SIAM J. MATRIX ANAL. APPL.
Vol. 37, No. 2, pp. 719–743

MATRIX ARITHMETIC-GEOMETRIC MEAN AND THE COMPUTATION OF THE LOGARITHM*

JOÃO R. CARDOSO† AND RUI RALHA‡

Abstract. We investigate the stability of the matrix arithmetic-geometric mean (AGM) iteration. We show that the classical formulation of this iteration may be not stable (a necessary and sufficient condition for its stability is given) and investigate the numerical properties of alternative formulations. It turns out that the so-called Legendre form is the right choice for matrices. Due to its fast convergence and good numerical properties, our AGM formulation has the potential to play an important role in the computation of matrix functions. In fact, we developed an algorithm, whose main block is an optimized AGM scheme, for the computation of the logarithm of a matrix, which is shown to be competitive, in terms of accuracy, with the state-of-the-art methods. Methods that do not require an initial reduction to the Schur form are potentially more efficient on parallel computers. For this reason, our current implementation does not include such reduction and operates with full matrices till the end. As compared to the state-of-the-art reduction free algorithm, our method relies more heavily on matrix multiplications, which are highly suited to modern architectures, and requires a smaller number of multiple right-hand-side linear systems, making it competitive also in terms of computational efficiency. Our claims are supported with analysis and also with numerical results produced with a MATLAB code.

Key words. arithmetic-geometric mean iteration, Fréchet derivative, stability, convergence, matrix logarithm, matrix square root, elliptic integrals

AMS subject classifications. 65F60, 65F30, 33E05

DOI. 10.1137/140998226

1. Introduction.

The arithmetic-geometric mean (AGM) of two positive real numbers, or two complex numbers with positive real parts, first appeared in a paper by Lagrange but it was Gauss who studied it in depth and discovered many of its important properties as well as interesting connections with elliptic integrals. One of the reasons for their interest in the AGM was the need for accurately calculating the perimeter of an ellipse and in turn the elliptical orbit of planets. In modern times, it has been used successfully in fast and high precision computations of many elementary functions such as $\log x$, e^x, $\cos x$, and $\sin x$. For details on theoretical and computational issues on this iteration we refer the reader to [7, 6, 8, 9, 13] and the references therein.

The algorithm for the scalar case generalizes in a straightforward way to matrices. Given a square matrix A with eigenvalues having positive real parts, the matrix AGM iteration is defined by

$$\begin{aligned} A_{k+1} &= \frac{A_k + B_k}{2}, \qquad A_0 = I, \\ B_{k+1} &= (A_k\, B_k)^{1/2}, \quad B_0 = A, \end{aligned} \tag{1.1}$$

*Received by the editors December 1, 2014; accepted for publication (in revised form) by A. Frommer March 29, 2016; published electronically June 2, 2016.
http://www.siam.org/journals/simax/37-2/99822.html

†Polytechnic Institute of Coimbra/ISEC, Rua Pedro Nunes, 3030-199 Coimbra, Portugal, and Institute of Systems and Robotics, University of Coimbra, Pólo II, 3030-290 Coimbra, Portugal (jocar@isec.pt).

‡Center of Mathematics, School of Sciences, University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal (r_ralha@math.uminho.pt).

where $k = 0, 1, 2, \ldots$, and $X^{1/2}$ denotes the principal matrix square root. It was shown in [32] that sequences (A_k) and (B_k) have a common limit, which is denoted by $\mathrm{AGM}(A)$, and that the sequence defined by $C_k := A_k - B_k$ converges quadratically to the zero matrix.

The iteration (1.1) is the "natural" procedure for computing the AGM of two commuting matrices with eigenvalues having positive real parts and it has been used by Stickel [32] in the computation of the matrix logarithm. However, Stickel did not investigate the stability of the procedure (1.1) and, as far as we know, we have been the first to deal with the stability issue. There is also a variant of the method of [32] proposed by the same author in [33] but apparently it offers no advantage. It is based on a modified AGM iteration, with convergence of fourth order, but each iterative step is much more expensive than a step of the standard AGM. Moreover, it involves fourth roots instead of square roots.

To investigate the stability of a given matrix iterative scheme, we proceed as in [19, sect. 5] (see also [11, 20]), where the analysis is carried out by means of the Fréchet derivative. In this work, we characterize the spectrum of the Fréchet derivative operator associated with iteration (1.1) and derive a necessary and sufficient condition for such iteration to be stable, using the eigenvalues of $\mathrm{AGM}(A)$. It is shown in particular that (1.1) may be not stable for some matrices with nonreal eigenvalues. To overcome this drawback, we propose stable versions and discuss their convergence and efficient computation. It comes out that the so-called Legendre iterative formula for AGM has many attractive features that make it a good candidate for an efficient computation of the AGM. Furthermore, it allows the formulation of the AGM for any matrix with no eigenvalues on $\mathbb{R}_0^-$.

Other extensions of the scalar AGM iteration to matrices and operators can be found in the literature. Commutativity is not assumed in general, but it is required that the matrices involved are symmetric positive definite; in the case of operators, the request is that they are positive definite, linear self-adjoint, and bounded in Hilbert spaces (see, for instance, [12, 29, 3]). It is also worth mentioning that theoretical and numerical aspects of geometric means of positive definite matrices have been an important focus of research in recent years (see, for instance, [5] and the references therein).

The organization of the paper is as follows. In section 2 we recall some basic facts concerning the scalar AGM iteration, including some of its connections with elliptic integrals and logarithms. The theoretical convergence of the standard matrix AGM iteration is addressed in section 3 and its stability analysis is carried out in section 4 (a necessary and sufficient condition upon the eigenvalues of the limit matrix is given). In section 5 new AGM formulations are presented and shown to be stable; a new result on global convergence of AGM is also presented in this section. In section 6 we explain why the Legendre form is the right choice for the matrix AGM. In section 7 we discuss implementation issues; in particular we manage to replace the last iterations of the AGM with the computation of a few terms of a Taylor expansion, a key point for the efficiency of our algorithm. In section 8 we present the algorithm that computes the logarithm of A from $\mathrm{AGM}(\epsilon A)$, for a carefully chosen small ϵ; a comparison of this algorithm with the state-of-the-art methods appears in section 8.1 for the computational efficiency and in section 8.2 for the numerical accuracy. Finally, in section 9 some conclusions are drawn.

Notation. $\|.\|$ denotes a consistent matrix or operator norm; $\|.\|_F$ and $\|.\|_2$ stand for the Frobenius norm and the 2-norm, respectively.

2. Revisiting the scalar AGM iteration. An interesting historical account of the deep work of Gauss in the scalar AGM, in particular the relation of AGM with other fields of mathematics, may be found in [13]. Most of the material of this section was taken from [7, Chap. 1], [9, sect. 4.8] and we omit proofs and other technical details. See also [28, 32]. Let $a^{1/2}$ denote the principal square root of a complex number a and let $\mathrm{Re}(a)$ denote its real part. Assume that z and w are complex numbers with positive real parts. Then the sequences (a_k) and (b_k) defined recursively by

$$\begin{aligned} a_{k+1} &= \frac{a_k + b_k}{2}, \quad a_0 = z, \\ b_{k+1} &= (a_k\, b_k)^{1/2}, \quad b_0 = w, \end{aligned} \tag{2.1}$$

with $k = 0, 1, 2, \ldots$, converge to a common limit called the AGM of z and w which is denoted by $\mathrm{AGM}(z, w)$, that is, $\mathrm{AGM}(z, w) = \lim_{k\to\infty} a_k = \lim_{k\to\infty} b_k$. Note that the restrictions $\mathrm{Re}(z) > 0$ and $\mathrm{Re}(w) > 0$ ensure that $a_k b_k$ belongs to the domain of analyticity of the principal square root $\mathbb{C}\backslash]-\infty, 0]$. The convergence of (2.1) is quadratic in the sense that the sequence (c_k) defined by $c_k := a_k - b_k$ converges quadratically to zero. The iteration is homogeneous, that is, $\mathrm{AGM}(\lambda z, \lambda w) = \lambda\ \mathrm{AGM}(z, w)$, where $\mathrm{Re}(\lambda) > 0$, and, consequently, $\mathrm{AGM}(z, w) = z\ \mathrm{AGM}(1, z^{-1}w)$. Thus, in most cases, we can regard AGM as a function depending on a single variable. We will write simply $\mathrm{AGM}(z)$ instead of $\mathrm{AGM}(1, z)$ as meaning the complex valued function that assigns, to each z such that $\mathrm{Re}(z) > 0$, the limit given by (2.1) for $a_0 = 1$ and $b_0 = z$. An alternative way to express the AGM is

$$\mathrm{AGM}(1, z) = \prod_{n=0}^{\infty} \frac{1}{2}\,(1 + \kappa_n)\,, \tag{2.2}$$

where $\kappa_0 = z$ and

$$\kappa_{n+1} = \frac{2\sqrt{\kappa_n}}{1 + \kappa_n}. \tag{2.3}$$

This is the so-called Legendre form of the AGM (see, for instance, [7, p. 3]) and will play an important role in our work.

One important property of $\mathrm{AGM}(z)$ is that it is analytic in the open right half plane. Figure 1 gives an illustration of the values of $|\,\mathrm{AGM}(z)|$ on a rectangle.

We now recall some interesting connections between the AGM and complete elliptic integrals of the first kind. The standard notation for these integrals is:

$$K(z) := \int_0^{\pi/2} \frac{d\theta}{\sqrt{1 - z^2 \sin^2\theta}} \quad \text{and} \quad K'(z) := \int_0^{\pi/2} \frac{d\theta}{\sqrt{1 - (1 - z^2)\sin^2\theta}}.$$

Here one assumes that $|z| < 1$. The variable z is known as the *modulus* and $z' := \sqrt{1 - z^2}$ as the *complementary modulus*. Note that here the "prime" notation does not stand for derivatives. It is easy to observe that $K(z') = K'(z)$. The identity

$$\frac{1}{\mathrm{AGM}(z)} = \frac{2}{\pi} K'(z), \tag{2.4}$$

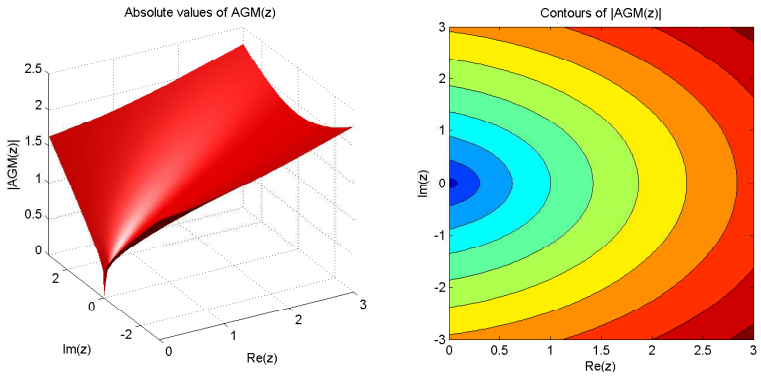

FIG. 1. *Absolute values of the* AGM *(left) and contours (right) for complex numbers in the rectangle* $0 \le x \le 3,\ -3 \le y \le 3$.

which was discovered by Gauss, allows the computation of the elliptic integral $K'(z)$ by the AGM iteration. These elliptic integrals are related with the logarithm by

$$\begin{aligned} K'(z) &= \frac{2}{\pi}\log\left(\frac{4}{z}\right)K(z) \\ &\quad -2\left[\left(\frac{1}{2}\right)^2\left(\frac{1}{1\times 2}\right)z^2 + \left(\frac{1\times 3}{2\times 4}\right)^2\left(\frac{1}{1\times 2}+\frac{1}{3\times 4}\right)z^4\right. \\ &\quad \left. + \left(\frac{1\times 3\times 5}{2\times 4\times 6}\right)^2\left(\frac{1}{1\times 2}+\frac{1}{3\times 4}+\frac{1}{5\times 6}\right)z^6+\cdots\right] \end{aligned} \tag{2.5}$$

(see [7, (1.3.10)]), which in turn allows a connection between the AGM and the logarithm:

$$\frac{\pi}{2\,\mathrm{AGM}(z)} = \log\left(\frac{4}{z}\right)\left(1+O(z^2)\right).$$

The representation (2.5) of $K'(z)$ combined with (2.4) has been used to derive algorithms for the computation of the logarithm by the AGM iteration. Since $K(z)$ can be represented in terms of a hypergeometric series [7, (1.3.6)]

$$K(z) = \frac{\pi}{2}F\left(\frac{1}{2},\frac{1}{2};1;z^2\right) = \frac{\pi}{2}\sum_{i=0}^{\infty}\left(\frac{\left(\frac{1}{2}\right)_i}{i!}\right)^2 z^{2i},$$

where $|z|<1$ and the symbol $(a)_i$ denotes the Pochhammer symbol (or rising factorial), which is defined by $(a)_0 = 1$ and $(a)_i := a(a+1)\dots(a+i-1)$ for any integer $i \ge 1$, from (2.5) some calculation leads to the series expansion

$$K'(z) = \log\left(\frac{4}{z}\right) + \sum_{i=1}^{\infty}\left(\frac{\left(\frac{1}{2}\right)_i}{i!}\right)^2\left(\log\left(\frac{4}{z}\right) - \sum_{j=0}^{i-1}\frac{2}{(2j+1)(2j+2)}\right)z^{2i}. \tag{2.6}$$

3. Convergence of the matrix AGM iteration. Let $A \in \mathbb{R}^{n\times n}$ have eigenvalues on the open right half plane, that is, $\mathrm{Re}(\lambda) > 0$, for all $\lambda \in \sigma(A)$; $\sigma(A)$ stands for the spectrum of A. Let (A_k) and (B_k) be the sequences of matrices defined

in (1.1). It is easy to show by induction that for each $k = 1, 2, \ldots$, A_k and B_k commute. This property is very important to guarantee that the recurrence (1.1) is well defined. In particular, it ensures that $\sigma(A_k B_k) \subset \{\lambda\mu : \ \lambda \in \sigma(A_k),\ \mu \in \sigma(B_k)\}$, which allows one to conclude that the eigenvalues of $A_k B_k$ do not lie on the closed negative real axis and that the principal square root $(A_k B_k)^{1/2}$ is well defined. Besides, the eigenvalues of A_k and B_k have positive real parts. Arguments based on the uniform convergence of (2.1) and the analyticity of $\mathrm{AGM}(z)$ are used in [32] to show that (A_k) and (B_k) always converge to a common limit $\mathrm{AGM}(A)$, that is, $\mathrm{AGM}(A) := \lim_{k\to\infty} A_k = \lim_{k\to\infty} B_k$. In section 5 we give our own proof of this global convergence. Note that all the iterates A_k and B_k are functions of A in the usual sense (see [20, Chap. 1]) and the same holds for their limit $\mathrm{AGM}(A)$. Therefore, the eigenvalues of $\mathrm{AGM}(A)$ are of the form $\mathrm{AGM}(\lambda)$, where $\lambda \in \sigma(A)$, and can be computed by means of the scalar AGM iteration. Moreover, since A is assumed to have eigenvalues with positive real parts, the same holds for the eigenvalues of $\mathrm{AGM}(A)$.

Another issue discussed in [32] is the quadratic convergence of the sequence defined by $C_k = A_k - B_k$ to the zero matrix. Theorem 3.1 below provides a different approach for understanding the behavior of (1.1) and also proves the quadratic convergence.

THEOREM 3.1. *Assume that* (A_k) *and* (B_k) *are the sequences defined in* (1.1) *and that* $C_k = A_k - B_k$. *Then, for any consistent matrix norm,*

$$\lim_{k\to\infty} \frac{\|C_{k+1}\|}{\|C_k\|^2} = \alpha,$$

where $\alpha \le \frac{1}{8}\left\|M^{-1}\right\|$ *with* $M := \mathrm{AGM}(A)$.

Proof. For any $k \ge 1$, we have, on one hand,

$$C_{k+1} = A_{k+1} - B_{k+1} = \frac{A_k + B_k}{2} - (A_k B_k)^{1/2} = \frac{1}{2}\left(A_k^{1/2} - B_k^{1/2}\right)^2,$$

and, on the other hand, $C_k = A_k - B_k = (A_k^{1/2} - B_k^{1/2})(A_k^{1/2} + B_k^{1/2})$. This gives $C_k^2 (A_k^{1/2} + B_k^{1/2})^{-2} = 2C_{k+1}$ and

$$\|C_{k+1}\| \le \frac{1}{2}\|C_k\|^2 \left\| \left(A_k^{1/2} + B_k^{1/2}\right)^{-2} \right\|. \tag{3.1}$$

Since A_k and B_k converge both to M, the result follows (we should point out that the same result follows from the relation (3) in [26]). ∎

Looking at the bound given by $\alpha \le \frac{1}{8}\left\|M^{-1}\right\|$, one may fear that convergence is dramatically slowed down when $M = \mathrm{AGM}(A)$ is close to being singular. Fortunately, the eigenvalues of M will not get too small, even when A has tiny eigenvalues (all with positive real part). Let λ be such a tiny eigenvalue of A. We note that to produce $\mathrm{AGM}(1, \lambda)$, one single scalar iteration of (2.1) with $a_0 = 1$ and $b_0 = \lambda$ will produce a_1 close to $1/2$, whereas $b_1 = \lambda^{1/2}$; in the following steps, the growth of b_k will be much faster than the decrease of a_k, until they become of the same order of magnitude. This is why $\mathrm{AGM}(1, \lambda)$ is not too small even for very small λ. For instance, we have $\mathrm{AGM}(1, 1e-15) \approx 4.3724e-2$. To illustrate better this nice behavior of the AGM iteration, we present the following.

Example 3.2. The so-called Hilbert matrix, $A = (a_{ij})$ with $a_{ij} = 1/(i + j - 1)$, is positive definite but very close to singularity even for moderate sizes n. Therefore, it

provides a good case to test the convergence of the AGM iteration. For $n = 11$ (the smallest eigenvalue of A is $3.3631e-015$) we get the results presented in Table 1. Starting with $\|(I + A^{1/2})^{-2}\|_2 \approx 1$, there is only a very moderate increase, with k, of the values $\|(A_{k-1}^{1/2} + B_{k-1}^{1/2})^{-2}\|_2$ in the third column of the table. As a result of this, the error $\|A_k - B_k\|_2$ converges to zero in a clear quadratic regimen. We finish the example with the observation that the equality holds in (3.1), in the 2-norm, for a normal matrix A with eigenvalues on the open right half plane.

TABLE 1
The convergence of iteration (1.1) *for* `A = hilb(11)`.

k	$\|A_k - B_k\|_2$	$\left\|\left(A_{k-1}^{1/2} + B_{k-1}^{1/2}\right)^{-2}\right\|_2$
1	5.0000e-001	1.0000e+000
2	2.4983e-001	1.9986e+000
3	1.1856e-001	3.7990e+000
4	3.7231e-002	5.2976e+000
5	3.8268e-003	5.5215e+000
6	4.0447e-005	5.5239e+000
7	4.5185e-009	5.5239e+000
8	1.9627e-015	5.5239e+000

The use of $\|A_k - B_k\|$ as a measure for the convergence is well justified by the fact that, for A_k and B_k close enough, i.e., for $\left\|I - A_k B_k^{-1}\right\| < 1/2$, it is

$$\|\mathrm{AGM}(A) - A_k\| \leq \frac{1}{2}\|A_k - B_k\|\left(1 + \frac{1}{8} + \cdots + \frac{1}{8^{2^k-1}} + \cdots\right),$$

which gives $\|\mathrm{AGM}(A) - A_k\| \leq \|A_k - B_k\|$ (this is Corollary of Lemma 5 in [26]) and is easily improved to

$$\|\mathrm{AGM}(A) - A_k\| \leq \frac{1}{2}\|A_k - B_k\|\left(1 + \frac{1/8}{1 - 1/8^2}\right) \approx 0.5635\,\|A_k - B_k\|.$$

4. Stability analysis of matrix AGM iteration. It is well known that the stability of a matrix iteration is crucial for its success [20, sect. 4.9], [19]. In finite precision arithmetic a loss of commutativity and rounding errors amplification are frequent so that unstable matrix iterations can fail to converge. An important tool to assess the stability of a matrix iteration is the Fréchet derivative. Given a map $f : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$, the Fréchet derivative of f at $X \in \mathbb{R}^{m\times n}$ in the direction of $E \in \mathbb{R}^{m\times n}$ is a linear operator $L_f(X)$ that maps the "direction matrix" E to $L_f(X, E)$ such that

$$\lim_{E\to 0} \frac{\|f(X + E) - f(X) - L_f(X, E)\|}{\|E\|} = 0.$$

The Fréchet derivative of the matrix function f may not exist at X, but if it does it is unique and coincides with the directional (or Gâteaux) derivative of f at X in the direction E. Hence the existence of the Fréchet derivative guarantees that for any $E \in \mathbb{R}^{m\times n}$,

$$L_f(X, E) = \lim_{t\to 0} \frac{f(X + tE) - f(X)}{t}.$$

Any consistent matrix norm $\|.\|$ on $\mathbb{R}^{m\times n}$ induces the operator norm $\|L_f(X)\| := \max_{\|E\|=1} \|L_f(X, E)\|$. Note that X and E above are not necessarily square. Here

one uses the same notation to denote both the matrix norm and the induced operator norm. For more information on the Fréchet derivative and its properties see, for instance, [4, Chap. X] and [20, Chap. 3].

Now we recall a necessary and sufficient condition for a matrix iteration to be stable (this should not be confused with the more usual "numerical stability"). See [20, p. 97] and also Problem 4.6 and its solution on p. 357 of [20].

LEMMA 4.1. *Consider the matrix iteration $X_{k+1} = f(X_k)$ with a fixed point M and assume that f is Fréchet differentiable at M. Then the iteration is stable in a neighborhood of M if and only if the Fréchet derivative $L_f(M)$ has bounded powers, i.e., if and only if $\rho(L_f(M)) \leq 1$, where $\rho(.)$ denotes the spectral radius, and any eigenvalue λ of $L_f(M)$ such that $|\lambda| = 1$ is semisimple, that is, λ appears only in Jordan blocks of size 1×1.*

To investigate the stability of (1.1), consider the iteration function

$$f(X,Y) = \begin{bmatrix} \frac{X+Y}{2} \\ \\ (XY)^{1/2} \end{bmatrix},$$

where $X,\ Y \in \mathbb{R}^{n\times n}$ are square matrices such that XY has no eigenvalues on the closed negative real axis. Some calculation gives the following expression for the Fréchet derivative of f at (X,Y) in the direction of (E,F):

$$L_f(X,Y;E,F) = \begin{bmatrix} \frac{E+F}{2} \\ \\ L \end{bmatrix},$$

where L is the unique solution of the Sylvester equation $(XY)^{1/2}L + L(XY)^{1/2} = EY + XF$. If $Z \in \mathbb{R}^{n\times n}$ has eigenvalues with positive real parts, any pair of the form (Z,Z) is a fixed point of f. However, since both sequences (A_k) and (B_k) converge to $\mathrm{AGM}(A)$, the fixed point that matters in our analysis is $(\mathrm{AGM}(A), \mathrm{AGM}(A))$ for some matrix $A \in \mathbb{R}^{n\times n}$ with eigenvalues on the open right half plane. At the fixed point (M,M) (we recall that $M = \mathrm{AGM}(A)$) the Fréchet derivative simplifies to

$$L_f(M,M;E,F) = \begin{bmatrix} \frac{E+F}{2} \\ L \end{bmatrix},$$

where L is the unique solution of the Sylvester equation

$$ML + LM = EM + MF. \tag{4.1}$$

According to Lemma 4.1, one needs to find the eigenvalues of the linear operator $L_f(M,M) : \mathbb{R}^{2n\times n} \longrightarrow \mathbb{R}^{2n\times n}$. We fix a basis in $\mathbb{R}^{2n\times n}$ and then find the corresponding matrix $\mathcal{M}$ which has size $2n^2 \times 2n^2$. To simplify, one uses the canonical basis. As we shall see below in Lemma 4.2, the spectral radius $\rho(\mathcal{M})$ always satisfies $\rho(\mathcal{M}) \geq 1$. If $\rho(\mathcal{M}) > 1$, then $L_f(M,M)$ is not power bounded, but if $\rho(\mathcal{M}) = 1$, then $L_f(M,M)$ is power bounded if and only if any eigenvalue $\lambda \in \sigma(\mathcal{M})$ with $|\lambda| = 1$ is semisimple (see Lemma 4.1). Until the end of this section, we shall determine the matrix $\mathcal{M}$ and investigate its eigenstructure.

Let λ be an eigenvalue of $L_f(M,M)$. Then there exists a pair of not simultaneously zero matrices (E,F) such that

$$L_f(M,M;E,F) = \lambda \begin{bmatrix} E \\ F \end{bmatrix}. \tag{4.2}$$

Equation (4.2) can be written as $\left[\begin{smallmatrix} \frac{E+F}{2} \\ L \end{smallmatrix}\right] = \lambda \left[\begin{smallmatrix} E \\ F \end{smallmatrix}\right]$, which in turn is equivalent to

$$\begin{bmatrix} \operatorname{vec}\left(\frac{E+F}{2}\right) \\ \operatorname{vec}(L) \end{bmatrix} = \lambda \begin{bmatrix} \operatorname{vec}(E) \\ \operatorname{vec}(F) \end{bmatrix}, \tag{4.3}$$

where vec(.) is the linear operator that stacks the columns of a matrix into a long vector. For basic details on the vec(.) operator and Kronecker products see [16] and [20, App. B.13]. Applying the vec(.) operator to (4.1), we find

$$\operatorname{vec}(L) = N^{-1}\left(M^\top \otimes I_n\right)\operatorname{vec}(E) + N^{-1}\left(I_n \otimes M\right)\operatorname{vec}(F),$$

where $N := M^\top \otimes I_n + I_n \otimes M$, with $\otimes$ standing for the Kronecker product and I_n for the identity matrix of order n. Replacing $\operatorname{vec}(L)$ in (4.3), a little calculation shows that (4.3) is equivalent to

$$\begin{bmatrix} \frac{1}{2}I_{n^2} & \frac{1}{2}I_{n^2} \\ N^{-1}\left(M^\top \otimes I_n\right) & N^{-1}\left(I_n \otimes M\right) \end{bmatrix} \begin{bmatrix} \operatorname{vec}(E) \\ \operatorname{vec}(F) \end{bmatrix} = \lambda \begin{bmatrix} \operatorname{vec}(E) \\ \operatorname{vec}(F) \end{bmatrix}. \tag{4.4}$$

This shows that the matrix of the linear operator $L_f(M, M)$ with respect to the canonical basis is the $2n^2 \times 2n^2$ matrix

$$\mathcal{M} := \begin{bmatrix} \frac{1}{2}I_{n^2} & \frac{1}{2}I_{n^2} \\ N^{-1}\left(M^\top \otimes I_n\right) & N^{-1}\left(I_n \otimes M\right) \end{bmatrix}. \tag{4.5}$$

Let $N_1 := N^{-1}\left(M^\top \otimes I_n\right)$ and $N_2 := N^{-1}\left(I_n \otimes M\right)$. An interesting feature of the matrices N_1 and N_2 is that

$$N_1 + N_2 = I_{n^2}. \tag{4.6}$$

From now on, we will use I without the subscript to denote the identity matrix. The size is clear from the context.

LEMMA 4.2. *With the notation above, the characteristic polynomial of the matrix* $\mathcal{M}$ *defined in* (4.5) *is given by* $p(\lambda) = (1-\lambda)^{n^2} \det\left(\left(N_2 - \frac{1}{2}I\right) - \lambda I\right)$.

Proof. Using standard properties of the determinant and formulae for the determinant of (2×2)-block matrices (see, for instance, [27, sect. 6.2]) and (4.6),

$$\begin{aligned} p(\lambda) &= \det(\mathcal{M} - \lambda I) \\ &= \det \begin{bmatrix} \left(\frac{1}{2} - \lambda\right) I & \frac{1}{2}I \\ N_1 & N_2 - \lambda I \end{bmatrix} \\ &= \det \begin{bmatrix} (1-\lambda)I & \frac{1}{2}I \\ (1-\lambda)I & N_2 - \lambda I \end{bmatrix} \\ &= (1-\lambda)^{n^2} \det\left(\left(N_2 - \frac{1}{2}I\right) - \lambda I\right). \end{aligned}$$

∎

Because $\lambda = 1$ is a semisimple eigenvalue of $\mathcal{M}$ (it admits n^2 linear independent eigenvectors, as many as its algebraic multiplicity), Lemma 4.2 shows that a sufficient condition for our operator to be power bounded is the spectral radius of $N_2 - \frac{1}{2}I$ to be smaller than unity. We have the following.

THEOREM 4.3. *For a given $n \times n$ matrix A with eigenvalues on the open right half plane, if $\lambda_i, i = 1 \cdots n$, are the eigenvalues of* AGM(A)*, then the operator defined in* (4.2) *is power bounded if and only if, for every pair $(i, j), i, j = 1, \ldots, n$,*

$$\left| \frac{\lambda_i}{\lambda_i + \lambda_j} - \frac{1}{2} \right| \leq 1, \tag{4.7}$$

where λ_i and λ_j are semisimple eigenvalues when the equality in (4.7) *holds.*

Proof. Assume as before that $M = \text{AGM}(A)$, $N = M^T \otimes I_n + I_n \otimes M$, and $N_2 = N^{-1}(I_n \otimes M)$. Because N is the Kronecker sum $M \oplus M^T$, its eigenvalues are $\lambda_i + \lambda_j$, $i, j = 1, \ldots, n$. In fact (see Theorem 13.16 in [25]) if (x_i, λ_i) and (z_j, λ_j) are eigenpairs of M and M^T, respectively, then we have $\left[(I_n \otimes M) + \left(M^T \otimes I_n\right)\right](z_j \otimes x_i) = (\lambda_i + \lambda_j)(z_j \otimes x_i)$.

Now, from the block diagonal structure of $I_n \otimes M$, with blocks equal to M, it follows that $(I_n \otimes M)(z_j \otimes x_i) = \lambda_i(z_j \otimes x_i)$ and, according to Theorem 4.3.17 in [23], a Jordan block of size larger than one is associated with an eigenvalue λ_i of $I_n \otimes M$ if and only if the same is true for λ_i as an eigenvalue of M. Finally, we have

$$\left[(I_n \otimes M) + \left(M^T \otimes I_n\right)\right]^{-1} (I_n \otimes M)(z_j \otimes x_i) = \frac{\lambda_i}{\lambda_i + \lambda_j}(z_j \otimes x_i). \tag{4.8}$$

Therefore, the matrix $\mathcal{M}$, as given in (4.5), has, in addition to the semisimple eigenvalue $\lambda = 1$ (of multiplicity n^2), the n^2 eigenvalues $\frac{\lambda_i}{\lambda_i + \lambda_j} - \frac{1}{2}, i, j = 1, \ldots, n$. Because of what we said about the eigenvectors $(z_j \otimes x_i)$, we conclude that $\frac{\lambda_i}{\lambda_i + \lambda_j} - \frac{1}{2}$ is semisimple if and only if λ_i and λ_j are both semisimple. ∎

There are many possible values for λ_p and λ_q $(p, q = 1, \ldots, n)$ such that

$$\left| \frac{\lambda_p}{\lambda_p + \lambda_q} - \frac{1}{2} \right| = 1. \tag{4.9}$$

With $\lambda_p = a + bi$ and $\lambda_q = c + di$, $(a, b, c, d) \in \mathbb{R}^4$, (4.9) gives place to $3(a^2 + b^2 + c^2 + d^2) + 10(ac + bd) = 0$, and this second-degree polynomial in the four variables has, of course, many zeros in $\mathbb{R}^4$ with a and c positive. For instance, if we fix $a = c = \frac{1}{2}$ and $b = 1$, we get $d = -1$ and $d = -\frac{7}{3}$, so that $\lambda_p = \frac{1}{2} + i$ satisfies (4.9) both with $\lambda_q = \frac{1}{2} - i$ and $\lambda_q = \frac{1}{2} - \frac{7}{3}i$.

Theorem 4.3 gives a necessary and sufficient condition to find if, for a given matrix A with eigenvalues with positive real parts, the AGM iteration is stable or not. Instead of using the spectrum of the Fréchet derivative operator as in Lemma 4.1, Theorem 4.3 enables one to verify the stability of (1.1), in an easier way, by means of the spectrum of AGM(A). Note that this spectrum can be determined by the scalar AGM iteration.

Remark 4.4. In the particular case of A having real positive eigenvalues only, the same is true for $M = \text{AGM}(A)$ and it is easy to conclude that the inequality in (4.7) holds. This means that in this particular case (1.1) is stable and it explains the good results obtained in Example 3.2.

The following example gives a matrix A for which the condition (4.7) is not satisfied, to illustrate that iteration (1.1) may be not stable.

Example 4.5. Let

$$A = 10 * \begin{bmatrix} 48 & -35 & 357 & 137 \\ -157 & 122 & 222 & 169 \\ -185 & -10 & -128 & -5 \\ -57 & -45 & 279 & 212 \end{bmatrix}$$

and $M = \mathrm{AGM}(A)$. The eigenvalues of A are nonreal and are located on the open right half plane. Some of them do not satisfy (4.7), which means that (1.1) is not stable. All the computations were performed in MATLAB, with unit roundoff $u \approx 1.1 \times 10^{-16}$. Till $k = 6$, the values $\|C_k\| = \|A_k - B_k\|$ approach zero (it is $\|C_6\| \approx 10^{-11}$) but then the errors become dominant and make $\|C_k\|$ to increase.

5. Stable iterations for the matrix AGM. To avoid the possible instability of the classical AGM formulation, we provide two stable variants of (1.1). The first modification proposed for (1.1) is

$$M_{k+1} = \frac{M_k + N_k}{2}, \qquad M_0 = I,$$

$$N_{k+1} = \frac{(M_k\, N_k)^{1/2} + (N_k\, M_k)^{1/2}}{2}, \quad N_0 = A, \tag{5.1}$$

and the second one is

$$P_{k+1} = 2P_k^{1/2}\,(I + P_k)^{-1}\,,\ P_0 = A, \tag{5.2}$$

$$Q_{k+1} = \frac{Q_k}{2}\,(I + P_{k+1})\,, \quad Q_0 = \frac{I + A}{2}. \tag{5.3}$$

In exact arithmetic, the recursive formulae (1.1) and (5.1) are equivalent, that is, for any k, $A_k = M_k$ and $B_k = N_k$. However, in finite precision arithmetic they produce quite different results (we shall see that the symmetry of the expression defining N_k changes the stability behavior, as (M_k) and (N_k) always converge both to $\mathrm{AGM}(A)$). The iteration (5.2)–(5.3) was obtained from (1.1) by setting $P_k := A_k^{-1}B_k$ and its connection with the scalar Legendre form in (2.2)–(2.3) is clear. Since $Q_0 = A_1$, a simple induction in (5.3) allows us to conclude that $Q_k = A_{k+1}$ for $k \geq 0$. Therefore, if convergence occurs (and this will always be the case, as we prove later), (Q_k) tends to $\mathrm{AGM}(A)$ as (P_k) converges to the identity matrix. In an alternative to (5.2)–(5.3) we may write, with $R_k := B_k^{-1}A_k$ (i.e., $R_k = P_k^{-1}$),

$$R_{k+1} = \frac{1}{2}\left(R_k^{1/2} + R_k^{-1/2}\right),\ R_0 = A, \tag{5.4}$$

$$Q_{k+1} = \frac{Q_k}{2}\left(I + R_{k+1}^{-1}\right), \quad Q_0 = \frac{I + A}{2}, \tag{5.5}$$

but this formulation should be avoided whenever $A^{1/2}$ is ill-conditioned for inversion.[1] Because, as observed before, in exact arithmetic, it is $A_{k+1} = M_{k+1} = Q_k$, the theoretical convergence properties are the same for all our AGM iterations (note that Q_k in (5.5) is the same as in (5.3)). It is therefore enough to give a full proof of the convergence behavior of one of those iterations, disregarding the stability issue for the moment.

THEOREM 5.1. *Let A have eigenvalues on the open right half plane. The sequence (R_k) defined in* (5.4) *always converges to I and*

$$\lim_{k\to\infty} \frac{\|R_{k+1} - I\|}{\|R_k - I\|^2} \leq \frac{1}{8}\|I\|^2, \tag{5.6}$$

i.e., convergence is ultimately quadratic.

[1] For instance, for `A = hilb(11)`, the difference between matrices Q_1 obtained with (5.5) and (5.3) has norm $\mathcal{O}\left(10^{-11}\right)$.

Proof. To prove that R_k always converge to I, we start by showing that its eigenvalues converge to unity. From (5.4), we have for each eigenvalue of R_{k+1}

$$\lambda(R_{k+1}) = \frac{1}{2}\left(\lambda(R_k)^{1/2} + \lambda(R_k)^{-1/2}\right) = \frac{1+\lambda(R_k)}{2\sqrt{\lambda(R_k)}}. \tag{5.7}$$

The argument of each one of the complex eigenvalues tends to zero because

$$|\theta_{k+1}| < \frac{1}{2}|\theta_k|, \tag{5.8}$$

where θ_{k+1} and θ_k denote the arguments of $\lambda(R_{k+1})$ and $\lambda(R_k)$, respectively. Let us prove (5.8). From (5.7), we may write $\theta_{k+1} = \overline{\theta}_k - \theta_k/2$, where $\overline{\theta}_k$ is the argument of $1+\lambda(R_k)$. For $\theta_k \in [0, \pi[$, it is $0 \le \overline{\theta}_k < \theta_k$ and then $-\theta_k/2 < \theta_{k+1} < \theta_k/2$. Similarly, if $\theta_k \in]-\pi, 0]$, it is $\theta_k < \overline{\theta}_k \le 0$ and $\theta_k/2 < \theta_{k+1} < -\theta_k/2$. Since the argument of the eigenvalues of $R_0 = A$ is assumed to be in $]-\pi/2, \pi/2[$, from (5.8) we may write

$$|\theta_k| < \frac{\pi}{2}\left(\frac{1}{2}\right)^k. \tag{5.9}$$

To show that the modulus ρ_k of each eigenvalue $\lambda(R_k)$ tends to unity, we start with the observation that $\rho_{k+1} = \frac{1}{2}(\rho_k^{1/2} + \rho_k^{-1/2}) \ge 1$ because the arithmetic mean of a positive number x and its inverse is not smaller than unity (and attains this minimum at $x = 1$). Now, the iteration function $\Phi(\rho) := \frac{1}{2}\left(\rho^{1/2} + \rho^{-1/2}\right)$ has the unique fixed point $\rho = 1$, because the equation $\rho = \Phi(\rho)$ has the unique root $\rho = 1$. From the theory of fixed point iterations (see, for instance, Theorem 2.4 in [10]), we know that $|\rho_{k+1} - 1| \le |\Phi'(\gamma)|\,|\rho_k - 1|$, where γ is a point between ρ_k and 1. Therefore, it is $\gamma > 1$ and

$$|\Phi'(\gamma)| = \frac{1}{4}\gamma^{-1/2}\left|1 - \gamma^{-1}\right| \le \frac{1}{4}\frac{1}{\sqrt{3}}\left(1 - \frac{1}{3}\right) \approx 0.0962$$

(note that $\Phi''(\rho) = 0$ for $\rho = 3$). This shows that fast convergence of ρ_k always occurs (and convergence gets asymptotically faster because $\Phi'(\gamma)$ tends to zero as ρ_k approaches the limit 1). At this point, we may conclude that the fixed point of (5.4) is similar to an upper triangular matrix T with units in the main diagonal. It follows that $T = \frac{1}{2}\left(T^{1/2} + T^{-1/2}\right)$ and, since $T^{1/2}$ is also upper triangular with units in the main diagonal, the entries in the diagonal above the main one, $\mathrm{diag}(T^{-1/2}, 1)$, of its inverse, are the reciprocals of the corresponding entries in $T^{1/2}$. So, $\mathrm{diag}(T, 1)$ has null entries. Because $T^{1/2}$ inherits such structure, simple calculations with Gauss–Jordan elimination show that the entries in $\mathrm{diag}(T^{-1/2}, 2)$ and $\mathrm{diag}(T^{-1/2}, 3)$ are the reciprocals of the corresponding ones in $T^{1/2}$. Repeating the reasoning for the remaining diagonals, one concludes that $T = I$. Finally, we prove the quadratic convergence by writing $R_{k+1} - I = \frac{1}{4}(R_k - I)^2 R_k^{-1}(R_{k+1} + I)^{-1}$ and

$$\frac{\|R_{k+1} - I\|}{\|R_k - I\|^2} \le \frac{1}{4}\|R_k^{-1}\| \cdot \|(R_{k+1} + I)^{-1}\|,$$

which implies (5.6). ∎

The convergence of (Q_k) follows from (5.5) and from the convergence of (R_k) to I. This implies that (A_k) and (M_k) also converge, in exact arithmetic, and their convergence is ultimately quadratic.

Now, we address the stability of our alternative AGM computations. It is important to stress that, unlike (A_k, B_k) and (M_k, N_k), the iterations P_k and Q_k are decoupled (the same is true for R_k and Q_k), i.e., one may iterate with (5.2) alone until P_s, for some s, is sufficiently close to the identity, and then compute Q_s with s matrix multiplications in (5.3), but, of course, it is preferable to carry out the update of Q_k, for $k \leq s$, immediately after computing P_k. Therefore, for the sake of the stability of the iterations, (5.3) and (5.5) may be left out.

As already pointed out, the stability of the iterations we are tackling here is not the numerical stability in the usual sense. Later, we will look at the numerical errors in the finite process of computing Q_s.

THEOREM 5.2. *Let A have eigenvalues on the open right half plane. Then the iterative formulae* (5.1)*,* (5.2)*, and* (5.4) *are stable.*

Proof. Let

$$g(X,Y) = \begin{bmatrix} \frac{X+Y}{2} \\ \\ \frac{(XY)^{1/2}+(YX)^{1/2}}{2} \end{bmatrix}$$

be the iteration function associated to (5.1). Given matrices X and Y with eigenvalues on the open right half plane, a little calculation shows that the Fréchet derivative of g at (X, Y) in the direction of (E, F) is given by

$$L_g(X,Y;E,F) = \begin{bmatrix} \frac{E+F}{2} \\ \frac{L_1+L_2}{2} \end{bmatrix},$$

where L_1 is the unique solution of the Sylvester equation $(XY)^{1/2}L_1 + L_1(XY)^{1/2} = EY + XF$ and L_2 is the unique solution of $(YX)^{1/2}L_2 + L_1(YX)^{1/2} = FX + YE$. At the fixed point (M, M), with $M = \mathrm{AGM}(A)$, the Fréchet derivative is

$$L_g(M,M;E,F) = \begin{bmatrix} \frac{E+F}{2} \\ \frac{L_1+L_2}{2} \end{bmatrix},$$

where L_1 and L_2 are, respectively, the solutions of the following Sylvester equations

$$ML_1 + L_1M = EM + MF, \tag{5.10}$$

$$ML_2 + L_2M = FM + ME. \tag{5.11}$$

Consider now the Sylvester equation

$$ML + LM = (E+F)M + M(E+F), \tag{5.12}$$

which has a unique solution. It is obvious that $L = E + F$ is a solution of (5.12). It turns out, however, that, by (5.10) and (5.11), $L = L_1 + L_2$ is another solution of (5.12). Thus $E + F = L_1 + L_2$. Hence

$$\begin{aligned} L_g^2(M,M;E,F) &= L_g\left(M,M;L_g(M,M;E,F)\right) \\ &= L_g(M,M;E,F), \end{aligned}$$

that is, $L_g(M, M)$ is idempotent and so the iteration (5.1) is stable.

Let $h(X) = 2X^{1/2}(I+X)^{-1}$ be the iteration function associated to (5.2). Given X with eigenvalues on the open right half plane, the Fréchet derivative of h at X in

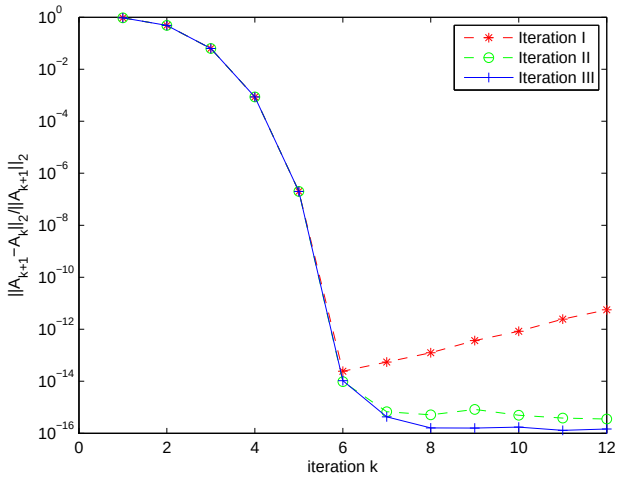

FIG. 2. *Convergence of* $\|A_{k+1}-A_k\|_2/\|A_{k+1}\|_2$ *to zero in iterations* I, II, *and* III *which refers, respectively, to* (1.1), (5.1), *and* (5.2)–(5.3).

the direction of E is $L_h(X,E) = 2L(I+X)^{-1} - 2X^{1/2}(I+X)^{-1}E(I+X)^{-1}$, where L is the unique solution of the Sylvester equation $X^{1/2}L + LX^{1/2} = E$. h has a fixed point at I, where the Fréchet derivative is zero, that is, $L_h(I,E) = 0$. Therefore (5.2) is stable.

Let $\Phi(X) = \left(X^{1/2} + X^{-1/2}\right)/2$ be the iteration function associated to (5.4). Given X with eigenvalues on the open right half plane, the Fréchet derivative of Φ at X in the direction of E is $L_\Phi(X,E) = \left(L - X^{-1/2}LX^{-1/2}\right)/2$, where L is the unique solution of the Sylvester equation $X^{1/2}L + LX^{1/2} = E$. Φ has a fixed point at I, where the Fréchet derivative is zero, that is, $L_h(I,E) = 0$. Therefore (5.4) is also stable. ∎

To illustrate the stability behavior of iterations (1.1), (5.1), and (5.2)–(5.3), we consider again the matrix A of Example 4.5. In Figure 2 we plot (in a logarithmic scale) the values $\|A_{k+1} - A_k\|_2/\|A_{k+1}\|_2$. For a fair comparison, here we have not used $\|A_k - B_k\|_2$ because no B_k is computed in (5.2)–(5.3). Since $\|A_{k+1} - A_k\| = \|A_k - B_k\|/2$ this is also a good measure for convergence. As already seen, iterations produced with (1.1) fail to converge. In contrast, the other two iterative formulae perform well.

6. Removing the restriction on the real part of the eigenvalues of A. Until now, and following Stickel [32], we have restricted the computation of AGM(A) to the case when the eigenvalues of A have positive real parts. The reason for this restriction is to avoid the possibility of A_kB_k, for some k, to have some real negative eigenvalues, in which case the principal square root of A_kB_k is not defined. If some eigenvalues of A have negative real parts, such problem may indeed occur when using (1.1) and (5.1) (in the latter case, for the eigenvalues of M_kN_k and N_kM_k). Even if the eigenvalues of such products do not fall down on $\mathbb{R}_0^-$, and thus their principal square roots do exist, important errors are expected if there are eigenvalues close to the closed negative real axis. This is because of the ill-conditioning of the square roots computation in this case [20, p. 134]. Fortunately, such problem

does not exist in our Legendre variant (5.2), as a result of (5.8). This also holds for the eigenvalues of P_k (it is the inverse of R_k). To illustrate this important difference in our AGM variants, in MATLAB, we define $\texttt{z} = (\texttt{1} + \texttt{sqrt(2)/2}) * (\texttt{-1} + \texttt{i})$; $\texttt{A} = \texttt{compan(poly([1 + i, 1 - i, z, conj(z)]))}$, that is, A is the (real) companion matrix of the polynomial with the given roots ($\bar{z}$ is the complex conjugate of z). For $k = 2$ in (1.1) we get $B_2 = (A_1 B_1)^{1/2}$, with $A_1 B_1 = (I + A)A^{1/2}/2$, and

$$z^{1/2}\,(1+z)\,/2 = \bar{z}^{1/2}\,(1+\bar{z})\,/2 = -\frac{1}{2}\left(\frac{\sqrt[4]{8}}{4} + \sqrt{3 + \frac{17}{8}\sqrt{2}}\right)$$

is in fact real negative. As a consequence, iterations (5.1) deliver complex M_2 and N_2 and fail to produce a real matrix for AGM(A). This is not the case of (5.2), which converges to a real matrix even when the eigenvalues of A have negative real parts (but not lying in the closed negative real axis). For this reason, it is more practical to consider the limit of (Q_k) in (5.3) as being AGM(A). This does not disagree with the standard definition of the AGM (see [13, sect. 2]) and produces a correct result for the logarithm of A in our algorithm to be presented in section 8.

7. Efficient computation of the Legendre matrix AGM. From now on, we are interested only in the variant expressed in (5.2)–(5.3) because, as we have seen in the previous two sections, it is the one with the best numerical properties. It is therefore the basis for our code for the AGM computation.

For the sake of efficiency, the computation of a function of a matrix, $f(A)$, for complex A, usually starts with the Schur reduction $Q^* A Q = T$, where Q is unitary (for stability reasons) and T is triangular (usually upper). For real A, complex arithmetic may be avoided by using real orthogonal transformations that produce a quasi-upper triangular T. In both cases, the computation of $f(T)$ is then carried out and finally one computes $f(A) = Q f(T) Q^*$. There is, however, also an interest in algorithms that work on the original matrix A without the use of a Schur decomposition. For instance, in [11, 1] transformation free algorithms are proposed for the computation of matrix logarithms. These algorithms require only matrix multiplications and the solution of multiple right-hand-side linear systems (MRHSLS). The authors have observed that they are potentially more efficient on a parallel computer and they may also be attractive for higher precision computation.

The AGM, being a function of matrices in the usual sense, may be computed with or without the initial Schur reduction and this initial choice influences the decision on which methods to use for computing the required square roots. An efficient algorithm for the square roots of a matrix in the Schur form is given in [14]. However, because we are aiming at an algorithm tailored for parallel implementation, we will concentrate on a reduction free algorithm.

The standard Newton method for computing the principal square root of a matrix P having no eigenvalues on the closed negative real axis may be defined by the iterative formula

$$X_{p+1} = \frac{1}{2}\left(X_p + X_p^{-1} P\right) \tag{7.1}$$

with $p = 0, 1, 2, \ldots$. Starting at $X_0 = P$, the sequence (X_k) converges quadratically to $P^{1/2}$.

It is well known that the Newton iteration (7.1) is unstable, unless the eigenvalues λ_i of P are very closely clustered, in the sense that (λ_i/λ_j) lies in a ball of radius 2

about $z = 1$ in the complex plane, for all i and j (see [20, p. 147]). Stable variants of the Newton method obtained from (7.1) by a suitable change of variables are studied in [20, pp. 139–148], being the product form of the Denman and Beavers iteration (DB-iteration) one of the most popular (see also [15, 11]). An effective implementation of the product form DB-iteration requires the multiplication of each iteration by a scaling parameter μ_p. A widely used version of the scaled product form DB-iteration is

$$\begin{aligned} \mu_p &= |\det(Y_p)|^{-1/(2n)}, \\ Y_{p+1} &= \frac{1}{2}\left(I + \frac{\mu_p^2 Y_p + \mu_p^{-2} Y_p^{-1}}{2}\right), \\ Z_{p+1} &= \frac{1}{2}\mu_p Z_p \left(I + \mu_p^{-2} Y_p^{-1}\right), \end{aligned} \tag{7.2}$$

with $p = 0, 1, 2, \ldots$. If P has no negative eigenvalues and $Y_0 = Z_0 = P$, it is known that $\lim_{p\to\infty} Y_k = I$ and $\lim_{p\to\infty} Z_k = P^{1/2}$.

The computation of $\det(Y_p)$, for the scaling factor μ_p in (7.2), does not involve a significant extra cost if the inverse Y_p^{-1} is computed from the LU decomposition of Y_p. However, underflow or overflow may occur in the computation of the determinant. Overflow causes the failure of function `sqrtm_dbp`,[2] for example, for the matrices $\mathtt{P} = \mathtt{gallery}(\mathtt{'invhess'}, \mathtt{n})$ with $n > 150$, since $\det(P)$ is equal to the product of the first n odd numbers. In section 8 we will need to compute square roots of matrices scaled by very small numbers and, in this context, underflow in the computation of $\det(Y_p)$ is likely to occur. Because of this, if $U = [u_{ij}]_{i,j=1,\ldots,n}$, we will compute $|\det(Y_p)|^{-1/(2n)} = |u_{11}|^{-1/(2n)} \cdots |u_{nn}|^{-1/(2n)}$.

For the computation of $P_k^{1/2}$ in (5.2) we use `sqrtm_dbp` with the modification just described and $Y_0 = Z_0 = P_k$. As soon as the norm of $\Delta_k := I - P_k$ becomes small enough (we discuss this later), one may switch to using

$$P_k^{1/2} = I - \frac{\Delta_k}{2} - \frac{\Delta_k^2}{8} - \frac{\Delta_k^3}{16} - \frac{5}{128}\Delta_k^4 - \frac{7}{256}\Delta_k^5 - \frac{21}{1024}\Delta_k^6 - \frac{33}{2048}\Delta_k^7 - \cdots \tag{7.3}$$

and

$$\begin{aligned} (I + P_k)^{-1} &= [2I - (I - P_k)]^{-1} \\ &= \frac{I}{2} + \frac{\Delta_k}{4} + \frac{\Delta_k^2}{8} + \frac{\Delta_k^3}{16} + \frac{\Delta_k^4}{32} + \frac{\Delta_k^5}{64} + \frac{\Delta_k^6}{128} + \frac{\Delta_k^7}{256} + \cdots. \end{aligned} \tag{7.4}$$

These series converge for $\|I - P_k\| < 1$ and $\|I - P_k\| < 2$, respectively. Performing the product of the power series expansions (7.3) and (7.4), we get

$$P_{k+1} = I - \Delta_{k+1} \tag{7.5}$$

with

$$\Delta_{k+1} = \frac{\Delta_k^2}{8} + \frac{\Delta_k^3}{8} + \frac{13}{128}\Delta_k^4 + \frac{5}{64}\Delta_k^5 + \frac{61}{1024}\Delta_k^6 + \frac{47}{1024}\Delta_k^7 + \cdots. \tag{7.6}$$

If $\|\Delta_k\| = \|I - P_k\| < 1$, the series (7.6) converges and its expression puts in evidence the quadratic convergence of Δ_k to the zero matrix and of P_k to the identity. The

[2] This is an implementation of (7.2) from the the Matrix Function Toolbox [21], which computes $\det(Y_p)$.

analytical expression for the coefficient d_m of Δ_k^m, for $m \geq 2$, can be derived by the Cauchy product formula and is given by $d_m = -\frac{1}{2^m} \sum_{\ell=0}^{m} \frac{2^\ell}{\ell!} \left(-\frac{1}{2}\right)_\ell$, where $(a)_\ell$ is the rising factorial, already mentioned in section 2. These coefficients are easily computed from $d_{m+1} = d_m/2 + r_{m+1}$, where $r_{m+1} = r_m \times (m - 1.5)/m$, and starting values $d_2 = 1/8$ and $r_2 = 1/8$. This also shows that $\lim_{m\to\infty} d_m = 0$.

Therefore, in the last iterations of (5.2)–(5.3), the computation of a square root and an inverse will be replaced by a few matrix products in the implementation of (7.5). There is no need to explicitly form P_{k+1} because from (5.3) we may write

$$Q_{k+1} = Q_k \left(I - \frac{\Delta_{k+1}}{2} \right) \tag{7.7}$$

and a convenient stopping criterium follows from

$$\frac{\|Q_{k+1} - Q_k\|}{\|Q_k\|} \leq \frac{\|\Delta_{k+1}\|}{2}. \tag{7.8}$$

A practical use of (7.7) requires the approximation of Δ_{k+1} in (7.6) by a suitable Taylor polynomial $\Delta_{k+1}^{(p)}$ of degree p, with $p \geq 2$. The choice of a convenient p depends on $\|\Delta_k\|$, of course. There is no point in computing Δ_{k+1} to high relative accuracy because such accuracy will be wasted in (7.7). In fact, in the computation of the diagonal entries of $I - \frac{\Delta_{k+1}}{2}$ errors of the size of the arithmetic precision will be produced and these diagonal entries dominate the computation. More precisely, with $\widetilde{Q}_{k+1} := Q_k(I - \Delta_{k+1}^{(p)}/2)$, we have

$$Q_{k+1} - \widetilde{Q}_{k+1} = Q_k \cdot \frac{\Delta_{k+1} - \Delta_{k+1}^{(p)}}{2} \tag{7.9}$$

and

$$\frac{\|Q_{k+1} - \widetilde{Q}_{k+1}\|}{\|Q_{k+1}\|} \leq \frac{\|Q_k\|}{\|Q_{k+1}\|} \cdot \frac{\|\Delta_{k+1} - \Delta_{k+1}^{(p)}\|}{2}, \tag{7.10}$$

which shows that absolute errors of the size of the arithmetic precision in the approximation $\Delta_{k+1} \approx \Delta_{k+1}^{(p)}$ provide good relative accuracy in $\widetilde{Q}_{k+1}$ (note that $\|Q_k\|/\|Q_{k+1}\|$ is close to unity).

Important savings in the number of matrix multiplications may be achieved if the polynomials in (7.6) are expressed in a convenient manner. For instance, the seventh order polynomial may be computed with just four matrix multiplications in the form

$$\frac{\Delta_k^2}{8} + \frac{13}{128}\Delta_k^4 + \frac{61}{1024}\Delta_k^6 + \Delta_k \left(\frac{\Delta_k^2}{8} + \frac{5}{64}\Delta_k^4 + \frac{47}{1024}\Delta_k^6 \right).$$

As compared to the Horner method, there is a saving of two matrix multiplications at the cost of storing Δ_k^2, Δ_k^4, and Δ_k^6. For larger values of p the savings are more significant [20, pp. 244–245]. A slight improvement on this technique for computing the matrix polynomial has been recently proposed in [31, sect. 3.2].

The first two rows of Table 2 give the highest degree p of the polynomial to approximate Δ_{k+1} in (7.6) that may be evaluated with π_p matrix multiplications, with $\pi_p = 1, \ldots, 8$ (see Table 10.3 in [20]).

TABLE 2
Order of polynomials and number of matrix multiplications for different values $\|\Delta_k\|$.

p	2	3	5	7	9	13	17	21
π_p	1	2	3	4	5	6	7	8
$\overline{\|\Delta_k\|}$	1.2e-5	2.0e-4	3.7e-3	1.6e-2	3.8e-2	0.10	0.17	0.24

Since the sequence of coefficients d_m is monotonic decreasing, we may write

$$\|\Delta_{k+1} - \Delta_{k+1}^{(p)}\| \le d_{p+1} \frac{\|\Delta_k\|^{p+1}}{1 - \|\Delta_k\|} \tag{7.11}$$

and from this we have computed, for each p, an upper bound $\overline{\|\Delta_k\|}$ that guarantees a truncation error not larger than $\varepsilon = 2^{-52}$. These bounds are given in the third row of Table 2.

To illustrate the efficient computation of AGM(A) by a combination of the iterations (5.2)–(5.3) with (7.7), we consider again the matrix $\texttt{A} = \texttt{gallery}(\texttt{'invhess'}, \texttt{50})$ and use the Frobenius norm. Starting at $P_0 = A$ and $\|\Delta_0\| = \|P_0 - I\| = 1.04e+03$, with (5.2), we get $P_k, k = 1, \ldots, 7$, with $\|I - P_k\|$ as given in the second row of Table 3. On the other hand, switching to the Taylor approximations after the third iteration with (5.2), we compute $\Delta_4^{(17)}, \Delta_5^{(5)}, \Delta_6^{(2)}$ with norms as given in the third row of Table 3.

TABLE 3
Sequence of (approximated) values $\|\Delta_k\|$ *obtained only with* (5.2) *and from* (5.2) *combined with* (7.6) *for the matrix* $\texttt{A} = \texttt{gallery}(\texttt{'invhess'}, \texttt{50})$.

k	1	2	3	4	5	6	7
$\|I - P_k\|$ (5.2)	4.95	1.55	0.11	7.2e-04	4.2e-08	1.35e-15	1.11e-16
$\|\Delta_k^{(p)}\|$ (7.6)				7.2e-04	4.2e-08	1.54e-16	

Thus, in our approach that uses (7.6), only the first three iterations involve the computation of matrix square roots; the remaining iterations are computed essentially by matrix products. It is interesting to observe that (7.6) computes $\Delta_6^{(2)}$ such that $\|\Delta_6^{(2)}\|_F < \varepsilon = 2^{-52}$ but for P_6, computed with (5.2), we do not have $\|I - P_6\|_F < 2^{-52}$ and an extra step is required. This is due to subtractive cancellation in $I - P_6$ which, of course, does not occur in our alternative formulation. Each step in (5.2) requires the computation of a square root and the solution of an MRHSLS.[3] The computation of the square root in each one of the last four steps in (5.2) requires 4, 3, 1, and 1 iterations with (7.2) and each one of these essentially involves the computation of one matrix inverse and the product of two square matrices. This gives a total of 4 MRHSLS, 9 matrix inversions, and 9 matrix products and this is what we avoid by using (7.6) with a much lower cost of 11 matrix products (since we have used polynomials of degree 17, 5, and 2).

The following observation is relevant for the algorithm to be presented in the next section: when $\|A\| \le 2^{-26}$, the computation of $P_1 = 2A^{1/2}(I + A)^{-1}$ in (5.2) may be replaced with $P_1 = 2A^{1/2}(I - A)$, since, for norms such that $\|I\| = 1$, we have

$$\| (I + A)^{-1} - (I - A) \| \le \|A\|^2 + \|A\|^3 + \cdots = \frac{\|A\|^2}{1 - \|A\|}.$$

This saves an MRHSLS at the cost of a matrix multiplication.

[3] In our MATLAB code, to compute $X = ZW^{-1}$ we used $X = Z/W$ (right matrix divide) which is given by $X = \left(W^T \backslash Z^T\right)^T$.

8. The AGM iteration and the matrix logarithm. Given a nonsingular matrix $A \in \mathbb{R}^{n\times n}$, any solution of the matrix equation $e^X = A$, where e^X denotes the exponential of the matrix X, is called a *logarithm* of A. In general, a nonsingular real matrix may have an infinite number of real and complex logarithms. However, if A has no eigenvalues on the closed negative real axis, then there exists a unique real logarithm of A whose eigenvalues lie on the open strip $\{z \in \mathbb{C} : -\pi < \operatorname{Im} z < \pi\}$ of the complex plane. This unique logarithm is called the *principal* logarithm of A and will be denoted by $\log(A)$. For background on matrix logarithms and general matrix functions see [20, 23].

With convenient restrictions on the spectrum of a matrix A and appropriated modifications, some of the results recalled in section 2 can be extended to matrices. For instance, if $\|A\| < 1$ we can set

$$K(A) := \int_0^{\pi/2} \left(I - A^2 \sin^2\theta\right)^{-1/2} d\theta \quad \text{and}$$
$$K'(A) := \int_0^{\pi/2} \left(I - (I - A^2) \sin^2\theta\right)^{-1/2} d\theta,$$

where the integrals are defined componentwise. The following identity also holds:

$$K'(A) = \frac{\pi}{2} \left[\mathrm{AGM}(A)\right]^{-1}. \tag{8.1}$$

Now we revisit two results connecting the matrix AGM with the matrix logarithm that were originally stated by Stickel in [32, sect. 3]. We provide different statements and proofs that in our view are more clear and concise. See also [26].

THEOREM 8.1. *Let A have no eigenvalues on the closed negative real axis and assume in addition that $\|A\| < 1$. Then*

$$\left\|\log(A) - \left(\log(4)I - \frac{\pi}{2}[\mathrm{AGM}(A)]^{-1}\right)\right\| \le \frac{\|A\|^2}{1-\|A\|^2}\left(3.4 + \|\log A\|\right). \tag{8.2}$$

Proof. Assume that A satisfies the conditions of the theorem and consider the extension of (2.6) to matrices:

$$K'(A) - \log\left(4A^{-1}\right) = \sum_{i=1}^{\infty} \left(\frac{\left(\frac{1}{2}\right)_i}{i!}\right)^2 \left(\log\left(4A^{-1}\right) - \sum_{j=0}^{i-1} \frac{2}{(2j+1)(2j+2)}\right) A^{2i}. \tag{8.3}$$

A little calculation shows that for any integer $i \ge 0$, $\left(\left(\frac{1}{2}\right)_i / i!\right)^2 \le 1$, and, for any integer $i \ge 1$, $\sum_{j=0}^{i-1} \frac{2}{(2j+1)(2j+2)} \le 2$. Applying norms in (8.3) yields

$$\begin{aligned}\left\|K'(A) - \log\left(4A^{-1}\right)\right\| &\le \left\|\log\left(4A^{-1}\right)\right\| \sum_{i=1}^{\infty} \|A\|^{2i} + 2\sum_{i=1}^{\infty} \|A\|^{2i} \\ &\le \left(\left\|\log\left(4A^{-1}\right)\right\| + 2\right) \frac{\|A\|^2}{1-\|A\|^2}.\end{aligned}$$

Attending to (8.1) and $\log\left(4A^{-1}\right) = (\log 4)I - \log A$, the result follows. ∎

Obviously, for (8.2) to be of practical value for the computation of $\log A$ using $\mathrm{AGM}(A)$, it needs to be applied to matrices with very small norm. A simple scaling,

which is a standard procedure for computing logarithms, makes the approximation in (8.2) of general use, independently of $\|A\|$.

COROLLARY 8.2. *Let A have no negative real eigenvalues and let ϵ be a sufficiently small positive number such that $\|A\| < 1/\epsilon$. Then*

(8.4)
$$\left\|\log(A) - \left(\log(4/\epsilon)I - \frac{\pi}{2}[\mathrm{AGM}(\epsilon A)]^{-1}\right)\right\| \leq \frac{\epsilon^2\|A\|^2}{1-\epsilon^2\|A\|^2}\left(3.4 + |\log\epsilon| + \|\log A\|\right),$$

Proof. Replace A by ϵA in (8.2) and use properties of the logarithm. ∎

By taking ϵ sufficiently small we may get the error in our approximation for $\log A$ as small as required. For the relative error of the approximation of $\log(A)$ obtained from $\mathrm{AGM}(\epsilon A)$ we have

$$\frac{\left\|\log(A) - \left(\log(4/\epsilon)I - \frac{\pi}{2}[\mathrm{AGM}(\epsilon A)]^{-1}\right)\right\|}{\|\log(A)\|} \leq \frac{\epsilon^2\|A\|^2}{1-\epsilon^2\|A\|^2}\left(1 + \frac{3.4 + |\log\epsilon|}{\|\log(A)\|}\right), \tag{8.5}$$

which, for practical purposes, may be simply taken as $\frac{\epsilon^2\|A\|^2}{1-\epsilon^2\|A\|^2}$. Of course, the bound in the right side of (8.5) grows without limit as A approaches the identity matrix (whose logarithm is the null matrix) but this is an intrinsic problem of the logarithm, not an issue raised by our approximation (8.4).

What makes our method a serious candidate for computing $\log(A)$ is that a very small ϵ has only a moderate impact in the number of iterations required to compute $\mathrm{AGM}(\epsilon A)$. For $\mathtt{A} = \mathtt{gallery('invhess',50)}$, the convergence of the iterations (5.2)–(7.7) for $\mathrm{AGM}(\epsilon A)$, with $\epsilon = 1$, $\epsilon = 10^{-8}$, $\epsilon = 2^{-26}/\|A\|_F$, and $\epsilon = 10^{-16}$, may be observed in Table 4 (in brackets we list those values $\|\Delta_k\|_F = \|I - P_k\|_F$ which correspond to iterations carried out with (5.2); the rest of them do correspond to iterations with (7.7)).

TABLE 4
The convergence to zero of $\|I - P_k\|_F$ in the computation of $\mathrm{AGM}(\epsilon A)$.

k	$\epsilon = 1$	$\epsilon = 10^{-8}$	$\epsilon = 2^{-26}/\|A\|_F$	$\epsilon = 10^{-16}$
1	(4.9530)	(7.0622)	(7.0707)	(7.0711)
2	(1.5489)	(6.5742)	(6.9738)	(7.0661)
3	(1.1104e-01)	(3.6160)	(5.4475)	(6.6965)
4	7.2434e-04	(4.6234e-01)	(1.5898)	(3.9907)
5	4.2029e-08	(5.1101e-03)	(6.1127e-02)	(5.7805e-01)
6	1.5496e-16	9.8080e-07	8.5775e-05	(6.7611e-03)
7		7.2727e-14	2.9873e-10	9.8948e-07
8		5.0857e-28	7.1832e-21	3.5730e-14
9				9.7042e-29

For the computation of $\log(A)$, the adequate choice is

$$\epsilon = 2^{-26}/\|A\| \tag{8.6}$$

since such ϵ makes the bound in (8.5) close to 2^{-52} provided that $\|\log(A)\|$ is not small. In this framework, we are to compute $\mathrm{AGM}(P_0)$, with $\|P_0\| = 2^{-26}$, and the inverse $(I + P_0)^{-1}$ will be well approximated by the polynomial $I - P_0$, as explained at the end of the previous section. In the next step, $\|P_1\|$ is typically $\mathcal{O}\left(10^{-4}\right)$ and, in the computation of P_2, in (5.2), $(I + P_1)^{-1}$ may be replaced with $I - P_1 + P_1^2 - P_1^3 + P_1^4$. But

this corresponds to trading an MRHSLS for four matrix multiplications (depending upon the computing platform it may still pay off).

The complete algorithm is as follows.

Algorithm 8.3. Given A with no real negative eigenvalues (matrix square roots are computed with `sqrtm_dbp` modified to avoid underflow):

$\epsilon = 2^{-26}/\|A\|_F$
$P_0 = \epsilon \cdot A$
$P_1 = 2P_0^{1/2}(I - P_0)$
$Q_1 = (I + P_0)(I + P_1)/4$
$k = 1$
while $\|I - P_k\|_F > 0.24$
  $P_{k+1} = 2P_k^{1/2}(I + P_k)^{-1}$ (solve MRHSLS)
  $Q_{k+1} = Q_k(I + P_{k+1})/2$
  $k = k + 1$
end while
$\Delta_k = I - P_k$
while $\|\Delta_k\|_F > 2^{-52}$
  Use values in Table 2 to find degree of polynomial in (7.6) and compute Δ_{k+1}
  $Q_{k+1} = Q_k \cdot (I - \Delta_{k+1}/2)$
  $k = k + 1$
end while
$\log(A) = \log(4/\epsilon)I - \frac{\pi}{2}Q_k^{-1}$

8.1. `logm_agm` versus the state-of-the-art methods. The best existing codes for computing the logarithm of a matrix are based upon inverse scaling and squaring (`iss`). In simple terms, with or without an initial reduction to triangular form, a number of square roots are carried out until $A^{1/2^s}$ is sufficiently close to identity so that a Padé approximant will produce an accurate approximation for $\log\left(A^{1/2^s}\right) = \log\left(I + X_s\right)$, where $X_s = A^{1/2^s} - I$. Because of the linear convergence of $A^{1/2^s}$, the success of the `iss` approach is a careful balance between the number of initial square roots and the degree of the Padé approximant. The state-of-the-art `iss` codes are those presented in [1] and [2], which improve significantly in speed and accuracy on those of Higham [20, Algor. 11.9, 11.10], which in turn are refinements of those of Cheng et al. [11] and Kenney and Laub [24]. Algorithms 4.1 and 5.2 from [1] are implemented in MATLAB codes `logm_new` and `logm_iss_full` [22], respectively. The first one reduces the given matrix to complex triangular form, whereas the second one does not include this initial reduction. The algorithms presented in [2] are implemented in MATLAB codes available from [30]. For real matrices, the MATLAB function `logm_frechet_real` reduces the matrix to Schur form, avoiding complex arithmetic (in [2] this is shown to be twice as fast as the complex arithmetic code). Furthermore, with the codes in [30] it is also possible to compute the Fréchet derivative of $\log(A)$ in any direction E and an estimate of the condition number of the logarithm, to give some idea of the accuracy. In the next section we will deal with some numerical issues and will compare the accuracy of $\log(A)$, for real A, produced with `logm_frechet_real`, `logm_iss_full` and our own code, `logm_agm`, that computes the logarithm of A from $\mathrm{AGM}(\epsilon A)$.

We devote the rest of this section to computational efficiency and, on this respect, we restrict the comparison of our code with `logm_iss_full` because, for the reasons expounded at the beginning of section 7, we are interested in reduction free algorithms.

For a matrix of order n, the number of flops involved in `logm_iss_full` is essentially given by [1, p. 162]

$$\left(\sum_{j=1}^{s} it_j\right) 4n^3 + 8mn^3/3 + 2(s-2/3)n^3, \tag{8.7}$$

where s is the number of square roots, it_j is the number of iterations (7.2) in the jth square root, and m is the degree of the Padé approximant. Each iteration (7.2) involves one matrix multiplication and one matrix inversion, each one of these operations requiring $2n^3$ flops [20, p. 336]. The m degree Padé approximant needs the solution of m MRHSLS and each one of these takes $8n^3/3$ flops. Finally, the last term in (8.7) accounts for the implementation of Algorithm 5.1 in [1] that avoids the direct computation of $X_s = A^{1/2^s} - I$, where subtractive cancellation in the diagonal entries of $A^{1/2^s}$ (which tends to I) may be severe. This costs $s-2$ matrix multiplications and one MRHSLS.

We ran `logm_iss_full` and `logm_agm` for matrices `A = expm(L)` (this guarantees that A has no real negative eigenvalues), where `L = rand(n)`, for different values of n. For instance, for $n = 20$, `logm_iss_full` required five square roots with (by this order) 11, 8, 6, 5, and 5 iterations (7.2) and $m = 9$. On its turn, `logm_agm` used 5 steps (5.2)–(5.3) and the number of iterations (7.2) was 11, 8, 6, 5, and 4; on top of this, for the total cost of those 5 steps, there are 4 MRHSLS and 6 matrix multiplications (in the first step we trade an MRHSLS for a matrix multiplication). Then, for the updates (7.7), `logm_agm` used polynomials of degrees 21, 5, and 2, which require, according to Table 2, a total of 12 matrix multiplications. Finally, there is the inverse computation $[\mathrm{AGM}(\epsilon A)]^{-1}$. In Table 5 we give the number of matrix multiplications (MM), inverses (Inv), and MRHSLS required by the two codes for some values of n (the results presented may vary slightly from run to run since we are using random matrices).

TABLE 5

Major operations required by the two codes: `logm_iss_full` *(*`logm_agm`*) for full matrices* `A = expm(rand(n))`*. Number of flops: MM (*$2n^3$*), Inv (*$2n^3$*), MRHSLS (*$8/3n^3$*).*

n	10	15	20	25	30
MM	25 (42)	29 (47)	38 (52)	42 (55)	51 (59)
Inv	23 (28)	27 (32)	35 (35)	39 (44)	47 (47)
MRHSLS	10 (4)	10 (4)	10 (4)	11 (5)	14 (5)

In this sample of tests, the number s of square roots carried out by `logm_iss_full` is equal to the number t of AGM steps (5.2)–(5.3) only for the sizes $n = 20$ ($s = t = 5$) and $n = 30$ ($s = t = 6$). For the remaining cases, it is $t = s + 1$ with four iterations (7.2) for the computation of the square root in the last of those AGM steps. The number of inverses is larger in `logm_agm` when $t > s$. In every case, the number of MRHSLS is smaller in `logm_agm` $(t-1)$ than in `logm_iss_full` $(m+1)$. On the other hand, `logm_agm` takes a larger number of matrix multiplications. The decision on the fastest method will depend in many cases on the computing platform; in the case $n = 30$ the winner is certainly `logm_agm`.

8.2. Accuracy issues. In this section we consider certain aspects of the numerical stability (in the usual sense) of our algorithm which uses $\mathrm{AGM}(\epsilon A)$ for the computation of $\log(A)$. With ϵ given in (8.6), half of the significant figures of the

diagonal entries of A are lost in the computation of $Q_0 = (I + \epsilon A)/2$ and, at first glance, one might fear that this will ruin the accuracy of the final result. Fortunately this is not the case, as we will now show. We have $Q_k = \left(\frac{I+\epsilon A}{2}\right) \prod_{i=1}^{k} \left(\frac{I+P_i}{2}\right)$ and the following lemma is a direct application of the error bound given in [17, Lem. 3.6] to our particular case.

LEMMA 8.4. *Assume that P_k is the matrix defined in* (5.2) *and set $X_0 := (I + \epsilon A)/2$, $X_j := (I + P_j)/2$, $j = 1, \ldots, k$. Let $\Delta X_j \in \mathbb{R}^{n\times n}$ and $\delta_j > 0$ be such that $\|\Delta X_j\| \leq \delta_j \|X_j\|$. If $\overline{Q}_k := \prod_{j=0}^{k}(X_j + \Delta X_j)$, then*

$$\left\|\overline{Q}_k - Q_k\right\| \leq \left(\prod_{j=0}^{k}(1+\delta_j) - 1\right) \prod_{j=0}^{k} \|X_j\|.$$

It is straightforward to conclude that, in double format IEEE arithmetic, it is $\|\Delta X_0\| \leq 2^{-53}\|X_0\|$ (the division by 2 is exact, of course) and the first term of the sequence converging to $\mathrm{AGM}(\epsilon A)$ has a tiny relative error. The computation of P_1, in the next step, involves the square root of $P_0 = \epsilon A$ and the full representation of every entry of A will be considered. Because of the excellent properties of the iteration (7.2) (see [20, p. 146]), it will be $\|\Delta X_1\| \leq \delta_1 \|X_1\|$ with a small δ_1, provided that the problem of computing $A^{1/2}$ is well-conditioned (this has been discussed in section 6). For the rest of the computation, some growth of δ_j is to be expected but since we are using standard operations on well-conditioned problems, such growth will be very moderate. In fact, big errors in solving linear systems or inverting a matrix (these operations are in (5.2) and (7.2)) are usually associated with ill-conditioning of the matrix. Fortunately, we are in a good position in respect to this. For our initial matrix $P_0 = \epsilon A$, the condition number of $I + P_0$ is very close to one, and then it grows moderately before decreasing toward one as P_k converges to I. For example, in the computation $\mathrm{AGM}(\epsilon A)$ with $\mathtt{A} = \mathtt{gallery('invhess',50)}$, the sequence of condition numbers of $I + P_k$ is

$$1.0001,\ 1.0204,\ 1.2409,\ 2.2446,\ 6.2542,\ 1.0000,\ 1.0000,\ 1.0000.$$

So, we claim that our computation of $\mathrm{AGM}(\epsilon A)$ is stable. The final computation in

$$\log(A) \approx \mathtt{logm\_agm}(A) := \log(4/\epsilon)I - \frac{\pi}{2}[\mathrm{AGM}(\epsilon A)]^{-1} \tag{8.8}$$

raises the issue of subtractive cancellation in the computation of the diagonal entries of $\mathtt{logm\_agm(A)}$ when the magnitude of these entries is significantly smaller than $|\log(4/\epsilon)|$. The impact of this in the global error

$$\frac{\|\log(A) - \mathtt{logm\_agm}(A)\|}{\|\log(A)\|} \tag{8.9}$$

depends upon how much decrease in the norm of $\frac{\pi}{2}[\mathrm{AGM}(\epsilon A)]^{-1}$ occurs in the subtraction in (8.8). Such decrease will be dramatic when $\frac{\pi}{2}[\mathrm{AGM}(\epsilon A)]^{-1}$ is close to the diagonal matrix $\log(4/\epsilon)I$ which gives $\log(A)$ close to the null matrix. But this is just how our method expresses the ill-conditioning of $\log(A)$ in such case. If $\log(A)$ is well-conditioned, the error (8.9) will not be large, even if severe cancellation occurs

in the diagonal entries. To illustrate this, we have tested our code with the matrix[4]

$$A = \begin{pmatrix} \cos(\lambda) & -\sin(\lambda) \\ \sin(\lambda) & \cos(\lambda) \end{pmatrix}$$

for which, with $\lambda = 1$, we have

$$\log(A) = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}.$$

Our code produces

$$\frac{\pi}{2}[\mathrm{AGM}(\epsilon A)]^{-1} = \begin{pmatrix} 19.754694645958452 & 1.000000000000000 \\ -1.000000000000000 & 19.754694645958445 \end{pmatrix}$$

and $\log(4/\epsilon) = 19.754694645958441$. We recall that $\epsilon = 2^{-26}/\|A\|_F$. The cancellation in the diagonal entries makes the relative error (8.9) grow proportionally to the decrease in norm of the resulting matrix `logm_agm(A)` giving an error equal to $7.14e-15$, whereas the errors of `logm_iss_full`(A) and `logm_frechet_real`(A) are equal to $7.90e-16$ and 0, respectively.

Similarly, a moderate relative error due to such cancellation in the diagonal entries also occurs with `A = gallery('invhess',100)`. In fact, we have $\|\frac{\pi}{2}[\mathrm{AGM}(\epsilon A)]^{-1}\|_F \approx 235$ and $\|$`logm_frechet_real`$(A)\|_F \approx 46$. To compute the error (8.9) we take for $\log(A)$ the result of evaluating the logarithm at 100-decimal digit precision using the Symbolic Math Toolbox and rounding the result to double precision (in doing this, we follow the approach used in [1, sect. 6]). In this way, we have computed the errors of the approximations given by `logm_agm`, `logm_iss_full`, and `logm_frechet_real`, which are, respectively, $7.73e-15$, $1.54e-15$, and $5.05e-15$.

In Figure 3 we give a plot of the normwise relative errors in the approximations given by the three codes for `A = expm(rand(n))` and $n = 10, 15, 20, 25$. We obtained the "exact" $\log(A)$ in the way described before. Solid lines stand for $\kappa_{\log}(A)\cdot u$, where $\kappa_{\log}(A)$ is the estimate for the condition number given by `logm_frechet_real` and u is the unit roundoff. We plot the results obtained in a set of 20 runs for each size n.

In these tests, we found that, for the larger values of n, `sqrtm_db`[5] produces errors about one order of magnitude smaller than the product form `sqrtm_dbp`. A version of `logm_agm` incorporating `sqrtm_db` instead of `sqrtm_dbp` produces slightly more accurate results than the other two codes but it does not compensate for the cost of the extra inverse computations (the product form has the advantage in efficiency over DB that it trades one of the matrix inversions for a matrix multiplication [20, p. 142]).

9. Conclusions and further work. We have shown that the extension of the standard scalar AGM iteration to matrices may be not stable for some matrices with nonreal eigenvalues. We proved several results that made it clear that the so-called Legendre formulation not only is stable for any matrix with no eigenvalues on $\mathbb{R}_0^-$ but also has excellent numerical properties. Furthermore, we found ways to optimize its computation. Because of all this, our algorithm that uses $\mathrm{AGM}(\epsilon A)$ (for a conveniently chosen small ϵ) to compute $\log(A)$ appears to be very promising. The best

[4]This is used in [1] to show the numerical instability of a straightforward implementation of Briggs formula $\log(A) \approx 2^s \log(A^{1/2^s})$.

[5]`sqrtm_db` implements the standard formulation of the DB-iteration; see [21] and [15].

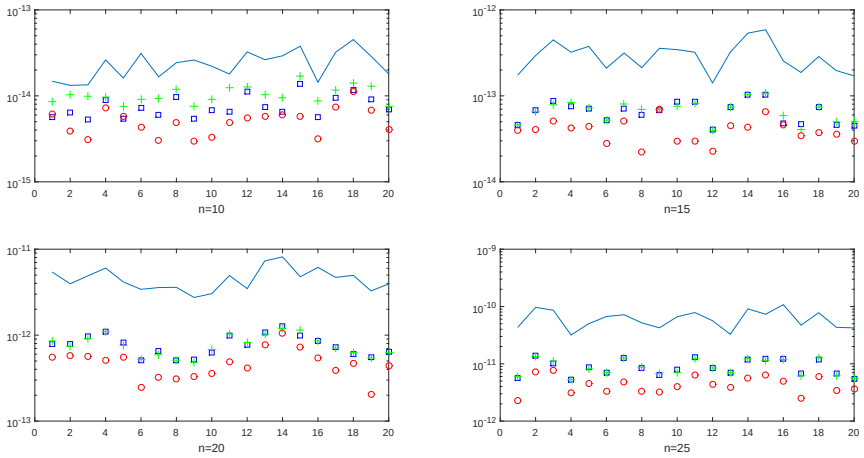

FIG. 3. *Relative errors with* `logm_iss_full` *(squares),* `logm_frechet_real` *(circles), and* `logm_agm` *(+) for* `A = expm(rand(n))`*, with* $n = 10$ *(top left),* $n = 15$ *(top right),* $n = 20$ *(bottom left), and* $n = 25$ *(bottom right). Solid lines are used for* $\kappa_{\log}(A) * u$.

known methods, all based on inverse scaling and squaring, have had many improvements since they first appeared in 1989. It is therefore a very promising fact that our algorithm already produces comparable results, in terms of numerical accuracy, with the best ones. On grounds of efficiency, our reduction-free algorithm, heavily relying upon matrix multiplications (well suited to parallel architectures), is very competitive with `logm_iss_full` for full matrices. Our study has convinced us that the matrix AGM has been unfairly overlooked as a useful tool for matrix computations. We are very much interested in studying the application of our Legendre AGM algorithm to the computation of other matrix functions.

REFERENCES

[1] A. H. Al-Mohy and N. J. Higham, *Improved inverse scaling and squaring for the matrix logarithm*, SIAM J. Sci. Comput., 34 (2012), pp. 153–169.
[2] A. H. Al-Mohy, N. J. Higham, and S. D. Relton, *Computing the Fréchet derivative of the matrix logarithm and estimating the condition number*, SIAM J. Sci. Comput., 35 (2013), pp. 394–410.
[3] A. Besenyei and D. Petz, *Successive iterations and logarithmic means*, Oper. Matrices, 7 (2013), pp. 205–218.
[4] R. Bhatia, *Matrix Analysis*, Springer-Verlag, New York, 1997.
[5] R. Bhatia, *Positive Definite Matrices*, Princeton University Press, Princeton, NJ, 2007.
[6] J. M. Borwein and P. B. Borwein, *The arithmetic-geometric mean and fast computation of elementary functions*, SIAM Rev., 26 (1984), pp. 351–366.
[7] J. M. Borwein and P. B. Borwein, *Pi and the AGM*, Monogr. Études Soc. Math. Canada, Wiley, Toronto, 1987.
[8] R. P. Brent, *Fast multiple-precision evaluation of elementary functions*, J. Assoc. Comput. Machinery, 23 (1976), pp. 242–251.
[9] R. P. Brent and P. Zimmermann, *Modern Computer Arithmetic*, Cambridge Monogr. Appl. Comput. Math., Cambridge University Press, Cambridge, UK, 2011.
[10] R. L. Burden and J. D. Faires, *Numerical Analysis*, 9th ed., Brooks/Cole, Cengage Learning, 2011.

[11] 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, (2001), pp. 1112–1125.
[12] J. E. Cohen and R. D. Nussbaum, *Arithmetic-geometric means of positive matrices*, Math. Proc. Cambridge Philos. Soc., 101, (1987), pp. 209–219.
[13] D. A. Cox, *The arithmetic-geometric mean of Gauss*, Enseign. Math., 30 (1984), pp. 275–330.
[14] E. Deadman, N. J. Higham, and R. Ralha, *Blocked Schur algorithms for computing the matrix square root*, in Applied Parallel and Scientific Computing: 11th International Conference, PARA 2012, Helsinki, P. Manninen and P. Oster, eds., Lecture Notes in Comput. Sci. 7782, Springer-Verlag, Berlin, 2013, pp. 171–182.
[15] E. D. Denman and A. N. Beavers, Jr., *The matrix sign function and computations in systems*, Appl. Math. Comput., 2, (1976), pp. 63–94.
[16] A. Graham, *Kronecker Products and Matrix Calculus with Applications*, Ellis Horwood, Chichester, England, 1981.
[17] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.
[18] N. J. Higham, *Stable iterations for the matrix square roots*, Numer. Algorithms, 15 (1997), pp. 227–242.
[19] N. J. Higham, D. S. Mackey, N. Mackey, and F. Tisseur, *Functions preserving matrix groups and iterations for the matrix square root*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 849–877.
[20] N. J. Higham, *Functions of Matrices: Theory and Computation*, SIAM, Philadelphia, 2008.
[21] N. J. Higham, *The Matrix Function Toolbox*, http://www.maths.manchester.ac.uk/~higham/mftoolbox/.
[22] N. J. Higham, *Matrix Logarithm*, MATLAB Central, http://www.mathworks.com/matlabcentral/fileexchange/33393-matrix-logarithm.
[23] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1994.
[24] C. S. Kenney and A. J. Laub, *Condition estimates for matrix functions*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 191–209.
[25] A. J. Laub, *Matrix Analysis for Scientists and Engineers*, SIAM, Philadelphia, 2005.
[26] W. Luther and W. Otten, *The Complex Arithmetic-Geometric Mean and Multiple-Precision Matrix Functions*, in Scientific Computing and Validated Numerics, Proceedings of SCAN-95, G. Alefeld and A. Frommer, eds., Akademie Verlag, Berlin, 1996.
[27] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*, SIAM, Philadelphia, 2000.
[28] T. Morita and T. Horiguchi, *Convergence of the arithmetic-geometric mean procedure for the complex variables and the calculation of the complete elliptic integrals with complex modulus*, Numer. Math., 20 (1973), pp. 425-430.
[29] R. D. Nussbaum and J. E. Cohen, *The arithmetic-geometric mean and its generalizations for noncommuting linear operators*, Ann. Sc. Norm. Super. Pisa Cl Sci. (4), 15 (1988), pp. 239–308.
[30] S. Relton, *Matrix Logarithm with Frechet Derivatives and Condition Number*, MATLAB Central, http://www.mathworks.com/matlabcentral/fileexchange/38894-matrix-logarithm-with-frechet-derivatives-and-condition-number.
[31] M. Shao, W. Gao, and J. Xue, *Aggressively truncated Taylor series method for accurate computation of exponentials of essentialy nonnegative matrices*, SIAM J. Matrix Anal. Appl., 35 (2014), pp.317–338.
[32] E. U. Stickel, *Fast computation of matrix exponential and logarithm*, Analysis, 5 (1985), pp. 163–173.
[33] E. U. Stickel, *An algorithm for fast high precision computation of matrix exponential and logarithm*, Analysis, 10 (1990), pp. 85–95.