BIT Numerical Mathematics (2019) 59:343–370
https://doi.org/10.1007/s10543-018-00744-1

BIT

Computation of matrix gamma function

João R. Cardoso[1,2] · **Amir Sadeghi**[3]

Received: 8 June 2018 / Accepted: 17 December 2018 / Published online: 2 January 2019

Abstract
Matrix functions have a major role in science and engineering. One of the fundamental matrix functions, which is particularly important due to its connections with certain matrix differential equations and other special matrix functions, is the matrix gamma function. This research article focus on the numerical computation of this function. Well-known techniques for the scalar gamma function, such as Lanczos, Spouge and Stirling approximations, are extended to the matrix case. This extension raises many challenging issues and several strategies used in the computation of matrix functions, like Schur decomposition and block Parlett recurrences, need to be incorporated to make the methods more effective. We also propose a fourth technique based on the reciprocal gamma function that is shown to be competitive with the other three methods in terms of accuracy, with the advantage of being rich in matrix multiplications. Strengths and weaknesses of the proposed methods are illustrated with a set of numerical examples. Bounds for truncation errors and other bounds related with the matrix gamma function will be discussed as well.

Keywords Gamma matrix function · Lanczos method · Spouge method · Stirling approximation · Reciprocal gamma function · Schur decomposition · Block Parlett recurrence

Mathematics Subject Classification 65F30 · 65F60 · 33B15

Communicated by Daniel Kressner.

✉ Amir Sadeghi
drsadeghi.iau@gmail.com

João R. Cardoso
jocar@isec.pt

1 Coimbra Polytechnic – ISEC, Coimbra, Portugal

2 Institute of Systems and Robotics, University of Coimbra, Pólo II, Coimbra, Portugal

3 Department of Mathematics, Robat Karim Branch, Islamic Azad University, Tehran, Iran

1 Introduction

It is well-known that the scalar gamma function is analytic everywhere in the complex plane, with the exception of non-positive integer numbers $\mathbb{Z}_0^-$ (see [5,31] and the references therein). Hence, the general theory of primary matrix functions [24,26] ensures that the matrix gamma function $\Gamma(A)$ is well defined for a given matrix $A \in \mathbb{C}^{n\times n}$ with no eigenvalues on $\mathbb{Z}_0^-$. In the particular case of A having eigenvalues with positive real parts (i.e., $\mathrm{Re}(\lambda) > 0$, for all λ belonging to the spectrum of A, $\sigma(A)$), the matrix gamma function allows the integral representation [29]

$$\Gamma(A) = \int_0^\infty e^{-t} t^{A-I} dt, \tag{1.1}$$

where $t^{A-I} := \exp((A-I)\log t)$. Recall that if z is a complex number not belonging to the closed negative real axis $\mathbb{R}_0^-$, we can define the "scalar-matrix exponentiation" z^M as the function from $\mathbb{C}\times\mathbb{C}^{n\times n}$ to $\mathbb{C}^{n\times n}$ which assigns to each pair (z, M) the $n\times n$ square complex matrix $z^M := e^{M\log z}$, with $\log(z)$ standing for the principal logarithm. This function is a particular case of the more general "matrix–matrix exponentiation" addressed recently in [10].

Likewise, bearing in mind that the reciprocal gamma function, here denoted by $\Delta(z) := \frac{1}{\Gamma(z)}$, is an entire function, we can define $\Delta(A) = (\Gamma(A))^{-1}$, for any matrix $A \in \mathbb{C}^{n\times n}$.

The matrix gamma function has connections with other special functions, which in turn play an important role in solving certain matrix differential equations; see [29] and the references therein. Two of those special functions are the matrix beta and Bessel functions. If both $A, B \in \mathbb{C}^{n\times n}$ have eigenvalues with positive real parts, the matrix beta function [29] is defined by

$$\mathscr{B}(A, B) := \int_0^1 t^{A-I}(1-t)^{B-I} dt.$$

In the case when A and B commute, beta and gamma functions can be related by $\mathscr{B}(A, B) = \Gamma(A)\Gamma(B)\Delta(A+B)$. The matrix Bessel function is defined by [36]:

$$\mathscr{J}_A(z) = \sum_{k=0}^{\infty} \frac{(-1)^k \Delta\,(A + (k+1)I)}{k!} \left(\frac{z}{2}\right)^{A+2kI},$$

where A is assumed to have eigenvalues with positive real parts and z is a complex number lying off the closed negative real axis.

There are several approaches to the computation of direct and reciprocal scalar gamma functions. The most popular are the Lanczos approximation [30,35], Spouge approximation [35,40], Stirling's formula [35,39], continued fractions [11], Taylor series [16], Schmelzer and Trefethen techniques [37] and Temme's formula [41]. In his Ph.D. thesis, Pugh [35] states that the Lanczos approximation is the most feasible and accurate algorithm for approximating the scalar gamma function. If extended to

matrices in a convenient way, we will see that, with respect to a compromise between efficiency and accuracy, the Lanczos method can also be viewed as a serious candidate for the best method for the matrix gamma function.

Another method for computing $\Gamma(A)$ that performs well in terms of accuracy is based on a Taylor expansion of $\Delta(A)$ around the origin, combined with the reciprocal version of the so-called Gauss multiplication formula (see [1, (6.1.20)]):

$$\Delta(z) = (2\pi)^{\frac{m-1}{2}} m^{\frac{1}{2}-z} \prod_{k=0}^{m-1} \Delta\left(\frac{z+k}{m}\right), \tag{1.2}$$

where m is a positive integer. The key point of this formula is that it exploits the fact that such a Taylor expansion is more accurate around the origin. To be more precise, suppose that z is far from the origin so that $|z| \geq 1$ and $m > 1$. Hence $|z| \geq k/(m-1)$, because $k \in \{0, \ldots, m-1\}$, which implies that

$$|z| \geq \frac{|z|+k}{m} \geq \left|\frac{z+k}{m}\right|,$$

showing that $\frac{z+k}{m}$ in the right-hand side of (1.2) is closer to the origin than z, for all k.

We also extend the Spouge and Stirling approximations to matrices in Sects. 4.2 and 4.3, respectively. However, this extension yields poor results if we simply replace the scalar variable z by A. The same holds for Lanczos and Taylor series methods. We must pay attention to some issues arising when dealing with matrices, namely the fact that a matrix may have eigenvalues with positive and negative real parts simultaneously. Our strategy has some similarities with the one used in [13], which includes, in particular, an initial Schur decomposition $A = UTU^*$, with U unitary and T upper triangular, a reorganization of the diagonal entries of T in blocks with "close" eigenvalues and a block Parlett recurrence. It is important to ensure, in particular, a separation between the eigenvalues with negative and positive real parts.

Little attention has been paid to the numerical computation of the matrix gamma function. According to our knowledge, we are the first to investigate thoroughly the numerical computation of this function. Indeed, we have found only two marginal references to the numerical computation of the matrix gamma function in the literature. Schmelzer and Trefethen [37] mentioned that the Hankel contour integral representation given in [37, Eq. (2.1)] can be generalized to square matrices A, and that their methods can be used to compute $\Delta(A)$. They state to have confirmed this by numerical experiments but no results are reported in their paper. They also state that "a drawback of such methods is that it is expensive to compute s_k^{-A} for every node; methods based on the algorithms of Spouge and Lanczos might be more efficient". In [23], at the end of Section 2, Hale et al. mention that their method for computing certain functions of matrices having eigenvalues on or close to the positive real axis can be applied to the gamma function of certain matrices and give an example with a diagonalizable matrix of order 2.

The paper is organized as follows. In Sect. 2 we revisit some properties of the scalar gamma function and recall some of the most well-known methods for its numerical

computation. Section 3 is focused on theoretical properties of the matrix gamma function and on the derivation of bounds for the norm of the matrix gamma and its perturbations. The extension of Lanczos, Spouge and Stirling approximations to the matrix case is carried out in Sect. 4, where a Taylor series expansion of the reciprocal gamma function is also proposed for computing the matrix gamma. To make the approximation techniques more reliable, in Sect. 4.5 we show how to incorporate the Schur–Parlett method and point out its benefits. Numerical experiments are included in Sect. 5 to illustrate the behaviour of the methods and some conclusions are drawn in Sect. 6.

2 Revisiting the scalar gamma and related functions

This section includes a brief revision of some topics related with the scalar gamma function that are relevant for the subsequent material. For readers interested in a more detailed revision, we suggest, among the vast literature, the works [1, Ch. 6], [5] and [14]. See also the references included in [7].

2.1 Definition and properties

Among the many equivalent definitions of the scalar gamma function, the following seems to be common:

$$\Gamma(z) = \int_0^\infty e^{-t} t^{z-1} dt, \quad \mathrm{Re}(z) > 0. \tag{2.1}$$

This integral function can be extended by analytic continuation to all complex numbers except the non-positive integers $z \in \mathbb{Z}_0^- = \{0, -1, -2, \ldots\}$, where the function has simple poles. Unless otherwise is stated, we shall assume throughout the paper that $z \notin \mathbb{Z}_0^-$. Integrating (2.1) by parts, yields

$$\Gamma(z+1) = z\,\Gamma(z). \tag{2.2}$$

Another important identity satisfied by the gamma function is the so-called reflection formula

$$\Gamma(z) = \frac{\pi}{\Gamma(1-z)\sin(\pi z)}, \quad z \notin \mathbb{Z}, \tag{2.3}$$

which is very useful in the computation of the gamma function on the left-half plane.

Closely related to this function is the reciprocal gamma function $\Delta(z)$, which is an entire function. Due to the amenable properties of the reciprocal gamma function, some authors have used it as a means for computing $\Gamma(z)$. Two reasons for this are: $\Delta(z)$ can be represented by the Hankel integral [1,37,42]

$$\Delta(z) = \frac{1}{2\pi i} \int_{\mathscr{C}} t^{-z} e^t \, dt,$$

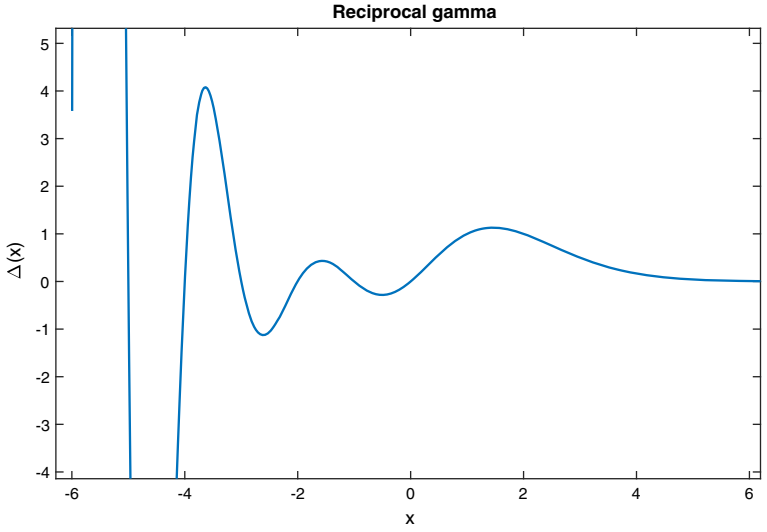

Fig. 1 Graph of the reciprocal gamma function for real arguments

where the path $\mathscr{C}$ is a contour winding around the negative real axis in the anti-clockwise sense, and by the Taylor series with infinite radius of convergence [1,16, 45,46]

$$\Delta(z) = \sum_{k=0}^{\infty} a_k z^k, \quad |z| < \infty, \tag{2.4}$$

where $a_1 = 1$, $a_2 = \gamma$ (here γ stands for the Euler–Mascheroni constant), and the coefficients a_k ($k \geq 2$) are given recursively by [8,45]

$$a_k = \frac{a_2 a_{k-1} - \sum_{j=2}^{k-1} (-1)^j \zeta(j) a_{k-j}}{k-1}, \tag{2.5}$$

with $\zeta(.)$ being the Riemann zeta function. Approximations to $a_2, \ldots, a_{41}$ with 31 digits of accuracy are provided in [45, Table 5]; see also [1, p. 256 (6.1.34)] and [8]. New integral formulae, as well as asymptotic values for a_k, have been recently proposed in [16]. By observing Fig. 1, which displays the graph of the reciprocal gamma as a real function with a real variable, large errors are expected when approximating $\Delta(x)$ by the series (2.4) for negative values of x with large magnitude. So a reasonable strategy is to combine (2.4) with the Gauss formula (1.2). By choosing a suitable m, the magnitude of the argument x is reduced and a truncation of (2.4) is used to approximate $\Delta(x)$ with x having small magnitude. Note that, as shown in Fig. 1, the values of $\Delta(x)$ are moderate for x small.

Another function related to gamma is the *incomplete gamma* function

$$\gamma(z, r) := \int_0^r e^{-t} t^{z-1} dt, \quad \mathrm{Re}(z) > 0, \ r > 0,$$

Table 1 Coefficients $c_k(\alpha)$ ($k = 0, 1, \ldots, 10$) in the Lanczos formula (2.6) for $\alpha = 9$, approximated with 22 digits of accuracy

k	$c_k(9)$
0	1.000000000000000174663
1	5716.400188274341379136
2	− 14815.30426768413909044
3	14291.49277657478554025
4	− 6348.160217641458813289
5	1301.608286058321874105
6	− 108.1767053514369634679
7	2.605696505611755827729
8	$-0.7423452510201416151527 \times 10^{-2}$
9	$0.5384136432509564062961 \times 10^{-7}$
10	$-0.4023533141268236372067 \times 10^{-8}$

which arises in many applications, namely, in Statistics [17–19,38,44].

2.2 Lanczos approximation

In the remarkable paper [30], Lanczos has derived the following formula for the gamma function:

$$\Gamma(z) = \sqrt{2\pi}(z+\alpha-1/2)^{z-1/2}\, e^{-(z+\alpha-1/2)} \left[c_0(\alpha) + \sum_{k=1}^{m} \frac{c_k(\alpha)}{z-1+k} + \epsilon_{\alpha,m}(z) \right], \tag{2.6}$$

where $\mathrm{Re}(z) > 0$, $\alpha > 0$, m is a positive integer, $c_k(\alpha)$ are certain coefficients and $\epsilon_{\alpha,m}(z)$ is the truncation error. The values α and m have been appropriately chosen in order to control either the error $\epsilon_{\alpha,m}$ or the number of terms in the partial fraction expansion.

The values of the coefficients $c_k(\alpha)$ for some parameters α are listed in [30, p. 94]. For instance, the choice $\alpha = 5$ and $m = 6$ guarantees a truncation error of at most 2×10^{-10} for all z in the right-half plane. A more complete list of values of the coefficients c_k is given in [35, App. C] together with empirical estimates for the truncation error. For computations in IEEE double precision arithmetic, Pugh [35] recommends using $\alpha = 10.900511$ and $m = 10$. However, in the implementation of the Lanczos method provided in [20], Godfrey uses $\alpha = 9$ and $m = 10$. It is stated that such values of α and m guarantee a relative error smaller than 10×10^{-13} for a large set of positive real numbers. A new method for computing the coefficients c_k is also suggested, because the one used by Lanczos is rather complicated and sensitive to rounding errors. Table 1 displays the values of the coefficients $c_k(9)$, with 22 digits of accuracy, obtained in [20].

Frequently, to avoid overflow in (2.6), it is more practical to use the following logarithmic version and then exponentiate:

$$\log[\Gamma(z)] = \frac{1}{2}\log(2\pi) + (z-1/2)\log(z+\alpha-1/2) - (z+\alpha-1/2) + \log\left[c_0(\alpha) + \sum_{k=1}^{m}\frac{c_k(\alpha)}{z-1+k} + \epsilon_{\alpha,m}(z)\right]. \tag{2.7}$$

2.3 Spouge approximation

An improvement of the work of Lanczos was given in 1994 by Spouge [40]. There, the following formula

$$\Gamma(z) = \sqrt{2\pi}(z-1+a)^{z-1/2}e^{-(z-1+a)}\left[d_0(a) + \sum_{k=1}^{m}\frac{d_k(a)}{z-1+k} + \varepsilon_a(z)\right], \tag{2.8}$$

which is valid for $\mathrm{Re}(z-1+a) \geq 0$, is proposed. The parameter a is a positive real number, $m = \lceil a \rceil - 1$ ($\lceil . \rceil$ denotes the ceiling of a number), ε_a is the truncation error, $d_0 = 1$, and $d_k(a)$ is given, for $1 \leq k \leq m$, by

$$d_k(a) = \frac{1}{\sqrt{2\pi}}\frac{(-1)^{k-1}}{(k-1)!}(-k+a)^{k-0.5}e^{-k+a}.$$

Let $G_a(z)$ denote the approximation to gamma function obtained from Spouge formula, that is,

$$G_a(z) := \sqrt{2\pi}(z-1+a)^{z-1/2}e^{-(z-1+a)}\left[d_0(a) + \sum_{k=1}^{m}\frac{d_k(a)}{z-1+k}\right],$$

and let $e_a(z)$ be the relative error of the approximation $\Gamma(z) \approx G_a(z)$, i. e.,

$$e_a(z) = \frac{\Gamma(z) - G_a(z)}{\Gamma(z)}. \tag{2.9}$$

Spouge's formula has the simple relative error representation

$$|e_a(z)| = \left|\frac{\varepsilon_a(z)}{\Gamma(z)(z-1+a)^{-(z-1/2)}e^{z-1+a}(\sqrt{2\pi})^{-1}}\right|,$$

where $\varepsilon_a(z)$ is the term appearing in (2.8). Hence,

$$|e_a(z)| \leq \frac{\sqrt{a}}{(2\pi)^{a+1/2}}\frac{1}{\mathrm{Re}(z-1+a)}, \tag{2.10}$$

provided that $a \geq 3$.

2.4 Stirling approximation

The classical Stirling's formula (see [1, p. 257 (6.1.40)] and [39]) is one of the most popular methods for approximating $\Gamma(z)$, when $|z|$ is sufficiently large and $|\arg(z)| < \pi$. This formula becomes effective for either large or small values of $|z|$ if combined with the functional relationship (2.2). The result is the following variant of Stirling's formula:

$$\log\left[\Gamma(z)\prod_{k=1}^{s}(z-1+k)\right] = \left(z+s-\frac{1}{2}\right)\log(z-1+s) - (z-1+s) + \frac{1}{2}\log(2\pi) + \sum_{k=1}^{m}\frac{B_{2k}}{2k(2k-1)\,(z-1+s)^{2k-1}} + R_{m,s}(z), \tag{2.11}$$

where s is a positive integer, and, for z such that $\theta = \arg(z-1+s) \in]-\pi,\ \pi[$,

$$|R_{m,s}(z)| \le \left(\frac{1}{\cos(\theta/2)}\right)^{2m+2}\left|\frac{B_{2m+2}}{(2m+2)(2m+1)(z-1+s)^{2m+1}}\right| \tag{2.12}$$

(see [15, Sec. 6.3]), with B_{2k} being the Bernoulli numbers.

If we fix a complex number z, we can estimate the values of m and s that minimize the bound (2.12); see [35]. The problem becomes more difficult if we want to find optimal values of m and s minimizing the bound for all complex numbers lying on a large region (for instance, the right-hand side of the complex plane). In the following, we describe a naive approach to estimating those values. It is new and works well in the experiments. We will use it later in the numerical experiments of Sect. 5.

Let us fix $m = 12$ to make Stirling approximation comparable with the Spouge method and let us assume that $\theta = \arg(z-1+s) \in]-\pi/2,\ \pi/2[$. Since $\cos(\theta/2) \le \sqrt{2}/2$, the expression on the right-hand side of (2.12) is bounded by

$$\frac{2^{13}B_{26}}{26\times 25|z-1+s|^{25}}. \tag{2.13}$$

Forcing (2.13) to be smaller than or equal to a certain tolerance η, gives

$$|z-1+s| \ge \left(\frac{2^{13}B_{26}}{26\times 25\eta}\right)^{1/25}. \tag{2.14}$$

For instance, if η is the unit roundoff $u = 2^{-53}$ of MATLAB, (2.14) gives $|z-1+s| \ge 8.3$. Denote the right-hand side of (2.14) by $f(\eta)$. A little calculation shows that we can take $s = 0$ whenever $\mathrm{Im}(z) \ge f(\eta)$ or $1 - \mathrm{Re}(z) + \sqrt{[f(\eta)]^2 - \mathrm{Im}(z)^2} \le 0$. Otherwise,

$$s = \left\lceil 1 - \mathrm{Re}(z) + \sqrt{[f(\eta)]^2 - \mathrm{Im}(z)^2}\right\rceil$$

is the smallest integer s such that (2.14) holds.

3 Matrix gamma function

We start this section by revisiting or stating some properties of the matrix gamma function, namely the ones that are relevant for the subsequent material. More theoretical background can be found in [12,28,29]. Then, we propose new bounds for the matrix gamma function and its perturbations and discuss their sharpness.

3.1 Basic properties

Lemma 3.1 *Let $A \in \mathbb{C}^{n\times n}$ have no eigenvalues on $\mathbb{Z}_0^-$. Then the following properties hold:*

(i) $\Gamma(I) = I$, *and* $\Gamma(A+I) = A\,\Gamma(A)$;
(ii) $\Gamma(A)$ *is nonsingular;*
(iii) *If A is block diagonal, $A = \mathrm{diag}(A_1, \ldots, A_m)$, then $\Gamma(A)$ is a block diagonal matrix with the same block structure, that is, $\Gamma(A) = \mathrm{diag}(\Gamma(A_1), \ldots, \Gamma(A_m))$;*
(iv) $\Gamma(A^*) = \Gamma(A)^*$;
(v) *If there is a nonsingular complex matrix S and a complex matrix B such that $A = SBS^{-1}$, then $\Gamma(A) = S\,\Gamma(B)\,S^{-1}$;*
(vi) *Assuming in addition that A does not have any integer eigenvalue, one has the matrix reflection formula*

$$\Gamma(A)\Gamma(I-A) = \pi\ [\sin(\pi A)]^{-1}\,. \tag{3.1}$$

Proof All the statements follow easily from the theory of matrix functions. For further information on the matrix sine function arising in (3.1), see [24, Ch. 12]. □

The next example shows that closed expressions for the matrix gamma function can be very complicated, even for diagonalizable matrices of order 2.

Example 3.1 Consider the diagonalizable matrix $A = \left[\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right]$, which has eigenvalues $\lambda_1 = \frac{(a+d)-\Omega}{2}$ and $\lambda_2 = \frac{(a+d)+\Omega}{2}$, where $\Omega = \sqrt{(a-d)^2+4bc}$ is assumed to be non zero. The eigenvalues and eigenvectors of A are

$$D = \mathrm{diag}(\lambda_1, \lambda_2) = \begin{pmatrix} \frac{(a+d)-\Omega}{2} & 0 \\ 0 & \frac{(a+d)+\Omega}{2} \end{pmatrix}, \quad X = \begin{pmatrix} -\frac{(a-d)+\Omega}{2c} & \frac{(a-d)+\Omega}{2c} \\ 1 & 1 \end{pmatrix}.$$

Hence, $\Gamma(A)$ can be evaluated by the spectral decomposition $\Gamma(A) = X\Gamma(D)X^{-1}$ as following:

$$\begin{aligned}\Gamma(A) &= \begin{pmatrix} -\frac{(a-d)+\Omega}{2c} & \frac{(a-d)+\Omega}{2c} \\ 1 & 1 \end{pmatrix} \begin{pmatrix} \Gamma(\lambda_1) & 0 \\ 0 & \Gamma(\lambda_2) \end{pmatrix} \begin{pmatrix} -\frac{(a-d)+\Omega}{2c} & \frac{(a-d)+\Omega}{2c} \\ 1 & 1 \end{pmatrix}^{-1} \\ &= \frac{1}{2\Omega} \begin{pmatrix} \Gamma(\lambda_1)(d-a+\Omega) + \Gamma(\lambda_2)(a-d+\Omega) & -2b(\Gamma(\lambda_1)-\Gamma(\lambda_2)) \\ -2c(\Gamma(\lambda_1)-\Gamma(\lambda_2)) & \Gamma(\lambda_1)(a-d+\Omega) + \Gamma(\lambda_2)(d-a+\Omega) \end{pmatrix}.\end{aligned}$$

3.2 Norm bounds

Before proceeding with investigations on bounding the norm of the matrix gamma function, we shall recall that the *incomplete gamma* function with matrix arguments can be defined by [36]

$$\gamma(A, r) := \int_0^r e^{-t} t^{A-I} dt,$$

and its *complement* by

$$\Gamma(A, r) := \int_r^\infty e^{-t} t^{A-I} dt,$$

where it is assumed that $A \in \mathbb{C}^{n\times n}$ satisfies $\mathrm{Re}(\lambda) > 0$, for all $\lambda \in \sigma(A)$, and r is a positive real number. Additionally, we remind the definition and notation to the *spectral abscissa* of A:

$$\alpha(A) := \max\{\mathrm{Re}(\lambda) : \lambda \in \sigma(A)\}. \tag{3.2}$$

Theorem 3.1 *Given $A \in \mathbb{C}^{n\times n}$ satisfying $\mathrm{Re}(\lambda) > 0$, for all $\lambda \in \sigma(A)$, let $A = U(D + N)U^*$ be its Schur decomposition, with U, D and N being, respectively, unitary, diagonal and strictly upper triangular matrices. If $r \geq 1$, then the complement of gamma function allows the following bound, with respect to the 2-norm:*

$$\|\Gamma(A, r)\|_2 \leq \sum_{k=0}^{n-1} \frac{\|N\|_2^k}{k!} \Gamma\left(\alpha(A) + k, r\right), \tag{3.3}$$

where $\alpha(A)$ is the spectral abscissa of A.

Proof We start by recalling the following bound to the matrix exponential [24, Thm. 10.12]:

$$\left\|e^{At}\right\|_2 \leq e^{\alpha(A)t} \sum_{k=0}^{n-1} \frac{\|Nt\|_2^k}{k!}. \tag{3.4}$$

From (3.4), and attending that $|\log(t)| \leq t$, for all $t \geq 1$, it easily follows that

$$\left\|t^{A-I}\right\| \leq e^{\alpha(A-I)\log(t)} \sum_{k=0}^{n-1} \frac{|\log(t)|^k \|N\|_2^k}{k!} \leq t^{\alpha(A)-1} \sum_{k=0}^{n-1} \frac{t^k \|N\|_2^k}{k!}.$$

Hence, for $r \geq 1$,

$$\begin{aligned}\|\Gamma(A,r)\|_2 &= \left\| \int_r^\infty e^{-t} t^{A-I}\, dt \right\|_2 \le \int_r^\infty e^{-t} \left\| t^{A-I} \right\|_2 dt \\ &\le \int_r^\infty e^{-t}\, t^{\alpha(A)-1} \sum_{k=0}^{n-1} \frac{t^k \|N\|_2^k}{k!}\, dt \\ &= \sum_{k=0}^{n-1} \frac{\|N\|_2^k}{k!} \int_r^\infty e^{-t} t^{\alpha(A)-1} t^k\, dt \;=\; \sum_{k=0}^{n-1} \frac{\|N\|_2^k}{k!}\, \Gamma\left(\alpha(A)+k, r\right).\end{aligned}$$

□

The previous theorem gives a scalar upper bound for the error arising in the approximation of the matrix gamma function by the matrix incomplete gamma function. Indeed, since $\Gamma(A) = \gamma(A,r) + \Gamma(A,r)$, for any $r > 0$, the error of the approximation $\Gamma(A) \approx \gamma(A,r)$, with $r \ge 1$, is bounded by (3.3). At this stage, we may ask why such an expensive upper bound involving the Schur decomposition of A (its computation requires about $25n^3$ flops) should be used instead of a cheaper one. In fact, as explained in [43], there are many cheaper bounds to the matrix exponential than (3.4), but they are not sharp in general. Other reason is that our algorithms, which will be proposed later, are based on the Schur decomposition and so bounds based on Schur decompositions can be computed at a negligible cost.

The next result provides an upper bound to the norm of the matrix gamma function.

Corollary 3.1 *Assume that the assumptions of Theorem 3.1 are valid and denote* $\beta(A) := \min\{\mathrm{Re}(\lambda) : \lambda \in \sigma(A)\}$. *Then:*

(i) *If* $N = 0$,

$$\|\Gamma(A)\|_2 \le \gamma\left(\beta(A), 1\right) + \Gamma\left(\alpha(A), 1\right); \tag{3.5}$$

(ii) *If* $N \neq 0$ *and* $\beta(A) > n-1$,

$$\|\Gamma(A)\|_2 \le \sum_{k=0}^{n-1} \frac{\|N\|_2^k}{k!} \left[\gamma\left(\beta(A)-k, 1\right) + \Gamma\left(\alpha(A)+k, 1\right)\right]. \tag{3.6}$$

Proof Accounting that $\Gamma(A) = \gamma(A,1) + \Gamma(A,1)$ [here we consider $r = 1$ because of the bound (3.7)], by Theorem 3.1, one just needs to show that

$$\|\gamma(A,1)\|_2 \le \sum_{k=0}^{n-1} \frac{\|N\|_2^k}{k!}\, \gamma\left(\beta(A)-k, 1\right).$$

This result follows if we use the inequality

$$|\log(t)| \le t^{-1}, \tag{3.7}$$

where $0 < t \le 1$, and the same strategy of the proof of Theorem 3.1. Notice that (3.7) does not hold for $t = 0$. However, this is not a problem because $\lim_{t\to 0^+} e^{-t} t^{A-I} = 0$.

□

The bound (3.5) provided in the previous corollary corresponds to the case when A is a normal matrix, that is, $A = UDU^*$, with U unitary and $D = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ diagonal. With respect to the 2-norm, we know that

$$\|\Gamma(A)\|_2 = \|\Gamma(D)\|_2 = \max_k |\Gamma(\lambda_k)| = \max_k |\gamma(\lambda_k, 1) + \Gamma(\lambda_k, 1)|. \tag{3.8}$$

Let λ_{k_0} be the eigenvalue of A where the maximum value in (3.8) is attained. If $\gamma(\lambda_{k_0}, 1)$ or $\Gamma(\lambda_{k_0}, 1)$ have small magnitude, we shall expect sharp values provided by (3.5). This happens, for instance, with Hilbert matrices of several sizes, where that bound gives very good results. If A is a scalar matrix, that is, $A = \alpha I$, with $\alpha \in \mathbb{C}$, then (3.5) gives the exact value to $\|\Gamma(A)\|_2$.

The sharpness of bound (3.6) in the non-normal case is more difficult to discuss. Sharp and non sharp results may be obtained, which apparently depend on the non normality of A.

We end this section with a perturbation bound to the matrix gamma function. Now the norm can be an arbitrary subordinate matrix norm.

Theorem 3.2 *Let $A, E \in \mathbb{C}^{n\times n}$ and assume that the eigenvalues of A and $A + E$ have positive real parts. Then, for a given subordinate matrix norm,*

$$\|\Gamma(A + E) - \Gamma(A)\| \leq \|E\| \left[\gamma(-\mu + 1, 1) + \Gamma(\mu + 1, 1)\right],$$

where $\mu := \max\{\|A + E - I\|, \|A - I\|\}$.

Proof From [10, Thm. 5.1], a simple calculation shows that the following inequality holds for any subordinate matrix norm:

$$\left\| t^{A+E-I} - t^{A-I} \right\| \leq \|E\|\, e^{|\log(t)|\,\mu}, \tag{3.9}$$

with $\mu := \max\{\|A + E - I\|, \|A - I\|\}$. Hence

$$\left\| t^{A+E-I} - t^{A-I} \right\| \leq \begin{cases} \|E\|\, t^{\mu}, & \text{if } t \geq 1 \\ \|E\|\, t^{-\mu}, & \text{if } 0 < t < 1 \end{cases}. \tag{3.10}$$

Since

$$\Gamma(A+E)-\Gamma(A)=\int_0^1 e^{-t}\left(t^{A+E-I} - t^{A-I}\right)\,dt + \int_1^\infty e^{-t}\left(t^{A+E-I} - t^{A-I}\right)\,dt,$$

the result follows by taking norms and attending to (3.10). □

4 Strategies for approximating the matrix gamma function

This section is dedicated to the numerical computation of the matrix gamma function. We start by extending the well-known scalar methods of Lanczos, Spouge and Stirling

to matrices. A method based on the reciprocal gamma function used in combination with the Gauss multiplication formula is also addressed.

4.1 Lanczos method

Before stating the Lanczos formula for matrices, we shall recall the concept of matrix–matrix exponentiation [6,10].

If A is an $n \times n$ square complex matrix with no eigenvalues on the closed negative real axis $\mathbb{R}_0^-$ and B is an arbitrary square complex matrix of order n, the matrix–matrix exponentiation A^B is defined as $A^B := e^{\log(A)B}$, where e^X stands for the exponential of the matrix X and $\log(A)$ denotes the principal logarithm of A. For background on matrix exponentials and matrix logarithms see [24,26] and the references therein. Regarding the computation of matrix exponential and logarithm in the recent versions of MATLAB, the function `expm` corresponds the algorithm provided in [2] and `logm` computes the matrix logarithm using an algorithm investigated in [3,4].

Assuming that A is an $n \times n$ matrix with all of its eigenvalues having positive real parts and $\alpha > 0$, the matrix version of Lanczos formula (2.6) can be written as

$$\Gamma(A) = \sqrt{2\pi}\,(A+(\alpha-0.5)I)^{A-0.5\,I}\;e^{-(A+(\alpha-0.5)I)} \times \left[c_0(\alpha)I + \sum_{k=1}^{m} c_k(\alpha)\,(A+(k-1)I)^{-1} + \epsilon_{\alpha,m}(A)\right], \tag{4.1}$$

where $c_k(\alpha)$ are the Lanczos coefficients, which depend on the parameter α. Discarding the error term $\epsilon_{\alpha,m}(A)$ in the right-hand side of (4.1), yields the approximation

$$\Gamma(A) \approx \sqrt{2\pi}\,(A+(\alpha-0.5)I)^{A-0.5\,I}\;e^{-(A+(\alpha-0.5)I)} \times \left[c_0(\alpha)I + \sum_{k=1}^{m} c_k(\alpha)\,(A+(k-1)I)^{-1}\right]. \tag{4.2}$$

Attending to our discussion in Sect. 2.2, in Algorithm 4.1 below we will consider the values $m = 10$ and $\alpha = 9$ suggested in [20], whose coefficients are given in Table 1. To avoid overflow, Algorithm 4.1 uses the following logarithmic version of Lanczos formula, which holds for matrices with spectrum on the right-hand side of the complex plane:

$$\log[\Gamma(A)] \approx 0.5\log(2\pi)I + (A-0.5I)\log(A+8.5\,I) - (A+8.5\,I) + \log\left[\sum_{k=1}^{10} c_k(9)\,(A+(k-1)I)^{-1}\right] \tag{4.3}$$

Algorithm 4.1 This algorithm approximates $\Gamma(A)$ by the Lanczos formula (4.2), where $A \in \mathbb{C}^{n\times n}$ is a matrix with spectrum satisfying one and only one of the following conditions: (i) $\sigma(A)$ is contained in the open right-half plane; or (ii) $\sigma(A)$ does not contain negative integers and lies on the open left-half plane.

1. `if` Re(trace$(A)) \geq 0$
2. Compute $\Gamma(A)$ by (4.3);
3. `else`
4. $S = \sin(\pi A)$;
5. Compute $G = \Gamma(I - A)$ by (4.3);
6. $\Gamma(A) \approx \pi(SG)^{-1}$;
7. `end`

In the more general case of A simultaneously having eigenvalues with positive and negative real parts, the Lanczos formula needs to be combined with a strategy separating the eigenvalues lying on the left-half plane with the ones in the right-half plane. This will be carried out in Sect. 4.5 by means of the so called Schur–Parlett method.

4.2 Spouge method

Let $A \in \mathbb{C}^{n\times n}$ be a matrix whose eigenvalues all have positive real parts and $a > 0$. The matrix version of Spouge formula (2.8) is:

$$\Gamma(A) = \sqrt{2\pi}\,(A + (a-1)I)^{A-0.5\,I}\; e^{-(A+(a-1)I)} \times \left[d_0(a)I + \sum_{k=1}^{m} d_k(a)\,(A + (k-1)I)^{-1} + \epsilon_a(A) \right], \tag{4.4}$$

where $d_k(a)$ are the Spouge coefficients, which vary with a, and $m = \lceil a \rceil - 1$. Ignoring the error term $\epsilon_a(A)$ in the right-hand side of (4.4), we have

$$\Gamma(A) \approx \sqrt{2\pi}\,(A + (a-1)I)^{A-0.5\,I}\; e^{-(A+(a-1)I)} \times \left[d_0(a)I + \sum_{k=1}^{m} d_k(a)\,(A + (k-1)I)^{-1} \right]. \tag{4.5}$$

Let us denote the relative truncation error of the approximation (4.5) by

$$\mathscr{E}_a(A) := \frac{\|\Gamma(A) - G_a(A)\|}{\|\Gamma(A)\|}, \tag{4.6}$$

where $G_a(A)$ denotes the right-hand side of (4.5); see also Sect. 2.3. Let $\kappa_p(X) := \|X\|_p \|X^{-1}\|_p$ denotes the condition number of the matrix X with respect to a p-norm, with $p = 1, 2, \infty$. The next lemma gives a bound for the relative error $\mathscr{E}_a(A)$ with respect to p-norms for the case when A is diagonalizable.

Lemma 4.1 *Let $A \in \mathbb{C}^{n\times n}$ be a diagonalizable matrix ($A = PDP^{-1}$, with P non-singular and $D := \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$) having all eigenvalues with positive real parts, that is, $\beta(A) = \min\{\mathrm{Re}(\lambda) : \lambda \in \sigma(A)\}$ satisfies $\beta(A) > 0$. For $a \geq 3$ and $\mathscr{E}_a(A)$ given as in (4.6),*

$$\mathscr{E}_a(A) \le \kappa_p(P) \frac{\sqrt{a}}{(2\pi)^{a+1/2}\,(\beta(A) - 1 + a)}. \tag{4.7}$$

Proof For any z in the open right-half plane, we know, from Sect. 2.3, that

$$\Gamma(z) - G_a(z) = e_a(z)\Gamma(z). \tag{4.8}$$

where $e_a(z)$ is defined by (2.9). Since A has all eigenvalues with positive real parts and the functions involved in (4.8) are analytic on the right half-plane, the identity $\Gamma(A) - G_a(A) = e_a(A)\Gamma(A)$ is valid. Now, because A is diagonalizable, $\Gamma(A) - G_a(A) = P\, e_a(D)\, P^{-1}\Gamma(A)$. Hence, for p-norms, we have

$$\|\Gamma(A) - G_a(A)\|_p \le \kappa_p(P)\|e_a(D)\|_p\|\Gamma(A)\|_p \le \kappa_p(P) \max_{i=1,\ldots,n} |e_a(\lambda_i)|\|\Gamma(A)\|_p,$$

and, consequently,

$$\frac{\|\Gamma(A) - G_a(A)\|_p}{\|\Gamma(A)\|_p} \le \kappa_p(P) \max_{i=1,\ldots,n} |e_a(\lambda_i)|.$$

Therefore, the inequality (4.7) follows from the Spouge scalar error bound (2.10). □

We have investigated the sharpness of bound (4.7) with some experiments involving diagonalizable matrices with eigenvalues lying on the right-half plane. The most relevant conclusion is that the sharpness depends mainly on the condition number of the matrix P. For normal matrices, for which P has norm one with respect to the 2-norm, the results are very satisfactory. In contrast, if $\kappa_p(P)$ is large, the results may be poor. An example is the matrix `A=expm(C)`, where C is the Chebyshev spectral differentiation matrix of order 5 (see "chebspec" in MATLAB's gallery), for which P has a condition number of about 10^{12}.

For a general matrix A (diagonalizable or not), assume that the function $E_a(z) = \Gamma(z) - G_a(z)$ (absolute error) is analytic on a closed convex set Ω containing the spectrum of A. A direct application of [24, Thm. 4.28] (check also [21, Thm. 9.2.2]), yields the bound (with respect to Frobenius norm)

$$\|E_a(A)\|_F \le \max_{i \le k \le n-1} \frac{\omega_k}{k!} \|(I - |N|)^{-1}\|_F, \tag{4.9}$$

where $U^*AU = T = \mathrm{diag}(\lambda_1, \ldots, \lambda_n) + N$ is the Schur decomposition of A, with T upper triangular, N strictly upper triangular, and $\omega_k = \sup_{z\in\Omega} |E_a^{(k)}(z)|$. One drawback of bound (4.9) is the need of the derivatives of $E_a(z)$ up to order $n - 1$.

Providing that A is diagonalizable ($A = SDS^{-1}$), with S not having a large condition number, the choice $a = 12.5$ (and hence $m = 12$) seems to be suitable for working in IEEE double precision environments. This has been confirmed by many numerical experiments (not reported here) we have carried out. $a = 12.5$ is also the value considered in [35] for scalars. The corresponding Spouge coefficients are given in Table 2.

Table 2 Coefficients $d_k(a)$ $(k = 0, 1, \ldots, 12)$ in the Spouge formula (4.4) for $a = 12.5$, approximated with 22 digits of accuracy

k	$d_k(12.5)$
0	1
1	133550.5029424774402287
2	− 492930.9352993603097275
3	741287.4736976117128506
4	− 585097.3776039966614917
5	260425.2703303852758836
6	− 65413.35339611420204164
7	8801.459635084211186040
8	− 564.8050241289801078892
9	13.803798339181415855137
10	$-0.8078176169895076585981 \times 10^{-1}$
11	$0.3479741445742458983261 \times 10^{-4}$
12	$-0.5689271227504240383584 \times 10^{-11}$

To avoid overflow, Algorithm 4.2 uses the logarithmic version of Spouge formula, which is valid for matrices with eigenvalues which have positive real parts:

$$\log[\Gamma(A)] \approx 0.5 \log(2\pi) + (A - 0.5\, I) \log(A + 11.5I) - (A + 11.5I) + \log\left[d_0(12.5) I + \sum_{k=1}^{12} d_k(12.5)\, (A + (k-1)I)^{-1} \right]. \quad (4.10)$$

Algorithm 4.2 This algorithm approximates $\Gamma(A)$ by the Spouge formula (4.4), where $a = 12.5, m = 12$ and $A \in \mathbb{C}^{n\times n}$ is a nonsingular matrix whose spectrum satisfies one and only one of the following conditions: (i) $\sigma(A)$ is contained in the closed right-half plane; or (ii) $\sigma(A)$ does not contain negative integers and lies on the open left-half plane.

1. `if` Re(trace(A)) ≥ 0
2. Compute $\Gamma(A)$ by (4.10);
3. `else`
4. $S = \sin(\pi A)$;
5. Compute $G = \Gamma(I - A)$ by (4.10);
6. $\Gamma(A) \approx \pi(SG)^{-1}$;
7. `end`

4.3 Stirling method

Let $A \in \mathbb{C}^{n\times n}$ have all eigenvalues with positive real parts. The extension of the logarithmic version of the Stirling formula (2.11) to matrices reads as:

$$\log\left[\Gamma(A)\prod_{k=1}^{s}(A+(k-1)I)\right] = (A+(s-0.5)I)\,\log(A+(s-1)I) - (A+(s-1)I) + 0.5\log(2\pi) + \sum_{k=1}^{m}\frac{B_{2k}}{2k(2k-1)\,(A+(s-1)I)^{2k-1}} + R_{m,s}(A). \tag{4.11}$$

Discarding the error term in (4.11) and denoting

$$S_{m,s}(A) := (A+(s-0.5)I)\,\log(A+(s-1)I) - (A+(s-1)I) + 0.5\log(2\pi) + \sum_{k=1}^{m}\frac{B_{2k}}{2k(2k-1)\,(A+(s-1)I)^{2k-1}},$$

we have the approximation

$$\Gamma(A) \approx \left(\prod_{k=1}^{s}(A+(k-1)I)\right)^{-1} e^{S_{m,s}(A)}. \tag{4.12}$$

A similar error analysis to the one did in Sect. 4.2 could be carried out for the Stirling method, but it is omitted. We just focus on stating the algorithm for the Stirling method that will be used in the experiments. An extension of the technique to find s proposed in Sect. 2.4 to matrices will be incorporated.

Algorithm 4.3 This algorithm evaluates $\Gamma(A)$ using the Stirling formula (4.11), where $m = 12$, $\eta = 2^{-53}$ in (2.14), and $A \in \mathbb{C}^{n\times n}$ is a nonsingular matrix with spectrum satisfying one and only one of the following conditions: (i) $\sigma(A)$ is contained in the closed right-half plane; or (ii) $\sigma(A)$ does not contain negative integers and lies on the open left-half plane.

1. `if` $\mathrm{Re}(\mathrm{trace}(A)) \geq 0$
2. $z = \mathrm{trace}(A)/n$;
3. `if` $\mathrm{Im}(z) \geq 8.3$ or $1 - \mathrm{Re}\left(z + \sqrt{8.3^2 - \mathrm{Im}(z)^2}\right) \leq 0$
4. $s = 0$
5. `else`
6. $s = \lceil 1 - \mathrm{Re}(z) + \sqrt{8.3^2 - \mathrm{Im}(z)^2}\rceil$
7. `end`
8. Compute $\Gamma(A)$ by (4.12);
9. `else`
10. $S = \sin(\pi A)$;
11. Compute $G = \Gamma(I - A)$ by (4.12);
12. $\Gamma(A) \approx \pi(SG)^{-1}$;
13. `end`

4.4 Reciprocal gamma function

For any matrix $A \in \mathbb{C}^{n\times n}$, the reciprocal matrix gamma function allows the following Taylor expansion around the origin:

$$\Delta(A) = (\Gamma(A))^{-1} = \sum_{k=0}^{\infty} a_k A^k, \tag{4.13}$$

where a_k can be evaluated through the recursive formula (2.5). According to our discussion in Sect. 2.1, truncating (4.13) to approximate $\Delta(A)$ is recommended only when the spectral radius of A, $\rho(A)$, is small. If A has a large spectral radius, then it is advisable to combine (4.13) with Gauss formula (1.2).

For matrices having small norm ($\|A\| \leq 1$), the next result proposes a bound for the truncation error of (4.13) in terms of a scalar convergent series.

Lemma 4.2 *If $A \in \mathbb{C}^{n\times n}$ with $\|A\| \leq 1$ and a_k are the coefficients in (4.13), then*

$$\left\| \Delta(A) - \sum_{k=1}^{m} a_k A_k \right\| \lesssim \frac{4}{\pi^2} \sum_{k=m+1}^{\infty} \frac{\sqrt{k!}}{(m+1)!(k-m-1)!}. \tag{4.14}$$

Proof Using the truncation error bound of [32], we have

$$\left\| \Delta(A) - \sum_{k=1}^{m} a_k A_k \right\| \leq \frac{1}{(m+1)!} \max_{s\in[0,1]} \left\| A^{m+1} \Delta^{(m+1)}(sA) \right\|.$$

Since the m-th derivative of $\Delta(z)$ is given by

$$\Delta^{(m)}(z) = \sum_{k=m+1}^{\infty} k(k-1)\dots(k-m+1) a_k z^{k-m},$$

we can write

$$\Delta^{(m+1)}(sA) = \sum_{k=m+1}^{\infty} k(k-1)\dots(k-m+1) a_k s^{k-m-1} A^{k-m-1},$$

yielding

$$A^{m+1}\Delta^{(m+1)}(sA) = \sum_{k=m+1}^{\infty} k(k-1)\dots(k-m+1) a_k s^{k-m-1} A^{k}.$$

Taking norms and accounting that $s \leq 1$ and $\|A\| \leq 1$,

$$\left\| A^{m+1}\Delta^{(m+1)}(sA) \right\| \leq \sum_{k=m+1}^{\infty} k(k-1)\dots(k-m+1)|a_k|.$$

Using $|a_k| \lesssim 4/(\pi^2\sqrt{\Gamma(n+1)})$ (see [8]), the relationship (4.14) follows. □

For convenience, let us change the index k in the series in the right-hand side of (4.14) to $k = p + m$. Then the series can be rewritten as

$$\frac{4}{\pi^2}\sum_{p=1}^{\infty}\frac{\sqrt{(p+m)!}}{(m+1)!(p-1)!}. \tag{4.15}$$

By the d'Alembert ratio test, we can easily show that (4.15) is convergent. Indeed, denoting

$$b_p := \frac{\sqrt{(p+m)!}}{(m+1)!(p-1)!},$$

one has $\lim_{p\to\infty} b_{p+1}/b_p = 0$. The exact value of (4.15) is unknown and so we will work with estimates. We have approximated the sum of the series (4.15) in MATLAB, using variable precision arithmetic with 250 digits, by taking $p = 2000$. Assuming that $\|A\| \leq 1$ and using the right-hand side of (4.14), we see that, for $m = 33$,

$$\Delta(A) \approx \sum_{k=1}^{33} a_k A_k,$$

with a truncation error of about 1.1294×10^{-17}. This means that 33 terms of the reciprocal gamma function series is a reasonable choice if the calculations are performed in IEEE double precision arithmetic environments.

For the more general case when $\|A\| > 1$, our strategy is to combine (4.13) with the Gauss multiplication formula

$$\Delta(A) = (2\pi)^{\frac{r-1}{2}}\, r^{I/2-A} \prod_{k=0}^{r-1} \Delta\left(\frac{A+kI}{r}\right), \tag{4.16}$$

where r is a positive integer.

Given a positive real number μ, we aim to find a positive integer r for which

$$\rho\left(\frac{A+(r-1)I}{r}\right) \leq \mu,$$

or, equivalently,

$$\rho\left(A+(r-1)I\right) \leq r\mu. \tag{4.17}$$

This guarantees that the arguments of the reciprocal gamma function arising in the right-hand side of (4.16) are matrices with eigenvalues lying on the circle with centre at the origin and radius μ. Hence, if μ is small enough, taking an r satisfying (4.17) and a suitable number of terms m in (4.13) will give an approximation to $\Delta(A)$ with good accuracy. More details are given in the following.

Since $\rho(A+B) \leq \rho(A) + \rho(B)$, for two given commuting matrices [27, p. 117], we know that

$$\rho\left(A+(r-1)I\right) \leq \rho(A)+(r-1).$$

Finding the smallest r such that

$$\rho(A)+(r-1) \leq r\mu, \tag{4.18}$$

yields an r satisfying (4.17). Hence, providing that $\rho(A) > 1$ and $\mu > 1$, one can take

$$r = \left\lceil \frac{\rho(A)-1}{\mu-1} \right\rceil.$$

What is difficult in this approach is finding optimal values for μ and r in order to minimize the number operations involved, while guaranteeing a small error. Based on several tests we have carried out (not reported here), a reasonable choice for working in IEEE double precision arithmetic seems to be $\mu = 3$ and $m = 50$ [number of terms taken in (4.13)].

Now we summarize the computation of the gamma function by means of its reciprocal in the following algorithm.

Algorithm 4.4 This algorithm approximates $\Gamma(A)$, where $A \in \mathbb{C}^{n\times n}$ is a non-singular matrix with no negative integers eigenvalues, by the reciprocal gamma function series combined with the Gauss multiplication formula. Assume that the coefficients $a_1, \ldots, a_{50}$ in (4.13) are available.

1. $\mu = 3$;
2. `if` $\rho(A) \leq \mu$
3. $\quad \widetilde{\Delta} = \sum_{k=1}^{50} a_k A^k$;
4. $\quad \Gamma(A) \approx (\widetilde{\Delta})^{-1}$;
5. `else`
6. $\quad$ Compute $r = \left\lceil \frac{\rho(A)-1}{\mu-1} \right\rceil$;
7. $\quad \widetilde{\Delta} = \sum_{k=0}^{50} a_k \left(\frac{A}{r}\right)^k$;
8. $\quad$ `for` $p = 1 : r-1$
9. $\quad\quad$ Compute $\widetilde{\Delta} = \widetilde{\Delta} \sum_{k=0}^{50} a_k \left(\frac{A+pI}{r}\right)^k$;
10. $\quad$ `end`
11. $\quad \widetilde{\Delta} = (2\pi)^{\frac{r-1}{2}}\, r^{0.5\,I-A}\, \widetilde{\Delta}$;
12. $\quad \Gamma(A) \approx (\widetilde{\Delta})^{-1}$;
13. `end`

Many techniques for evaluating the matrix polynomials in steps 3 and 9 of the previous algorithm are available [24, Sect. 4.2]. One of the most popular is the Horner method, which is accessible in MATLAB through the function `polyvalm`. However, in our implementations of the algorithms, whose results will be presented in Sect. 5, we will evaluate the matrix polynomials by means of the less expensive Paterson–Stockmeyer method [21,24,34]. A MATLAB code is available in [25].

4.5 Schur–Parlett approach

We start by revisiting the Schur decomposition and the block-Parlett recurrence. This block recurrence is an extension of the original Parlett method proposed in [33]. For additional information, we refer the reader to [13] and [24, Ch. 9].

Given $A \in \mathbb{C}^{n\times n}$, the Schur decomposition states that there exists a unitary matrix U and an upper triangular matrix T such that $A = UTU^*$, with T displaying the eigenvalues of A in the diagonal. Hence, assuming that A is nonsingular with no negative integers eigenvalues, $\Gamma(A) = U\,\Gamma(T)\,U^*$, meaning that the evaluation of $\Gamma(A)$ may be reduced to the computation of the gamma function of a triangular matrix. Let

$$T = \begin{bmatrix} T_{11} & T_{12} & \dots & T_{1p} \\ 0 & T_{22} & \dots & T_{2p} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & T_{pp} \end{bmatrix} \in \mathbb{C}^{n\times n}, \quad \sigma(T) \cap \mathbb{Z}_0^- = \emptyset, \tag{4.19}$$

be written as a $(p \times p)$-block-upper triangular, with the blocks T_{ii} $(i = 1, \dots, p)$ being square with no common eigenvalues, that is,

$$\sigma(T_{ii}) \cap \sigma(T_{jj}) = \emptyset, \quad i, j = 1, \dots, p, \quad i \neq j. \tag{4.20}$$

Let us denote

$$G := \Gamma(T) = \begin{bmatrix} G_{11} & G_{12} & \dots & G_{1p} \\ 0 & G_{22} & \dots & G_{2p} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & G_{pp} \end{bmatrix}, \tag{4.21}$$

where G_{ij} has the same size as T_{ij} $(i, j = \dots, p)$. Recall that the diagonal blocks of G are given by $G_{ii} = \Gamma(T_{ii})$. Since $GT = TG$, it can be shown that

$$G_{ij}T_{jj} - T_{ii}G_{ij} = T_{ij}G_{jj} - G_{ii}T_{ij} + \sum_{k=i+1}^{j-1} (T_{ik}G_{kj} - G_{ik}T_{kj}) \quad i < j. \tag{4.22}$$

To find the blocks of G, we start by computing the blocks on diagonal $G_{ii} = \Gamma(T_{ii})$. This can be done by Algorithms 4.1, 4.2 or 4.4. In terms of computational cost, there is the advantage of T_{ii} being triangular matrices.

Once the blocks G_{ii} have been computed, we can successively use (4.22) to approximate the remaining blocks of G. Note that for each $i < j$, the identity (4.22) is a Sylvester equation of the form

$$XM - NX = P, \tag{4.23}$$

where M, N and P are known square matrices and X has to be determined. Equation (4.23) has a unique solution if and only if $\sigma(M) \cap \sigma(N) = \emptyset$. Hence, the block-Parlett method requires the solution of several Sylvester equations with a unique solution.

Recall that $\sigma(T_{ii}) \cap \sigma(T_{jj}) = \emptyset,\ i \neq j$, is assumed to be valid. To avoid the ill conditioning of the Sylvester equations arising in the Parlett recurrence, the eigenvalues of the blocks T_{ii} and T_{jj}, $i \neq j$, need to be well separated in the following sense: there exists $\delta > 0$ (e.g., $\delta = 0.1$), such that

$$\min \{|\lambda - \mu| :\ \lambda, \mu \in \sigma(T_{ii}),\ \lambda \neq \mu\} > \delta \tag{4.24}$$

and, for every eigenvalue λ of a block T_{ii} with dimension bigger than 1, there exists $\mu \in \sigma(T_{ii})$ such that $|\lambda - \mu| \leq \delta$.

An algorithm for computing a Schur decomposition with "well separated" blocks was proposed in [13]. It is available in [25].

Now, if $\Gamma(T_{ii})$ is computed by one of the Algorithms 4.1, 4.2, 4.3 and 4.4, a framework combining those algorithms with the Schur–Parlett technique can be given as follows.

Algorithm 4.5 This algorithm approximates $\Gamma(A)$, where $A \in \mathbb{C}^{n\times n}$ is a non-singular matrix with no negative integers eigenvalues, by Schur–Parlett method combined with the Algorithms 4.1, 4.2, 4.3 or 4.4.

1. Compute a Schur decomposition $A = UTU^*$ (U is unitary and T upper triangular), where the blocks T_{ii} in the diagonal of T are well separated in the sense defined above; The value $\delta = 0.1$ in (4.24) is considered;
2. Approximate $G_{ii} = \Gamma(T_{ii})$ by one of the algorithms: Algorithms 4.1, 4.2, 4.3 or 4.4;
3. Solve the Sylvester equations (4.22), in order to compute all the blocks G_{ij}, with $i < j$;
4. $\Gamma(A) \approx UGU^*$, where $G = \left[G_{ij}\right]$.

4.6 Computational cost

Now we discuss the cost of the Schur–Parlett algorithm when combined with the methods of Lanczos, Spouge, Stirling and reciprocal gamma. As will be seen below, the computation of gamma function is expensive, if compared with other matrix functions like the matrix exponential or the matrix logarithm.

Let us consider the following abbreviations:

`par-lanczos` :	Algorithm 4.5 combined with Algorithm 4.1;
`par-spouge` :	Algorithm 4.5 together with Algorithm 4.2;
`par-stirling` :	Algorithm 4.5 together with Algorithm 4.3;
`par-reciprocal` :	Algorithm 4.5 with Algorithm 4.4.

The cost associated with the Schur–Parlett algorithm without matrix gamma computations (denoted below by C_{sp}) strongly depends on the eigenvalue distribution of A; it is discussed in [13] (see also, [24, Sec. 9.4]). Let us denote by C_{mmt}, C_{invt}, C_{mrhst}, C_{expmt}, C_{logmt}, C_{sinmt}, respectively, the cost of one matrix multiplication between upper triangular matrices ($n^3/3$), the cost of one inversion of a triangular matrices

($n^3/3$), the cost of solving a multiple right-hand linear system involving two upper triangular matrices ($n^3/3$), the cost of computing a matrix exponential of a triangular matrix (see [2,4] and [24, Sec. 10.3]), the cost of computing a matrix logarithm of a triangular matrix (see [3] and [24, Sec. 11.5]), and the cost of one sine of a triangular matrix using `funm` of [13]. Note that those costs pertain to the blocks arising in the diagonal of the triangular matrix of Schur decomposition, which have in general a much smaller size than A. Assuming that the series coefficients are available, we have the following estimations:

$$\begin{array}{ll}
\texttt{par-lanczos}: & C_{sp} + C_{mmt} + 10C_{invt} + C_{expmt} + 2C_{logmt}; \\
\texttt{par-spouge}: & C_{sp} + C_{mmt} + 12C_{invt} + C_{expmt} + 2C_{logmt}; \\
\texttt{par-stirling}: & C_{sp} + \phi(s)C_{mmt} + C_{invt} + C_{mrhst} + C_{expmt} + C_{logmt}; \\
\texttt{par-reciprocal}: & C_{sp} + 14r\, C_{mmt} + C_{invt} + C_{expmt}.
\end{array}$$

Note that for matrices whose eigenvalues all have negative real parts, the amount C_{sinmt} should be added.

We now explain the meaning of $\phi(s)$ appearing in `par-stirling`. The product $\prod_{k=1}^{s}(A + (k-1)I)$ involved in (4.12) can be represented by a polynomial in A by means of the so-called unsigned Stirling numbers of the first kind [22, p. 257]:

$$\prod_{k=1}^{s}(A + (k-1)I) = \sum_{k=0}^{s} \left[{s \atop k} \right] x^k.$$

The number of matrix multiplications involved in the evaluation of that polynomial by the Paterson–Stockmeyer method [24, Sec. 4.2] corresponds to $\phi(s)$.

The coefficient 14 in `par-reciprocal` is the number of matrix multiplications needed for evaluating the polynomials of degree 50 of Algorithm 4.4 by Paterson–Stockmeyer method.

Lanczos and Spouge methods have a similar cost, with Spouge method involving two more matrix inversions. Stirling method involves a few more matrix multiplications, depending on s. The most expensive method is `par-reciprocal`. However, it seems to be very promising because it is rich in matrix–matrix products, which turns it suitable for parallel architectures (note that due to the Schur–Parlett approach, such products are among matrices with small size if compared with the size of A) and it can be adapted to high precision computations by increasing the number of terms in the series or by reducing the parameter μ. It can also be implemented without the Schur–Parlett approach, because it works for matrices having simultaneously eigenvalues with positive and negative real parts. Recall that, without using the Schur–Parlett method, Lanczos, Spouge and Stirling approximations cannot be used for matrices having simultaneously eigenvalues with positive and negative real parts.

5 Numerical experiments

The four algorithms `par-lanczos`, `par-spouge`, `par-stirling` and `par-reciprocal` have been implemented in MATLAB R2018a, which has unit roundoff $u = 2^{-53}$.

In the first experiment, we have tested the algorithms with a set of 20 matrices, with real and non real entries and sizes ranging from $n = 5$ to $n = 15$. Some matrices are randomized, but almost all of them were taken from MATLAB's gallery. The list of matrices is (MATLAB style is used):

1. `3*randn(11)+2*1i*rand(11)`
2. `5*randn(10)`
3. `expm(2*randn(7)+1i*rand(7))`
4. `4*randn(14)+2*1i*rand(14)`
5. `gallery('moler',11,-1)`
6. `gallery('lehmer',11)`
7. `expm(gallery('dramadah',10))`
8. `hilb(5)`
9. `expm(gallery('cauchy',10))`
10. `gallery('condex',10)`
11. `gallery('minij',8)`
12. `gallery('frank',6)`
13. `gallery('gcdmat',11)`
14. `gallery('riemann',11)`
15. `gallery('ris',13)`
16. `gallery('chebspec',10,1)`
17. `gallery('invhess',15)`
18. `gallery('smoke',12)`
19. `gallery('prolate',11,0.3)`
20. `gallery('pei',15,-7)` `+2^(-20)*eye(15).`

Some care has been taken in the choice of those matrices in order to avoid overflow and to guarantee that gamma function is defined. We recall that Γ is not defined if A has any eigenvalue in $\mathbb{Z}_0^-$. It is worth pointing out that Γ grows very fast in the positive real axis.

Figure 2 displays the relative error of algorithms for the above mentioned 20 test matrices, compared with the relative condition number of Γ at A, times the unit roundoff: $\mathrm{cond}_\Gamma(A)u$. To compute those relative errors, we have considered as "exact" matrix gamma function the result obtained by our own implementation of the Lanczos method in MATLAB, using variable precision arithmetic with 250 digits. To compute the relative condition number $\mathrm{cond}_\Gamma(A)$, we have implemented Algorithm 3.17 in [24], where the Fréchet derivative of $L_\Gamma(A, E)$ at A in the direction of E was given by the (1, 2)-block of the matrix gamma function evaluated at $\begin{bmatrix} A & E \\ 0 & A \end{bmatrix}$. Recall that (see [24, (3.16)])

$$\Gamma\left(\begin{bmatrix} A & E \\ 0 & A \end{bmatrix}\right) = \begin{bmatrix} \Gamma(A) & L_\Gamma(A, E) \\ 0 & \Gamma(A) \end{bmatrix},$$

provided that Γ is defined at A.

In Fig. 2, by comparison of the relative errors with the solid line corresponding to $\mathrm{cond}_\Gamma(A)u$, we observe that algorithms `par-lanczos`, `par-stirling`, and `par-reciprocal` have a similar performance, evidencing a better stability than `par-spouge`. The twentieth matrix has been chosen to illustrate the situation when A has some eigenvalues close to $\mathbb{Z}_0^-$. The algorithms `par-reciprocal` and `par-stirling` can be viewed as complementary. While `par-reciprocal` is more suitable for matrices whose eigenvalues all have small magnitude, `par-stirling` works better for matrices with eigenvalues having large magnitude.

In the second experiment (see Fig. 3), we consider the symmetric positive definite matrix `gallery('moler',12,a)` for ten values of a: $a = 0.1, 0.2, \ldots, 1$. As a grows slowly in the positive real axis, the spectral radius grows very fast. It ranges

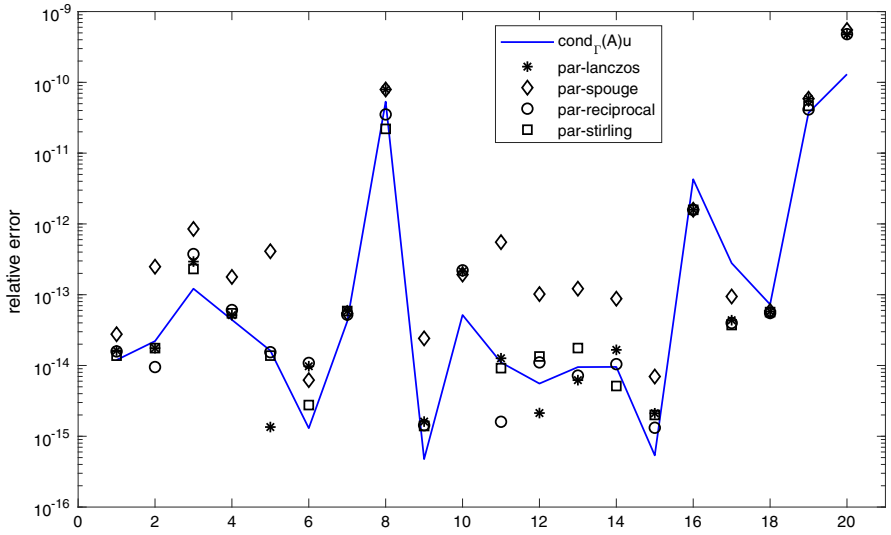

Fig. 2 Relative errors of the four proposed methods for 20 matrices together with the relative condition number of $\Gamma(A)$ times the unit roundoff of MATLAB

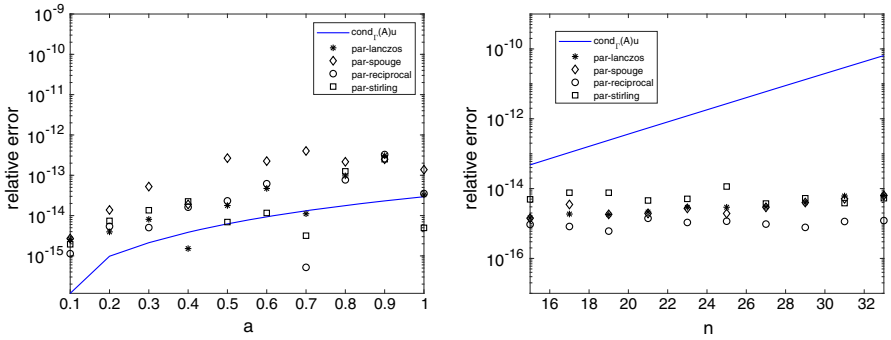

Fig. 3 Left: relative errors of the four proposed methods for the matrices obtained from `gallery('moler',12,a)`, by varying a from 0.1 to 1. Right: relative errors of the four methods for the matrices obtained from `gallery('kahan',n)` with size n increasing as $n = 15, 17, \ldots, 33$

from $\rho(A) = 2.5519$ for $a = 0.1$ to $\rho(A) = 63.4091$ for $a = 1$. We have found the results for smaller values of a to be fair when compared with the largest ones, where the gamma function attains very large values. Apparently, this is due to the eigenvalues being highly clustered for the lowers values of a. For instance, if $a = 0.1$, and assuming that a value of block separation of $\delta = 0.1$ has been used in Algorithm 4.5, the triangular matrix of the Schur decomposition of A has just three blocks: two blocks 1×1 and one block of order 10. In many experiments we have carried out involving Algorithms 4.1, 4.2, 4.3 and 4.4 (not reported here), we have observed that the performance of those algorithms, when running without the Schur–Parlett approach, deteriorates as the size of the matrices increased.

The third experiment (Fig. 3) illustrates the behaviour of the algorithms for the following ten matrices `gallery('kahan',n)` with size n increasing as $n = 15, 17, \ldots, 33$. The results are very satisfactory. Now the eigenvalues are close, but "well-separated" with $\delta = 0.1$ in (4.24), and the spectral radius is $\rho(A) = 1$. Not so good results are expected if increasing the size of A corresponds a significant growth in the magnitude of the eigenvalues.

6 Conclusions

In this work we have provided a thorough investigation on the numerical computation of $\Gamma(A)$. Four methods have been analysed: the Lanczos method, Spouge method, Stirling method and a method based on a Taylor expansion of the reciprocal gamma function combined with the Gauss multiplication formula. All of them have been implemented together with the Schur–Parllet method and tested with several matrices. The deviation of the relative error from $\mathrm{cond}_\Gamma(A)u$ is bigger in Spouge method, which led us to conclude that Lanczos, Stirling and reciprocal gamma approximations are preferable. New bounds for the norm of the matrix gamma function and its perturbations, and for the truncation errors arising in the approximation methods have been proposed as well.

Likewise the scalar case, the computation of matrix gamma function is a very challenging issue. We believe our contributions are a starting point for the effective computation of this important matrix function and could motivate researchers to continue extending computational methods for other related matrix functions, such as the incomplete gamma and the random gamma function [9] to the matrix scenario.

Acknowledgements We would like to thank the anonymous reviewers for their helpful suggestions and comments. The work of the first author was supported by ISR-University of Coimbra (project UID/EEA/00048/2013) funded by "Fundação para a Ciência e a Tecnologia" (FCT). The work of the corresponding author is supported by Robat Karim branch, Islamic Azad University, Tehran, Iran.

References

1. Abramowitz, M., Stegun, I.: Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables. Dover, New York (1970)
2. 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)
3. Al-Mohy, A.H., Higham, N.J.: Improved inverse scaling and squaring algorithms for the matrix logarithm. SIAM J. Sci. Comput. **34**(4), C153–C169 (2012)
4. Al-Mohy, A.H., Higham, N.J., Relton, S.D.: Computing the Fréchet derivative of the matrix logarithm and estimating the condition number. SIAM J. Sci. Comput. **35**(4), C394–C410 (2013)
5. Askey, R., Roy, R.: Gamma function. In: Olver, F., Lozier, D., Boisvert, R., Clark, C. (eds.) NIST Handbook of Mathematical Functions. Cambridge University Press, Cambridge (2010)
6. Barradas, I., Cohen, J.E.: Iterated exponentiation, matrix–matrix exponentiation, and entropy. J. Math. Anal. Appl. **183**, 76–88 (1994)
7. Borwein, J.M., Corless, R.M.: Gamma and factorial in the Monthly. Am. Math. Mon. **125**(5), 400–424 (2018)
8. Bourguet, L.: Sur les intégrales Eulériennes et quelques autres fonctions uniformes. Acta Mathematica **2**, 261–295 (1883)

9. Braumann, C.A., Cortés, J.C., Jódar, L., Villafuerte, L.: On the random gamma function: theory and computing. J. Comput. Appl. Math. **335**, 142–155 (2018)
10. Cardoso, J.R., Sadeghi, A.: On the conditioning of the matrix–matrix exponentiation. Numer. Algorithms **79**(2), 457–477 (2018)
11. Char, B.: On Stieltjes' continued fraction for the gamma function. Math. Comput. **34**(150), 547–551 (1980)
12. Cortés, J.C., Jódar, L., Solís, F.J., Ku-Carrillo, R.: Infinite matrix products and the representation of the matrix gamma function. Abstract and Applied Analysis, vol. 2015, Article ID 564287. https://doi.org/10.1155/2015/564287 (2015)
13. Davies, P.A., Higham, N.J.: A Schur–Parlett algorithm for computing matrix functions. SIAM J. Matrix Anal. Appl. **25**(2), 464–485 (2003)
14. Davis, P.J.: Leonhard Euler's integral: a historical profile of the gamma function. Am. Math. Mon. **66**, 849–869 (1959)
15. Edwards, H.M.: Riemanns Zeta Function. Academic Press, Cambridge (1974)
16. Fekih-Ahmed, L.: On the power series expansion of the reciprocal gamma function, HAL archives. https://hal.archives-ouvertes.fr/hal-01029331v1 (2014). Accessed 16 Mar 2018
17. Gautschi, W.: A computational procedure for incomplete gamma functions. ACM Trans. Math. Softw. **5**(4), 466–481 (1979)
18. Gautschi, W.: The incomplete gamma function since Tricomi. Atti Convegni Lincei **147**, 203–237 (1998)
19. Gautschi, W.: A note on the recursive calculation of incomplete gamma functions. ACM Trans. Math. Softw. **25**(1), 101–107 (1999)
20. Godfrey, P.: Lanczos implementation of the gamma function. http://www.numericana.com/answer/info/godfrey.htm (See also http://my.fit.edu/~gabdo/gamma.txt). Accessed 16 Mar 2018
21. Golub, G.H., Van Loan, C.F.: Matrix Computations, 4th edn. Johns Hopkins University Press, Baltimore (2013)
22. Graham, R., Knuth, D., Patashnik, O.: Concrete Mathematics, 2nd edn. Addison-Wesley, Boston (1994)
23. Hale, N., Higham, N.J., Trefethen, L.: Computing A^{α}, $\log(A)$, and related matrix functions by contour integrals. SIAM J. Numer. Anal. **46**, 2505–2523 (2008)
24. Higham, N.J.: Functions of Matrices: Theory and Computation. Society for Industrial and Applied Mathematics, Philadelphia (2008)
25. Higham, N.J.: The Matrix Function Toolbox. http://www.maths.manchester.ac.uk/~higham/mftoolbox/. Accessed 8 Feb 2018
26. Horn, R.A., Johnson, C.R.: Topics in Matrix Analysis, Paperback Edition. Cambridge University Press, Cambridge (1994)
27. Horn, R.A., Johnson, C.R.: Matrix Analysis, 2nd edn. Cambridge University Press, Cambridge (2013)
28. Jódar, L., Cortés, J.C.: On the hypergeometric matrix function. J. Comput. Appl. Math. **99**, 205–217 (1998)
29. Jódar, L., Cortés, J.C.: Some properties of gamma and beta functions. Appl. Math. Lett. **11**(1), 89–93 (1998)
30. Lanczos, C.: A precision approximation of the gamma function. J. Soc. Ind. Appl. Math. Ser. B Numer. Anal. **1**, 86–96 (1964)
31. Luke, Y.: The Special Functions and Their Approximations, vol. 1. Academic Press, New York (1969)
32. Mathias, R.: Approximation of matrix-valued functions. SIAM J. Matrix Anal. Appl. **14**, 1061–1063 (1993)
33. Parlett, B.N.: A recurrence among the elements of functions of triangular matrices. Linear Algebra Appl. **14**, 117–121 (1976)
34. Paterson, M.S., Stockmeyer, L.J.: On the number of nonscalar multiplications necessary to evaluate polynomials. SIAM J. Comput. **2**(1), 60–66 (1973)
35. Pugh, G.R.: An analysis of the Lanczos gamma approximation. Ph.D. thesis, University of British Columbia (2004)
36. Sastre, J., Jódar, L.: Asymptotics of the modified Bessel and incomplete gamma matrix functions. Appl. Math. Lett. **16**(6), 815–820 (2003)
37. Schmelzer, T., Trefethen, L.N.: Computing the gamma function using contour integrals and rational approximations. SIAM J. Numer. Anal. **45**, 558–571 (2007)
38. Smith, D.: Algorithm 814: fortran 90 software for floating-point multiple precision arithmetic, gamma and related functions. ACM Trans. Math. Softw. **27**(4), 377–387 (2001)

39. Spira, R.: Calculation of the gamma function by Stirling's formula. Math. Comput. **25**(114), 317–322 (1971)
40. Spouge, J.: Computation of the gamma, digamma, and trigamma functions. SIAM J. Numer. Anal. **31**(3), 931–944 (1994)
41. Temme, N.: Special Functions: An Introduction to the Classical Functions of Mathematical Physics. Wiley, New York (1996)
42. Trefethen, L.N., Weideman, J., Schmelzer, T.: Talbot quadratures and rational approximations. BIT Numer. Math. **46**, 653–670 (2006)
43. Van Loan, C.: The sensitivity of the matrix exponential. SIAM J. Numer. Anal. **14**(6), 971–981 (1977)
44. Winitzki, S.: Computing the incomplete gamma function to arbitrary precision. In: Kumar, V., et al. (eds.) Lectures Notes on Computer Science, vol. 2667, pp. 790–798 (2003)
45. Wrench, J.W.: Concerning two series for the gamma function. Math. Comput. **22**, 617–626 (1968)
46. Wrench, J.W.: Erratum: concerning two series for the gamma function. Math. Comput. **27**, 681–682 (1973)