give some experiments to illustrate the CS approximation and its advantage over the finite difference approximation (1.4).

For our first experiment we take `A = gallery('triw',10)`, the unit upper triangular matrix with every superdiagonal element equal to -1, and a random matrix `E = randn(10)`. The function is $f(A) = e^A$, which we compute both by MATLAB's `expm`, which implements the scaling and squaring method [11], and by MATLAB's `funm`, which handles general matrix functions via the Schur–Parlett method [4] and treats the diagonal Schur form blocks specially in the case of the exponential.

For h ranging from 10^{-3} to 10^{-20}, Figure 7.1 plots the normwise relative error $\|L_{\exp}(A, E) - \widehat{L}\|_1 / \|L_{\exp}(A, E)\|_1$, where $\widehat{L}$ represents the approximate Fréchet derivative from the finite-difference approximation (1.4) or the CS approximation (1.5). The "exact" $L_{\exp}(A, E)$ is obtained via the relation (4.1) evaluated at 100 digit precision using MATLAB's Symbolic Math Toolbox.

In this example the CS approximation has full accuracy when using `expm` with $h \le 10^{-8}$, reflecting its $O(h^2)$ error (see (3.2b)). By contrast, the finite difference approximation returns its best result at around $h = 10^{-8}$ with error of $O(h)$, and then diverges as h decreases, just as the theory on the choice of h suggests [12, Sec. 3.4]. Equipping the CS approximation with `funm` leads to poor results, due to the complex arithmetic inherent in the Schur form (see Section 6), though the results are superior to those obtained with finite differences. Note that the fact that A has real eigenvalues does not help: as a result of A being highly nonnormal (indeed defective, with a single Jordan block), the perturbed matrix $B = A + ihE$ has eigenvalues with imaginary parts of order 10^{-2} for all the chosen h!

Interestingly, the error for the CS approximation with `expm` remains roughly constant at around 10^{-16} for h decreasing all the way down to 10^{-292}, at which point it starts to increase, reaching an error of 10^{-1} by the time h underflows to zero at around 10^{-324}.

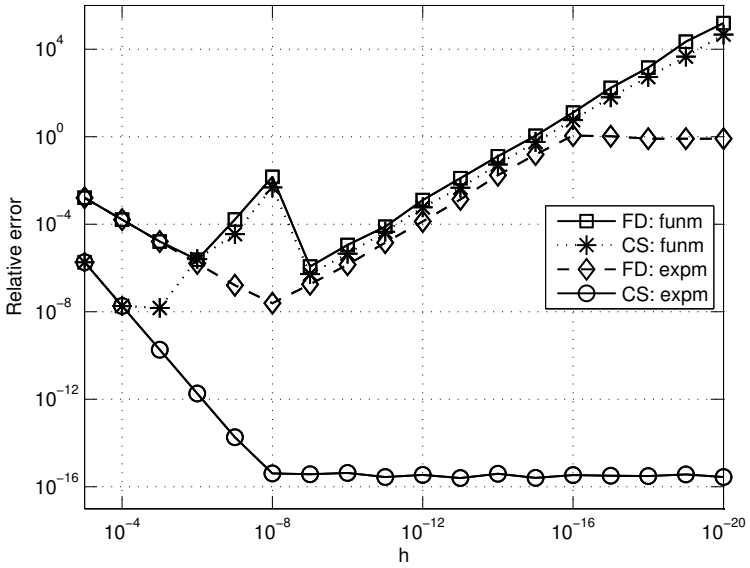

Fig. 7.1 Relative errors for approximating $L_{\exp}(A, E)$ using the CS approximation and the finite difference (FD) approximation (1.4), for $h = 10^{-k}$, $k = 3\colon 20$.

The performance of the CS approximation is of course method-dependent. Figure 7.2 repeats the previous experiment except that we set $a_{15} = 10^6$ and compare `expm` with `expm2`, the latter function using an improved scaling and squaring algorithm of Al-Mohy and Higham [2] designed to avoid overscaling. The large off-diagonal element of A causes `expm` to overscale in its scaling phase, that is, to reduce $\|A\|$ much further than necessary in order to achieve an accurate result: the relative error of the computed exponentials is of order 10^{-11} for `expm` and 10^{-16} for `expm2`. We see from Figure 7.2 that there is a corresponding difference in the accuracy of the Fréchet derivative approximations. But the superiority of the CS approximation over the finite difference approximation is evident for both `expm` and `expm2`.

Our next experiment involves a different function: the principal matrix square root, $A^{1/2}$. The Fréchet derivative $L_{\mathrm{sqrt}}(A, E)$ is the solution L of $LX + XL = E$, where $X = A^{1/2}$ [12, Sec. 6.1]. The product form of the Denman–Beavers iteration,

$$M_{k+1} = \frac{1}{2}\left(I + \frac{M_k + M_k^{-1}}{2}\right), \qquad M_0 = A, \tag{7.1}$$

$$X_{k+1} = \frac{1}{2}X_k(I + M_k^{-1}), \qquad X_0 = A, \tag{7.2}$$

is a variant of the Newton iteration, and $M_k \to I$ and $X_k \to A^{1/2}$ quadratically as $k \to \infty$ [12, Sec. 6.3]. With A the 8×8 Frank matrix (`gallery('frank',8)`),

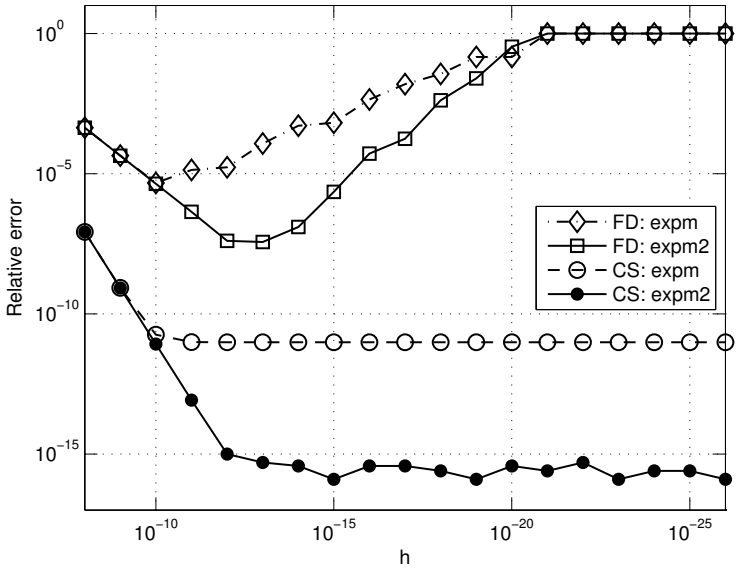

Fig. 7.2 Relative errors for approximating $L_{\exp}(A, E)$ using the CS approximation and the finite difference (FD) approximation (1.4), for $h = 10^{-k}/\|A\|_1$, $k = 2\colon 21$.

which has positive real eigenvalues, and `E = randn(8)`, we apply the CS approximation using this iteration as the means for evaluating $(A + ihE)^{1/2}$. Figure 7.3 shows the results, along with the errors from finite differencing. Again we see second order convergence of the CS approximations, which follows from the analyticity of the square root. Note, however, that the minimal relative error of order $10^4 u$. Since $\|L_{\mathrm{sqrt}}(A, E)\|_1 \approx 9 \times 10^3$ this is consistent with the maxim that the condition number of the condition number is the condition number [5].

Our final experiment illustrates the use of the CS approximation in condition estimation. As (1.2) shows, to estimate $\mathrm{cond}(f, A)$ we need to estimate $\|L_f(A)\|$, and this can be done by applying a matrix norm estimator to the Kronecker matrix form $K_f(A) \in \mathbb{C}^{n^2 \times n^2}$ of $L_f(A)$, defined by $\mathrm{vec}(L_f(A, E)) = K_f(A)\,\mathrm{vec}(E)$, where vec is the operator that stacks the columns of a matrix into one long vector [12, Chap. 3]. We will use the block 1-norm estimation algorithm of Higham and Tisseur [13], which requires the ability to form matrix products $K_f y \equiv \mathrm{vec}(L_f(A, E)$ and $K_f^T y \equiv \mathrm{vec}(L_f(A, E^T)^T)$, where $\mathrm{vec}(E) = y$ (where we are assuming $A \in \mathbb{R}^{n\times n}$ and $f : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$). We use a modified version of the function `funm_condest1` from the Matrix Function Toolbox [7], which interfaces to the MATLAB function `normest1` that implements the block 1-norm estimation algorithm. With f the exponential, evaluated by `expm`, we approximate the Fréchet derivative using three different approaches: the CS approximation, finite differences, and the method from

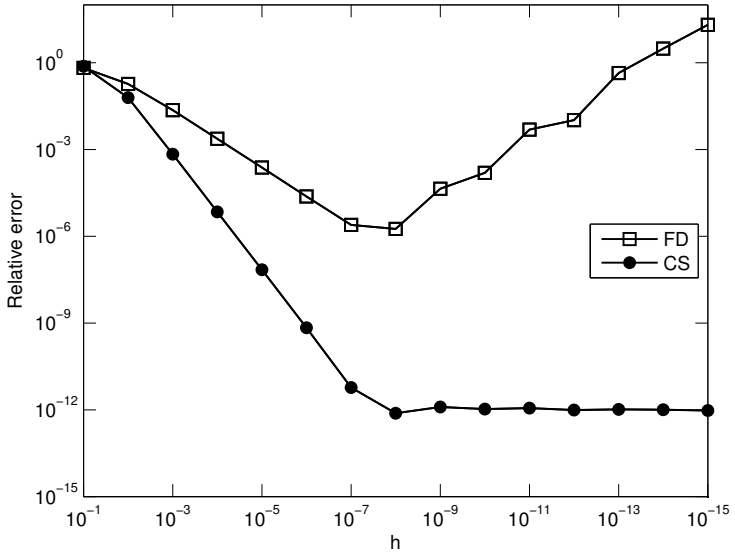

Fig. 7.3 Relative errors for approximating $L_{\mathrm{sqrt}}(A, E)$ using the CS approximation and the finite difference (FD) approximation (1.4) with the product form of the Denman–Beavers iteration, for $h = 10^{-k}/\|A\|_1$, $k = 1\colon 15$.

[1], which is a specialized method based on scaling and squaring and Padé approximation. We take a collection of 28 real matrices from the literature on methods for e^A, which are mostly ill conditioned or badly scaled and are all of dimension 10 or less. For the finite difference approximation (1.4) we take the value $h = (u\|f(A)\|_1)^{1/2}/\|E\|_1$, which is optimal in the sense of balancing truncation error and rounding error bounds [12, Sec. 3.4]. For the CS approximation we take $h = \mathrm{tol}\|A\|_1/\|E\|_1$ with $\mathrm{tol} = u^2$; we found that for $\mathrm{tol} = u^{1/2}$ and $\mathrm{tol} = u$ the estimates were sometimes very poor on the most badly scaled problems. The exact $\|K_f(A)\|_1$ is obtained by explicitly computing $K_f(A)$ using `expm_cond` from the Matrix Function Toolbox [7]. The ratios of the estimate divided by $\|K_f(A)\|_1$ are shown in Figure 7.4. All should be at most 1, so a value larger than 1 is a sign of inaccurate Fréchet derivative approximations. The results show that the condition estimates obtained with the CS approximation are significantly more reliable than those from finite differences (one estimate of the wrong order of magnitude as opposed to four), but that neither is as reliable as when the Fréchet derivative is computed by a method specialized to the exponential.

8 Concluding remarks

The CS approximation provides an attractive way to approximate Fréchet derivatives L_f when specialized methods for f but not L_f are available. This

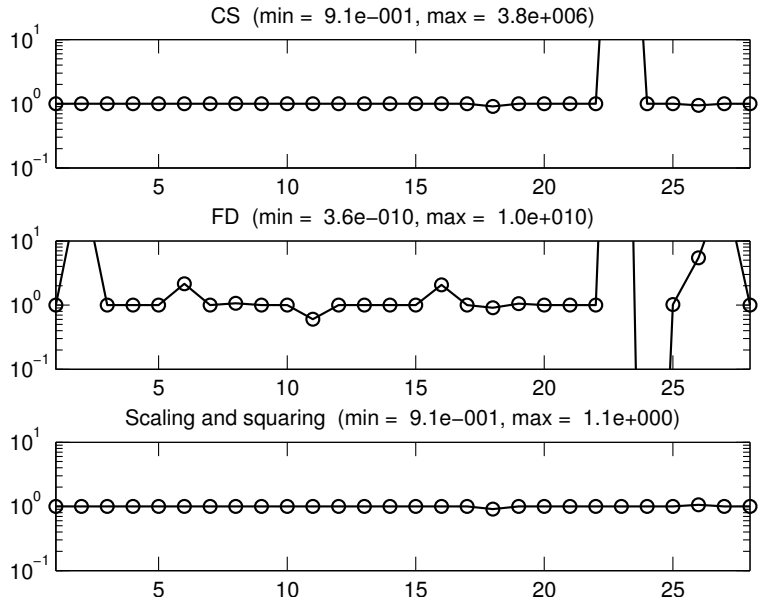

Fig. 7.4 Ratios of estimate of $\|K_f(A)\|_1$ divided by true value for $f(A) = e^A$, computed using a block 1-norm estimator, where the Fréchet derivative is approximated by the CS approximation, the finite difference (FD) approximation (1.4), and a scaling and squaring method.

situation pertains, for example, to the functions $\psi_k(z) = \sum_{j=0}^{\infty} z^j/(j+k)!$, $k \geq 0$ [12, Sec. 10.7.4], related to the matrix exponential, for which efficient numerical methods are available [16], [25]. The CS approximation is trivial to implement assuming the availability of complex arithmetic. In floating point arithmetic its accuracy is not limited by the cancellation inherent in finite difference approximations, and indeed the accuracy of the computed approximation is in practice remarkably insensitive to the choice of the parameter h, as long as it is chosen small enough: typically $h \leq u^{1/2}\|A\|/\|E\|$ suffices thanks to the $O(h^2)$ approximation error.

The main weakness of the CS approximation is that it is prone to damaging cancellation when the underlying method for evaluating f employs complex arithmetic. But for many algorithms, such as those based on real polynomial and rational approximations or matrix iterations, this is not a concern.

The CS approximation is particularly attractive for use within a general purpose matrix function condition estimator. We intend to update the function `funm_condest1` in the Matrix Function Toolbox [7] to augment the current finite difference approximation with the CS approximation, which will be the preferred option when it is applicable.

References

1. Al-Mohy, A.H., Higham, N.J.: Computing the Fréchet derivative of the matrix exponential, with an application to condition number estimation. SIAM J. Matrix Anal. Appl. **30**(4), 1639–1657 (2009)
2. Al-Mohy, A.H., Higham, N.J.: A new scaling and squaring algorithm for the matrix exponential. MIMS EPrint 2009.9, Manchester Institute for Mathematical Sciences, The University of Manchester, UK (2009)
3. Cox, M.G., Harris, P.M.: Numerical analysis for algorithm design in metrology. Software Support for Metrology Best Practice Guide No. 11, National Physical Laboratory, Teddington, UK (2004)
4. Davies, P.I., Higham, N.J.: A Schur–Parlett algorithm for computing matrix functions. SIAM J. Matrix Anal. Appl. **25**(2), 464–485 (2003)
5. Demmel, J.W.: On condition numbers and the distance to the nearest ill-posed problem. Numer. Math. **51**, 251–289 (1987)
6. Guo, C.H., Higham, N.J.: A Schur–Newton method for the matrix pth root and its inverse. SIAM J. Matrix Anal. Appl. **28**(3), 788–804 (2006)
7. Higham, N.J.: The Matrix Function Toolbox. `http://www.ma.man.ac.uk/~higham/mftoolbox`
8. Higham, N.J.: Exploiting fast matrix multiplication within the level 3 BLAS. ACM Trans. Math. Software **16**(4), 352–368 (1990)
9. Higham, N.J.: Stability of a method for multiplying complex matrices with three real matrix multiplications. SIAM J. Matrix Anal. Appl. **13**(3), 681–687 (1992)
10. Higham, N.J.: Accuracy and Stability of Numerical Algorithms. Second edn. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2002)
11. Higham, N.J.: The scaling and squaring method for the matrix exponential revisited. SIAM J. Matrix Anal. Appl. **26**(4), 1179–1193 (2005)
12. Higham, N.J.: Functions of Matrices: Theory and Computation. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2008)
13. Higham, N.J., Tisseur, F.: A block algorithm for matrix 1-norm estimation, with an application to 1-norm pseudospectra. SIAM J. Matrix Anal. Appl. **21**(4), 1185–1201 (2000)
14. Kågström, B., Ling, P., Van Loan, C.F.: GEMM-based level 3 BLAS: High performance model implementations and performance evaluation benchmark. ACM Trans. Math. Software **24**(3), 268–302 (1998)
15. Kelley, C.T.: Solving Nonlinear Equations with Newton's Method. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2003)
16. Koikari, S.: An error analysis of the modified scaling and squaring method. Computers Math. Applic. **53**, 1293–1305 (2007)
17. Koikari, S.: Algorithm 894: On a block Schur–Parlett algorithm for φ-functions based on the sep-inverse estimate. ACM Trans. Math. Software **36**(2), Article 12 (2009)
18. Lai, K.L., Crassidis, J.L.: Extensions of the first and second complex-step derivative approximations. J. Comput. Appl. Math. **219**, 276–293 (2008)
19. Lyness, J.N.: Numerical algorithms based on the theory of complex variable. In: Proceedings of the 1967 22nd National Conference, pp. 125–133. ACM, New York, NY, USA (1967)
20. Lyness, J.N., Moler, C.B.: Numerical differentiation of analytic functions. SIAM J. Numer. Anal. **4**(2), 202–210 (1967)
21. Martins, J.R.R.A., Sturdza, P., Alonso, J.J.: The connection between the complex-step derivative approximation and algorithmic differentiation (2001). AIAA paper AIAA-2001-0921
22. Martins, J.R.R.A., Sturdza, P., Alonso, J.J.: The complex-step derivative approximation. ACM Trans. Math. Software **29**(3), 245–262 (2003)
23. Najfeld, I., Havel, T.F.: Derivatives of the matrix exponential and their computation. Advances in Applied Mathematics **16**, 321–375 (1995)
24. Shampine, L.F.: Accurate numerical derivatives in MATLAB. ACM Trans. Math. Software **33**(4) (2007). Article 26, 17 pages

25. Skaflestad, B., Wright, W.M.: The scaling and modified squaring method for matrix functions related to the exponential. Appl. Numer. Math. **59**, 783–799 (2009)
26. Squire, W., Trapp, G.: Using complex variables to estimate derivatives of real functions. SIAM Rev. **40**(1), 110–112 (1998)
27. Sun, J.: Perturbation analysis of the matrix sign function. Linear Algebra Appl. **250**, 177–206 (1997)