NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Memorandum 33-686

A Comparison of Computational Methods and Algorithms for the Complex Gamma Function

Edward W. Ng

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

May 1, 1974

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Memorandum 33-686

A Comparison of Computational Methods and Algorithms for the Complex Gamma Function

Edward W. Ng

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

May 1, 1974

Prepared Under Contract No. NAS 7-100
National Aeronautics and Space Administration

PREFACE

The work described in this report was performed by the Data Systems Division of the Jet Propulsion Laboratory.

CONTENTS

PRECEDING PAGE BLANK NOT FILMED

CONTENTS (contd)

TABLES

FIGURE

ABSTRACT

This paper presents a survey and comparison of some computational methods and algorithms for gamma and log-gamma functions of complex arguments. All these methods and algorithms are reported recently in the open literature and include Chebyshev approximations, Padé expansion and Stirling's asymptotic series. The comparison leads to the conclusion that Algorithm 421 published in the Communications of ACM by H. Kuki is the best program either for individual application or for the inclusion in subroutine libraries.

I. Introduction

In the past two years there appear in the open literature a number of papers on the computation of gamma or log-gamma functions of complex arguments: $\Gamma(z)$ and $\ell n\Gamma(z)$. In particular, there are two published algorithms, A404 and A421 [3 and 2], there is Luke's analysis published in a SIAM Journal [4], there are Spira's study [5] and Cody's approximations [1]*, both reported in the Mathematics of Computation. The last approximations apply only to special cases where the argument values lie on straight lines parallel to the imaginary axis. In this article we attempt to compare and discuss the methods or algorithms given in these papers. We hope that such investigation may provide several useful functions. First, it surveys the recent activities in this area of computational mathematics. Second, it provides information on what good methods to use in computing this function and thereby eliminating the poor ones. Third, it helps to bring out a high-quality algorithm to be recommended either for individual use or for the inclusion in program libraries. Last, but not least, we hope this study may contribute some ideas to the methods and processes of evaluation of mathematical software.

II. Computational Methods

For the five papers mentioned we find three distinct methods proposed, viz., Chebyshev rational approximation, asymptotic expansion and Padé approximations. All these methods are applicable to arguments confined in some segments of the complex plane. An algorithm using any of these methods is therefore dependent on some analytic continuation to cover all four

*For conciseness, papers of multiple authorship will be referred to by the first author. The references will give full citations.

quadrants of the complex plane. For the function in question, we have simple formulae for such purpose, but even these simple formulae have to be implemented with caution in order to increase efficiency and enhance accuracy, as we shall discuss subsequently. We find it appropriate here to list these formulae for later references in this article. They are, respectively, well-known formulae for recursion, reflection, conjugation and duplication of arguments (Cf. [7]):

$$\Gamma(z+n) = \Gamma(z) \prod_{k=0}^{n-1} (z+k) \tag{1}$$

$$\Gamma(z)\Gamma(1-z) = \pi \operatorname{cosec} \pi z, \tag{2}$$

$$\Gamma(\overline{z}) = \overline{\Gamma(z)}, \tag{3}$$

$$\Gamma(2z) = 2^{2z-1}\Gamma(z)\Gamma(z+\tfrac{1}{2})/\sqrt{\pi} \tag{4}$$

1. Chebyshev Rational Approximations

First we turn to Cody's approximations, which are applicable to special cases of this function and should not be directly compared with the other investigations considered here. In his investigation, Cody's main concern is to provide minimax approximation for the Colomb phase shift, which occurs in the asymptotic behavior of the Coulomb wave functions, and is defined as

$$\sigma_L(\eta) = \mathrm{Im}\{\ell n\Gamma(L+1+i\eta)\} \tag{5}$$

where L is an integer. The zeroth order phase shift is represented by 3 sets of rational approximations of the following form:

$$\sigma_0(\eta) \simeq \eta(\eta^2-\eta_0^2)R_1(\ell m;\eta^2), \qquad 0 \leq \eta \leq 2.0, \tag{6}$$

$$\simeq \eta R_2(\ell m;\eta^2), \qquad 2.0 \leq \eta \leq 4.0, \tag{7}$$

$$\simeq \frac{1}{2}\ \arctan(\eta) + \eta[\ \frac{1}{2}\ \ell n(1+\eta^2) + R_3(\ell m;1/\eta^2)] \quad 4.0 \leq \eta, \tag{8}$$

where η_0 is the positive nontrivial zero of $\sigma_0(\eta)$, and the R's are rational functions of degree ℓ in the numerator and m in the denominator. We note here that the asymptotic approximation as expressed in equation (8) can be substantially improved for efficiency. To this end we may use

$$\sigma_0 = \sigma_{-1} + \pi/2,$$

from

$$\ell n\Gamma(1+i\eta) = (\ell n\eta + i\pi/2) + \ell n\Gamma(i\eta),$$

and an approximation of the form

$$\sigma_{-1} = \mathrm{Im}\ \ell n\Gamma(i\eta) \approx \eta\ell n\eta - \eta - \pi/4 - \hat{R}_3(\ell m; 1/\eta^2) \tag{8a}$$

In other words, the arctangent function may be alleviated in equation (8). Since it constitutes almost 1/2 of the cost in that equation, this amounts to a substantial savings. The difference in efficiency between R_3 and $\hat{R}_3$ should be insignificant because these two rational functions are approximating respectively the asymptotic behaviors $[1/(\eta^2+1)]^{-n}$ and $[1/\eta^2]^{-n}$, which are almost identical for large η.

Cody further suggests that higher order phase shifts may be computed from the identity

$$\sigma_L(\eta) = \sigma_0(\eta) + \sum_{j=1}^{L} \arctan(\eta/j), \tag{9}$$

which comes from the recursion formula (1).

In the context of our general discussion in this paper, the main application of Cody's approximations will be for the computation of the gamma function of pure imaginary arguments. Let

$$\ell n\Gamma(iy) = U + iV. \tag{10}$$

We then have (also, cf.[7]),

$$U = \frac{1}{2}\,\ell n\pi - \frac{1}{2}\,\ell n(y\sinh\pi y), \tag{11}$$

and

$$V = \sigma_0(y) - \pi/2. \tag{12}$$

Cody's approximations for $\sigma_0(y)$ are very efficient. For example, for a relative truncation error less than 10^{-8}, one needs only 4-4 rational functions for $y \leq 4.0$ and 2-2 rational functions for $y > 4.0$.

2. Stirling's Approximation

We now turn to the use of Stirling's approximation - the subject of the articles by Kuki, Lucas and Spira. Of the two well-known versions of this approximation, all the authors mentioned have chosen the more efficient form, viz., that for log-gamma

$$\ell n\Gamma(z) = (z - \frac{1}{2})\ell nz - z + \frac{1}{2}\,\ell n(2\pi) + \sum_{k=1}^{N} B_{2k}/[2k(2k-1)z^{2k-1}] + T_N(z) \tag{13}$$

where T_N is the truncation error term and B_{2k} are Bernouli numbers. There have been reported in the literature several bounds and estimates for the quantity T_N. Spira has summarized some of these. We give here an updated review of these error analyses and applications.

(i) Whittaker and Watson [8]

$$|T_N| \le \frac{|B_{2N+2}|\, K(z)}{(2N+1)(2N+2)\,|z|^{2N+1}} \tag{14}$$

where

$$K(z) = \text{upper bound } |z^2/(u^2+z^2)|, \quad u \ge 0$$

(ii) Nielsen [6]

$$|T_N| < \frac{|B_{2N+2}|}{(2N+1)(2N+2)\,|z|^{2N+1}[\cos(\frac{1}{2}\arg z)]^{2N+2}}, \quad |\arg z| < \pi \tag{15}$$

(iii) Spira [5]

$$\begin{aligned} |T_N| &\le 2|B_{2N}/(2N-1)|\cdot|\operatorname{Im} z|^{1-2N}, \quad \operatorname{Re} z < 0,\ \operatorname{Im} z \ne 0, \\ |T_N| &\le |B_{2N}/(2N-1)|\cdot|z|^{1-2N}, \quad \operatorname{Re} z \ge 0. \end{aligned} \tag{16}$$

(iv) Lucas and Terrill [3]

$$|\operatorname{Re} T_N| < |\operatorname{Re} S_{N+1}|, \quad |\operatorname{Im} T_N| < |\operatorname{Im} S_{N+1}|, \quad |\arg z| \le \pi/4 \tag{17}$$

where S_{N+1} is the (N+1)st term in the asymptotic series (13). This is actually derived from the bound (14).

(v) Kuki [2]

$$|T_N| \le \epsilon, \ \mathrm{Re}\, z \ge \max\{a(\epsilon), \min[b(\epsilon), c(\epsilon) - |\mathrm{Im}\, z|]\} > 0, \tag{18}$$

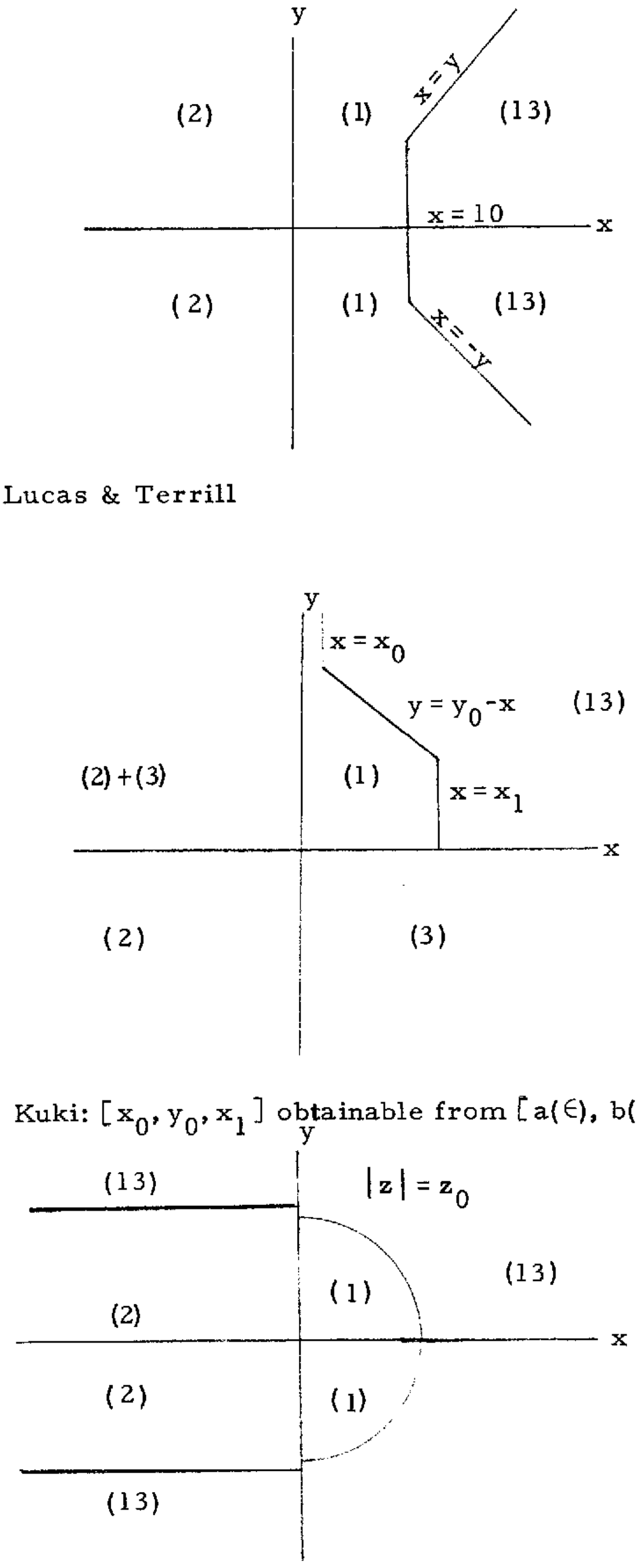
where a, b and c are constants dependent on a given ϵ and are derived from the condition (14).

The choice of a proper truncation error control is extremely important because such choice determines for a desired accuracy the region of applicability of Stirling's approximation which in turn affects the efficiency and final accuracy rendered by an algorithm. Of the five types of error bounds described above, we believe that in general Nielsen's formula is too expensive in the requirement of computing a cosine and an arctangent and Lucas' formula too inefficient in its exclusion of one-half of the complex plane for applicability. Kuki's truncation error control is realistic and most efficient, but suffers from serious inflexibility due to the requirement that the boundary curve need be derived for each different precision desired. All in all, we believe that Spira's error bound is a reasonable compromise choice for a general algorithm for the complex gamma function. It is simple to use and is fairly efficient in its permission of the applicability of Stirling's approximation in a large segment of the complex plane, thereby minimizing the use of recursion.

With the proper boundary curve provided by a particular truncation error control, Stirling's approximation must be used in combination with some or all of the analytic continuation properties given in equations (1) - (4).

The following figures show the implementation (or the proposed implementation in Spira's case) of the various authors being reviewed. For example, in Lucas' implementation, given an argument z on the left half of the complex plane formula (2) is used to reflect the computation to the right half. Then, if necessary, formula (1) is used to raise the argument such that (13) may be applied. Similar remarks apply to the other two cases.

Figure 1. Implementation of Stirling's approximation by different authors. Numbers in figures refer to formulae in this text. $x = \text{Re}\, z$, $y = \text{Im}\, z$.



(i) Lucas & Terrill

(ii) Kuki: $[x_0, y_0, x_1]$ obtainable from $[a(\in), b(\in), c(\in)]$

(iii) Spira

3. Padé Approximations

We next turn to the Padé approximation used by Luke [4]. These approximations are derived from those for two incomplete gamma functions, viz., $\Gamma(z) = \gamma(z,\alpha) + \Gamma(z,\alpha)$, where α may be considered a free parameter in the context of the present application. The approximant takes the following form.

$$\Gamma(z+1) = \frac{C_n(z)}{D_n(z)} + \frac{G_m(z)}{H_m(z)} + L_n(z) + U_m(z),\ R(z) \geq 0 \tag{19}$$

where C_n, D_n, G_m and H_m are constituted of hypergeometric functions of the type ${}_pF_q(a_1 \cdots a_p, b_1 \cdots b_q; t)$, and L_n and U_m are truncation errors of the two Padé rational functions. The approximants may be computed by recurrence relations. Both C_n and D_n satisfy the following recurrence,

$$(2n+z)\, C_{n+1} = [\,(2n+z)(2n+z+2) - \alpha z\,](2n+z+1)\, C_n + n(n+z)(2n+z+2)\,\alpha^2 C_{n-1}, \tag{20}$$

$$C_0 = \alpha^z e^{-\alpha},\ C_1 = \alpha^z e^{-\alpha}[\,\alpha+(z+1)(z+2)\,];$$

$$D_0 = 1, \qquad D_1 = (z+1)(z+2-\alpha)$$

Similarly both G_m and H_m satisfy the following,

$$G_{m+1} = (\alpha+2m+2-z)\, G_m - m(m+1-z)\, G_{m-1} \tag{21}$$

$$G_0 = \alpha^z e^{-\alpha} z,\ G_1 = \alpha^z e^{-\alpha} z(\alpha+1);$$

$$H_0 = \alpha, \qquad H_1 = \alpha(2+\alpha-z).$$

Luke has further provided very realistic estimates on the truncation error terms L_n and U_m. For our purpose it suffices to record here selected numerical values for the sake of comparison.

TABLE 1

Numerical Estimates of Relative Truncation Errors in Luke's Padé Approximants

$|L_n(z, \alpha)/\Gamma(z+1)|, \qquad z = x+i\,y, \qquad \alpha = 8$

n	x/y	0	2	4	6	8	10
11	0.5	.23(-07)	.21(-07)	.19(-07)	.32(-07)	.54(-07)	.25(-07)
16	0.5	.61(-16)	.56(-16)	.55(-16)	.66(-16)	.14(-15)	.18(-15)
10	2.5	.13(-07)	.12(-07)	.12(-07)	.18(-07)	.28(-07)	.13(-07)
15	2.5	.41(-16)	.40(-16)	.39(-16)	.45(-16)	.89(-16)	.11(-15)
9	4.5	.88(-08)	.86(-08)	.84(-08)	.12(-07)	.17(-07)	.83(-08)
14	4.5	.31(-16)	.31(-16)	.30(-16)	.35(-16)	.64(-16)	.79(-16)

$|U_m(z, \alpha)/\Gamma(z+1)|, \qquad z = x+i\,y, \qquad \alpha = 8$

n	x/y	0	2	4	6	8	10
11	0.5	.98(-17)	.20(-14)	.52(-12)	.84(-10)	.89(-08)	.65(-06)
16	0.5	.63(-20)	.14(-17)	.41(-15)	.82(-13)	.12(-10)	.11(-08)
20	0.5	.37(-22)	.82(-20)	.26(-17)	.59(-15)	.96(-13)	.11(-10)
11	2.5	.57(-16)	.11(-13)	.26(-11)	.35(-09)	.29(-07)	.16(-05)
17	2.5	.69(-20)	.15(-17)	.42(-15)	.81(-13)	.10(-10)	.93(-09)
20	2.5	.14(-21)	.30(-19)	.93(-17)	.20(-14)	.29(-12)	.30(-10)
11	4.5	.59(-15)	.11(-12)	.22(-10)	.23(-08)	.14(-06)	.59(-05)
17	4.5	.44(-19)	.93(-17)	.25(-14)	.42(-12)	.47(-10)	.35(-08)
20	4.5	.75(-21)	.16(-18)	.47(-16)	.92(-14)	.12(-11)	.11(-09)

III. Discussions and Comparison of Methods

In the consideration of various competing methods as candidates for algorithmic implementation, we believe the following points must be examined.

1. Truncation Errors in the Approximations

(a) The choice of relative versus absolute error criterion must be determined. Kuki and Spira both use an absolute criterion. This is quite consistent in the particular application in both cases, where the log-gamma function is being approximated in a region where the modulus of the function values typically range in [10, 50]. In such region the absolute error serves as an upper bound to the relative error. Since the approximation can be made quite efficient by the proper choice of region (e.g., a 7th degree polynomial for a 10^{-16} accuracy), the absolute criterion is satisfied at a low cost. On the other hand, if the gamma function is to be approximated, then relative criterion must be imposed, because the function values may be arbitrarily large. We find Luke's usage of this latter criterion also consistent with his application. Lucas' usage of relative criterion is somewhat obscured by his error control on the components of the function values, which we discuss next.

(b) For complex-valued functions, the choice between component and modulus accuracy is dependent on the particular application in question. Therefore for a general-purpose algorithm only some rather general arguments may be advanced to favor one over the other. In the present context, Kuki has proposed strong argument in favor of modulus accuracy: "Since an analytic function maps the complex plane locally formally, it maps a circular blur about the correct answer. This means that the concept of vector (or modulus) error is the natural one to use for compounding errors through successive computational steps." Probably for the same reason Luke and Spira have also chosen the modulus accuracy criterion. There is an additional reason to favor the choice of modulus accuracy in the applications of Kuki and Spira. In these cases where absolute accuracy are desired, the modulus of the error is an upper bound to the components.

(c) There is also the question of fixed versus variable precision control. In the former case a fixed order approximation is predetermined based on an error bound for the worst case. An example is Kuki's algorithm where he retains seven terms throughout the region of application of Stirling's approximation. The advantages include the elimination of extra storage and of testing, whereas the disadvantage lies in the obvious expense of an attempt to satisfy the most pessimistic error bound. Variable precision control is imposed through the computation of a sequence of approximants until the difference between two consecutive ones is less than some desired tolerance. This type of control, as found in the investigations of Luke and Lucas, seems to utilize an error bound or estimate optimally, but an expense is paid in terms of the testing needed. We believe that in general, if the order of approximation for a desired accuracy is a rapidly varying function of the argument, this type of control should be used, because the most pessimistic bound may incur high cost in the use of the approximation. Such consideration applies to Luke's method in which the use of variable precision control is, in our opinion, well justified. On the other hand, Lucas' usage of such control is not as mandatory. In discussing the merit of his method relative to Luke's, Kuki concluded that "this comparison presents an example of the high cost of variable precision programming". Whereas we agree with Kuki that Luke's method is expensive, we disagree with his contention that the high cost is due to variable precision control. In a subsequent paragraph we shall discuss the counts of arithmetic operations and shall suggest that the cost of Luke's method comes from the arithmetic requirements intrinsic in his approach.

2. Round-Off and Cancellation Errors

In addition to truncation, there are two other sources of errors in the use of Stirling's approximation. There is the accumulation of round-off errors in the summation of the asymptotic series. This summation may be readily arranged as a polynomial of fairly low degree. For example, in Kuki's case it is a third and seventh degree polynomials for single and double precision, respectively; in Lucas' case, since this approximation is applied in a region further removed from the origin, the required degree is certainly lower than 7. This low-degree polynomial of real coefficients is well-conditioned except near a zero where significance is lost in the evaluation of the polynomial. But since the application of this approximation is limited to large values of $|z|$ (say, $|z|$ in $[10, 30]$), the other terms in the expression (13) are also large in magnitudes. Hence full significance is not needed for the polynomial which is added to the other terms as a correction. All in all, there is no serious build-up of round-off errors in the evaluation of equation (13) and our observation here, viewed from an entirely different perspective, is in agreement with Kuki's conclusion. A second source of error, through cancellation, is much more serious. This comes about in the use of the logarithmic version of equation (1), in order to be consistent with equation (13) and to avoid premature overflow. In evaluating this expression, $\ell n\Gamma(z)$ is computed by subtracting a sum from $\ell n\Gamma(z+n)$, which is substantially larger than $\ell n\Gamma(z)$, thereby contributing the cancellation error. To deal with this problem, Kuki employs an excellent maneuver by combining equations (1) and (13) and analytically eliminate such cancellation. On the other hand, Lucas has not attended to this problem which is certainly an important cause of poor relative accuracy reported by him and confirmed by the present author.

In Luke's method, the main computational scheme involves four recurrences (20) and (21). Therefore the major concern about round-off errors centers on the numerical stability of applying these recurrences in a forward direction. Luke provides some arguments to contend that such procedure is stable. In particular his contention is based on the fact that both $E_n(z,\alpha)/C_n(z,\alpha)$ and $E_n(z,\alpha)/D_n(z,\alpha)$ go to zero rapidly as $n \rightarrow \infty$ where $E_n(z,\alpha) = D_n(z,\alpha)/L_n(z,\alpha)$, and similarly for G_m and H_m. We believe more detailed analysis will be needed on this point if Luke's method is to be applied. Wimp's recent theoretical work [9] will be helpful in this regard.

3. Multiple Values of Log-Gamma

In the course of computing log-gamma, any use of the complex logarithm must be carried out with care to avoid the extraneous addition or subtraction of a multiple of 2π to the imaginary part of the result. In Lucas' algorithm, log-gamma is only computed as intermediate result to obtain gamma. Therefore any extraneous quantity $2k\pi i$ will not affect the final answer because $\exp(2k\pi i) = 1$. In Luke's method, the process is exactly reversed. Here one is concerned with the proper way of taking the logarithm of a complex result. Analysis by Luke leads to the following result.

Let

$$\Gamma(z+1) = K + iL = r\,e^{i(\varphi+2\pi k)}, \tag{22}$$

$$\text{where} \quad \varphi = \tan^{-1}(L/K), \qquad 0 \le \varphi \le 2\pi,$$
$$k = 0,\ 1,\ 2,\ \ldots,$$

then k may be determined by the following.

$$|k-\Delta| < \frac{1}{720\pi},\ 2\pi\Delta = \beta\sin\xi(\ell n\beta-1)+\xi(\beta\cos\xi - 1/2)-\sin(\xi)/(12\beta)-\varphi \tag{23}$$

$$\text{where} \quad z+1 = \beta e^{i\xi}.$$

In Kuki's approach, precaution is first taken to insure that the function computed is continuous in the first quadrant of the complex plane. Then the term $\log\sin(\pi z)$ as used in the reflection formula, is analyzed for analytic continuity and reduction of round-off errors. In the end the reflection formula is implemented in the following form

$$\ell n\Gamma(z) = G(z) - \ell n H(z) - \ell n\Gamma(1-z) \tag{24}$$

where

$$G(z) = \ell n(2\pi) + \pi y + i\pi(1/2 - x),$$

$$H(z) = (1-e^{2\pi y}) + e^{2\pi y}[2\sin^2(\pi x) + i\sin(2\pi x)].$$

Now $\ell n H(z)$ can be replaced by its principal value for the following reason. Since $|e^{-2\pi zi}| = e^{2\pi y} \le 1$ if $y \le 0$, and $H(z) = 0$ only if $y = 0$ and x is an integer, it means that as a parameter ζ varies continuously in the lower half of the complex plane, $H(\zeta)$ follows a path entirely contained in the circle of radius 1 with the center at $1 + oi$. Therefore the principal value of $\log H(\zeta)$ varies continuously along the path.

All in all, we believe that this problem has been treated by all the authors carefully and accurately and where it is ignored, it is done so with justification. However, equation (23) is rather expensive to implement and in our opinion it is desirable to simplify this proposed procedure if possible.

4. Counts of Arithmetic Operations and External Function Calls

The number of arithmetic operations for a particular method can be estimated a priori. Such estimate provides an indication about the relative expense of a method, though the corresponding algorithm may still vary substantially depending on how it is implemented. For our purpose here we shall express all operations in terms of two basic units, viz., A for real floating point addition and M for real floating multiplication. For example, each complex addition will be assigned 2A, complex multiplication be given 4M+2A division be approximated by 10M+2A, and real division be approximated by 3M. In order to establish some common ground for comparison, we shall consider two target precisions, say, short and long: 10^{-7} and 10^{-16}, respectively. We shall consider the estimates of operations needed for each method to attain a truncation error less than these tolerances. Our attention is first drawn to the right half of the complex plane. Except for Kuki's all methods are symmetric with respect to the real axis, as far as number of arithmetic operations are concerned. In Kuki's case, since the conjugation relation is used for arguments in the fourth quadrant, it is required only an extra addition, which is of course negligible in our kind of estimate. Therefore we can further restrict our operation counts to arguments in the first quadrant. Table 3 gives the results of such operation counts based on the truncation error bounds or estimates provided by each author. We emphasize that these counts represent gross estimates only and that numerous machine-dependent factors can affect the performance of a particular method. For example, two such factors are whether complex arithmetic operations are coded on line in a compiler or floating division should be approximated as 3M. Nevertheless, even with such uncertainties, our estimates do provide an indication of the relative expense of each method. In addition to arithmetic operations, attention should be focused on the number and types of external functions used. Table 2 provides (in abbreviated notations) a summary of functions required: complex logarithm (CLOG), complex exponential (CEXP), real logarithm (LOG), real exponential (EXP), real inverse tangent (ATAN), and real hyperbolic sine (SINH).

TABLE 2

Elementary Function Calls Required on the Application of Various Methods For $\mathrm{Re}\, z \geq 0$

	GAMMA	LOG-GAMMA
Chebyshev Rational	CEXP + ATAN + 2LOG + SINH $\approx 2\frac{1}{6}L$	ATAN + 2LOG + SINH $\approx 1\frac{1}{6}L$
Stirling Asymptotic	CEXP + CLOG $\approx$ 2L	CLOG $\approx$ L
Pade Approximation	CEXP $\approx$ L	CEXP + CLOG $\approx$ 2L

In order to reduce our comparison to a common basis for discussion, we shall consider a unit L which is defined to be the amount of computation required for the complex logarithm. In a typical complex logarithm about two-thirds of the work is due to the computation of an inverse tangent and a real logarithm. Therefore the latter combination is given (2/3)L in Table 2. Moreover, the computational effort for the complex exponential may be approximated by L. Thus in Table 2 we also summarize the units of L required in each type of methods considered. However, the units of L given result from an optimal counting for each method, which in actual implementation may substantially exceed the count given. For example, where the logarithmic version of eq. (1) is used, at least one additional call to the complex logarithm is required, and where eq. (1) is not used optimally, many such calls are invoked (as in Lucas' case). In Table 3 we record for each author, together with the total number of arithmetic operations, all the additional units of L, either due to the need of a functional equation or due to a non-optimal arrangement. Some comments apply to the individual columns appearing in Table 3. For small arguments in Cody's method, no function calls are required and therefore the proper fraction of L is subtracted. Operation counts for Stirling's approximation are obtained in a straightforward manner. A cursory examination of eq. (13) shows the minimal requirement of (N+1) complex multiplications, (N+2) complex and 2 real additions. The counting for Spira's method is based on a variable-precision error control which favors the method somewhat because logical operations are not included in the total. For similar reason our counts are slightly biased against Kuki's implementation because some arithmetic operations are incorporated for the sake of reducing round-off errors which are not attended to in the other implementations. In the case of Luke's approximation we have computed the truncation error estimates for the three rays $\arg z = \epsilon$, $\pi/4$ and $\pi/2$, where ϵ is a small number of the order 10^{-2}. We use ϵ

because the approximation becomes exact for certain argument values on the real axis. Operation counts for Luke's approximation have been carried out by Kuki for a typical set of arguments. Our counts differ from Kuki's in yielding more optimistic results. The major difference comes from the method of estimating complex arithmetic operations. For example, the multiplication of a complex number by a real one is considered equivalent to two real multiplications rather than one full complex multiplication. In some operating systems, the latter may actually be applied. Thus our operation counts for Luke's case result in $Op(C) = 14A + 2CA + 7M + 6CM + 1CD$ and $Op(G) = 5A + 1CA + 2M + 2CM$ for the two recurrences (20) and (21), where CA stands for complex addition, etc. Expressing all operations in units of A and M and applying to the approximation (19) for C, D, G and H, we find that Luke's method requires $2n(34M + 30A) + 2m(10M + 11A)$, where n and m are dependent on the target precisions. It is apparent from Luke's analysis that his method is prohibitively expensive for large $|z|$. In fact he suggested the usage of Stirling's approximation in such cases. The error estimates he provided are restricted to $|z| < 11$. We have computed these estimates beyond $|z| = 15$ and have found that it becomes impractical to apply his method in those cases. In Table 3 this fact is noted and in general no meaningful operation counts can be provided for $|z| \geq 15$.

A survey of Table 3 provides some meaningful comparison among the various methods or implementations. For purely imaginary arguments, application of Cody's approximations render the most efficiency. This is not surprising because these are essentially best approximations for a particular function of one real variable. On the other hand, for large arguments there is room for a significant improvement in these approximations, as noted before. For an overall coverage of arguments in the entire complex plane, Kuki's and Spira's analyses definitely render the best approach in efficiency. Whereas Kuki's arrangement is superior in numerical stability, Spira's error control excells in simplicity and flexibility. Table 3 also shows inferior results for Lucas and Luke, due to different reasons. Luke's method is intrinsically expensive. It involves $(2n+2m)$ recurrences which contribute a large number of arithmetic operations. Even the saving of one function call for some arguments is far outweighed by the large number of arithmetic operations. Inefficiency in Lucas' application comes from three main sources. First, he applies eq. (1) in a non-optimal way, requiring n calls to the complex logarithms. Second, his truncation error bound is too restrictive, posing severe confinement in the application of Stirling's approximation. Third, his numerical data are not arranged carefully to avoid unnecessary overhead in arithmetic operations. For example, the eleven divisions involved in computing the coefficients C(J) are entirely superfluous.

So far we have only considered arithmetic operations for argument values on the right half complex plane. For those on the left half all authors recommend the use of the reflection formula (2), which involves a complex sine and, for log-gamma, a complex logarithm. Therefore it would be useful to alleviate this use for at least some arguments. Spira's analysis is a contribution in this respect. From eq. (16) and Fig. 1 (iii) we see how the use of (2) can be minimized.

TABLE 3

Counts of arithmetic operations and function calls in addition to those recorded in Table 2.

Target precisions are 10^{-7} for short and 10^{-16} for long.

|z|	arg z	Cody		Kuki		Lucas		Luke		Spira	
		Short	Long	Short	Long	Short	Long	Short	Long	Short	Long
1	0 (or ϵ in Luke's case)	Not Applicable		32M+32A+L	68M+64A+L	45M+22A+9L	61M+25A+9L	768M+682A	1356M+1240A	40M+39A+L	80M+75A+L
5				16M+16A	52M+48A+L	45M+18A+5L	61M+25A+5L	652M+584A	1260M+1164A	20M+20A	60M+56A+L
10				16M+16A	32M+28A	45M+13A	61M+25A	568M+536A	1156M+1094A	16M+16A	36M+31A
15				16M+16A	32M+28A	45M+13A	57M+22A	396M+406A	1100M+1062A	16M+16A	32M+28A
20				16M+16A	32M+28A	45M+13A	61M+25A	516M+538A	*	16M+16A	28M+25A
25				16M+16A	32M+28A	45M+13A	61M+25A	644M+572A	*	16M+16A	28M+25A
30				16M+16A	32M+28A	45M+13A	57M+22A	684M+612A	*	16M+16A	28M+25A
1	π/4			32M+32A+L	68M+64A+L	45M+27A+10L	61M+29A+9L	788M+704A	1416M+1306A	40M+39A+L	80M+75A+L
5				16M+16A	52M+48A+L	45M+23A+6L	61M+29A+5L	704M+656A	1392M+1324A	20M+20A	60M+56A+L
10				16M+16A	32M+28A	45M+18A+L	61M+29A	788M+778A	1416M+1380A	16M+16A	36M+31A
15				16M+16A	32M+28A	45M+18A+L	57M+26A	*	*	16M+16A	32M+28A
20				16M+16A	32M+28A	45M+18A+L	61M+29A	*	*	16M+16A	28M+25A
25				16M+16A	32M+28A	45M+18A+L	61M+29A	*	*	16M+16A	28M+25A
30				16M+16A	32M+28A	45M+18A+L	57M+26A	*	*	16M+16A	28M+25A
1	π/2	13M+8A - $\frac{1}{2}$L	24M+19A - $\frac{1}{2}$L	36M+36A+L	72M+68A+L	45M+23A+10L	61M+29A+10L	808M+726A	1416M+1306A	40M+39A+L	80M+75A+L
5		12M+5A + $\frac{1}{6}$L	18M+11A + $\frac{1}{6}$L	28M+28A+L	72M+68A+L	45M+23A+10L	61M+29A+5L	868M+792A	1576M+1482A	20M+20A	60M+56A+L
10		12M+5A + $\frac{1}{6}$L	18M+11A + $\frac{1}{6}$L	20M+20A	52M+48A+L	45M+23A+10L	61M+29A+10L	1008M+946A	1756M+1680A	16M+16A	36M+31A
15		12M+5A + $\frac{1}{6}$L	18M+11A + $\frac{1}{6}$L	20M+20A	36M+32A	45M+28A+15L	57M+26A+15L	1168M+1122A	*	16M+16A	32M+28A
20		12M+5A + $\frac{1}{6}$L	18M+11A + $\frac{1}{6}$L	20M+20A	36M+32A	45M+33A+20L	61M+29A+20L	*	*	16M+16A	28M+25A
25		12M+5A + $\frac{1}{6}$L	18M+11A + $\frac{1}{6}$L	20M+20A	36M+32A	45M+38A+25L	61M+29A+25L	*	*	16M+16A	28M+25A
30		12M+5A + $\frac{1}{6}$L	18M+11A + $\frac{1}{6}$L	20M+20A	36M+32A	45M+43A+30L	57M+26A+30L	*	*	16M+16A	28M+25A

* Convergence is too slow for any practical application.

IV. Comparison and Testing of Algorithms

Of the five authors mentioned above, only two have actually published algorithms accompanying their analyses. They appear in this journal as Algorithms 404 and 421, both coded in ANSI FORTRAN. In this section we mainly attend to the comparison of these two algorithms. As indicated in the last section, there is much theoretical evidence to believe in the superiority of A421 over A404. This conclusion will be further substantiated here by empirical data. Before such quantitative results are presented, it may be useful to render a few qualitative remarks about each algorithm.

1. Comments on Algorithm 404

(i) The overall algorithm is easy to follow, with sufficient comments at strategic places to indicate the different blocks of actions to be executed. However, throughout the program, we can find obvious instances of inefficient coding. For example, near statement 70, $\frac{1}{2}\log(2\pi$ is actually computed via a call to the logarithm and near statement 100, A = CMPLX(FLOAT(I-1), 0.) is realized by a statement that causes an unnecessary floating multiplication. Another instance of serious inefficiency is the generous but unnecessary usage of the complex logarithm. As mentioned in the last section, the complex logarithm is used n times for recursion. In addition, near statement 120 the logarithmic version of the reflection formula is used, followed by a complex exponential. The last process should be reversed, thereby saving a call to the complex logarithm.

(ii) The treatment of the function near the poles is somewhat mysterious and misleading. First it was given some remarks on an empirical relation between the number of significant figures obtained by Stirling's series and the distance from z to the nearest pole z_0, say,

$\delta = z-z_0$. These remarks are irrelevant to the algorithm because the use of Stirling's series is confined to the right half of the complex plane which does not contain any pole. In fact, the poles appear through the term $\sin \pi z$. When z is too close to a pole, i.e., $\delta <$ TOL = limit of precision of a computer system in question, the result 1/TOL is returned. This result can be misleading because TOL is not given as an exact machine constant in the algorithm and the result given deviates substantially from the true function value which should be approximated by $1/(\text{TOL}*z_0!)$. In fact a better approach would have been a test on whether z is exactly a negative integer and a return for just that case.

(iii) Numerical data are not in general given in the most efficient form. For example, the data vector C(I) could have been stored as floating point constants in DATA statements, with the rational forms given in comments for conversion to other computers. This arrangement saves 11 divisions. All these constants can then be put together with PI, TOL and IOUT as machine-dependent numbers. The change of these 11 clearly identified constants requires trivial effort.

(iv) It is more desirable to have the function subprogram in the form of a subroutine so as to include a call to log-gamma. There are two reasons for such desire. First, the use of log-gamma allows application in a much larger portion of the complex plane. Second, the algorithm first computes log-gamma and then takes the complex exponential to obtain the result. It would be inefficient for a user who desires log-gamma to compute the logarithm of a result which is the exponential of the desired answer.

(v) A cursory examination of the documentation of this algorithm reveals that the authors' testing is far from adequate. The test of the reflection formula, which is also used in the algorithm, provides very little information about its reliability. In fact, for all values of z outside the strip $0 < \text{Real}(z) \le 1$, this test shows no more than the proper incorporation of the formula in the subprogram. The testing of the algorithm against n! is likewise deficient. One wonders why it was not even tested again $\Gamma(x)$ which is provided on the IBM 360 Fortran library. The comparison of the algorithm against tabulated values may be more thorough, but the authors do not provide information on how extensive this was done. Did they compare 10 or 100 or 1000 cases? Such inadequate information about the testing of this algorithm raises serious quesions about its reliability.

(vi) It would be useful to include in comments a list of external references.

2. Comments on Algorithm 421

(i) The overall algorithm is meticulously coded to yield utmost efficiency and accuracy. For example a quantity squared is accomplished by a multiplication rather than a call to the exponentiation; such sequence as $\{(z+k)/(z+n)\}$ is not computed through straightforward addition so that round-off error accumulation is minimized. In short, it is a striking example of superb coding.

(ii) There are sufficient comments for a reader to follow through the code. However, there is a lack of identification of the constants stored in data. For example, it is not at all obvious that HL2P is $\ell n\sqrt{2\pi}$. It would be helpful if a group of logical flags (LFO, LF1, etc.) be more explicitly identified.

(iii) Like A404 there is no list of external references.

(iv) The answer returned by the algorithm when the argument is too close to a singularity should be Ω, instead of $\Omega + i\Omega$. (Ω is the largest floating point number representable in the machine.)

3. Accuracy Tests on Algorithm 421

This algorithm is compiled and executed on a UNIVAC 1108 computer, with minor changes of the machine-dependent constants EPS, OMEGA and DEO. Three stages of testing are carried out, in increasing degree of intensity.

(i) Blunder check - 40 sets of complex results are computed from the algorithm and compared with a published numerical table on Pp. 277-287 [7]. These results all agree to the last significant figure given by the table.

(ii) Identity check - The region $-30 \leq \mathrm{Re}\, z \leq 30$ and $-30 \leq \mathrm{Im}\, z \leq 30$ are divided into 2000 strips parallel to the imaginary axis. For a uniformly random argument z in each strip $\ell n\Gamma(z)$, $\ell n\Gamma(z+\frac{1}{2})$ and $\ell n\Gamma(2z)$ are computed and tested against the duplication formula (4). This procedure is repeated for 2000 strips parallel to the real axis. The maximum absolute discrepancy from this identity is 1.3D-15 for log-gamma, which is consistent with the magnitudes of error reported by the author of this algorithm.

(iii) Automatic tabular comparison - For the sake of more thorough testing, we have constructed a reference subprogram QPCGAM which computes the complex gamma function in extended precision using a package of subroutines in 70-bit (about 21 decimal) arithmetic, composed by C. L. Lawson and associates at the Jet Propulsion

Laboratory. This reference subprogram is based on a Stirling's approximation with 9 terms, having a truncation error less than 2×10^{-19}. QPCGAM is itself subjected to the kind of testing described in (ii) above and yields a maximum absolute discrepancy less than 10^{-18} for log-gamma. Algorithm 421 is then compared against QPCGAM for 7 rectangular regions of the complex plane. For comparison each region is divided into 500 strips parallel to the imaginary axis and for each strip a random argument is chosen, making a total of 500 test arguments for each region. For log-gamma this process is repeated for strips parallel to the real axis. The results of the comparison are summarized in Tables 4 and 5 where "error" means the difference between A421 and QPCGAM.

Performance statistics recorded here render empirical confirmation to our qualitative remarks made in the last section. For example, we see that the absolute error for log-gamma indeed serves as an upper bound to the relative error. We also observe that all the precautionary measures taken by Kuki to alleviate cancellation error and serious accumulation of round-off error are functioning properly. The performance statistics found here are consistent with those reported by him, except for the fact the errors found by us are uniformly smaller than those by him. The last fact can be readily understood in terms of the smaller truncation error for long precision arithmetic on the UNIVAC 1108 computer than that on the IBM 360 O/S. All in all, our intensive and extensive testing has provided us much confidence in the reliability of this algorithm.

TABLE 4

Relative Errors for Gamma Function by A421

Interval of Re z	Interval of Im z	Max. Relative Error	RMS Relative Error
[0, 15]	[0, 15]	8.3D-17	1.2D-17
[0, 15]	[15, 30]	2.2D-16	3.2D-17
[15, 30]	[0, 15]	1.0D-16	1.7D-17
[15, 30]	[15, 30]	1.7D-16	3.2D-17
[-30, 0]	[0, 30]	1.8D-16	2.7D-17
[-30, 0]	[-30, 0]	2.8D-16	4.7D-17
[0, 30]	[-30, 0]	2.5D-16	4.5D-17

TABLE 5

Absolute and Relative Errors for Log-Gamma Function by A421

(a) Random Arguments in Strips Parallel to Imaginary Axis

Interval of Re z	Interval of Im z	Maximum Relative Error	RMS Relative Error	Maximum Absolute Error	RMS Absolute Error
[0, 15]	[0, 15]	2.4D-17	2.1D-18	1.1D-16	2.4D-17
[0, 15]	[15, 30]	4.2D-18	8.7D-19	2.7D-16	6.0D-17
[15, 30]	[0, 15]	3.4D-18	7.1D-19	2.5D-16	3.9D-17
[15, 30]	[15, 30]	3.8D-18	8.0D-19	3.1D-16	6.5D-17
[-30, 0]	[0, 30]	4.2D-18	7.9D-19	5.8D-16	7.2D-17
[-30, 0]	[-30, 0]	4.0D-18	8.3D-19	4.0D-16	7.8D-17
[0, 30]	[-30, 0]	9.2D-18	1.1D-18	3.0D-16	5.6D-17

(b) Random Arguments in Strips Parallel to Real Axis

[0, 15]	[0, 15]	7.8D-17	5.1D-18	1.4D-16	2.4D-17
[0, 15]	[15, 30]	3.5D-18	7.5D-19	1.9D-16	4.0D-17
[15, 30]	[0, 15]	4.5D-18	9.3D-19	3.1D-16	6.2D-17
[15, 30]	[15, 30]	4.5D-18	8.1D-19	3.3D-16	6.8D-17
[-30, 0]	[0, 30]	6.4D-18	9.2D-19	3.0D-16	5.7D-17
[-30, 0]	[-30, 0]	4.3D-18	7.8D-19	5.3D-16	7.8D-17
[0, 30]	[-30, 0]	8.6D-18	9.0D-19	4.2D-16	7.5D-17

4. Accuracy Tests on Algorithm 404

A404 is compiled and executed on a UNIVAC 1108 computer, with minor changes of the machine-dependent constants IOUT, PI and TOL. Since this algorithm is written for short precision, it may be compared against A421 the validity and reliability of which have been established. Automatic tabular comparison as described above yields the performance statistics in Table 6. For Re $z \geq 0$, our results are consistent with those reported by Lucas.

5. Timing Tests on Algorithms 404 and 421

In a time-sharing operating system precise timing of a computer program is not too meaningful, because such timing is dependent on the transitory operating environment. For this reason we have conducted a set of relative timing tests at different times in a two - day period. The average so obtained should serve as a good indication of the efficiency of the program tested. As an additional aid to the interpretation of the timing tests, the double precision exponential function DEXP is tested along with each algorithm so that the efficiency can also be expressed in terms of units of DEXP. Thus for each set of tests all three programs are executed in the same computer run, for 1000 test arguments selected in the same way as described in the accuracy tests, with proper alternation between vertical and horizontal strips. The statistics are reported in Table 7. These results confirm our earlier remarks on the superiority in efficiency of A421 over A404. In fact, it is significant that A421 yields almost 3 times the precision as A404 and is still better than A404 in efficiency.

TABLE 6

Relative Errors for Gamma Function by A404

(a) Random Arguments in Strips Parallel to Imaginary Axis

Interval of Re z	Interval of Im z	Max. Relative Error	RMS Relative Error
[0, 15]	[0, 15]	1.4E-6	2.3E-7
[0, 15]	[15, 30]	3.8E-6	6.9E-7
[15, 30]	[0, 15]	2.1E-6	4.4E-7
[15, 30]	[15, 30]	3.2E-6	5.4E-7
[-15, 0]	[0, 15]	7.8E-6	1.1E-6
[-15, 0]	[-15, 0]	6.9E-6	1.1E-6
[0, 15]	[-15, 0]	2.1E-6	3.7E-7

(b) Random Arguments in Strips Parallel to Real Axis

[0, 15]	[0, 15]	1.3E-6	2.4E-7
[0, 15]	[15, 30]	4.4E-6	7.4E-7
[15, 30]	[0, 15]	2.5E-6	4.6E-7
[15, 30]	[15, 30]	2.9E-6	5.1E-7
[-15, 0]	[0, 15]	5.6E-6	1.1E-6
[-15, 0]	[-15, 0]	1.1E-5	1.2E-6
[0, 15]	[-15, 0]	1.0E-6	2.1E-7

TABLE 7

Average Timing for A421 and A404
With DEXP as a Reference Program

Tests	A404	A421	DEXP
1	2.85 msec	1.85 msec	0.19 msec
2	2.38	1.79	0.18
3	2.34	1.15	0.18
4	3.01	2.37	0.19
5	2.99	1.87	0.19
Average	2.71	1.93	0.19

V. Conclusion

We have investigated five suggested approaches for the computation of the complex gamma function. Our comparison, which is most concerned with accuracy and efficiency, leads us to conclude that Cody's approximation is best for this function of imaginary arguments and a combination of Kuki's and Spira's analyses renders the best method for this function of general complex arguments. Furthermore, this comparison also demonstrates that Kuji's meticulous rearrangement of mathematical formulae and precautionary steps in detail implementation lead to his high-quality algorithm. Therefore we recommend without reservation that this algorithm be used where appropriate, either in individual application or in program libraries.

Acknowledgement

Discussions with our colleagues Dr. C. L. Lawson and Dr. F. Krogh have been helpful, especially concerning the overall problem of testing mathematical software. This paper presents the result of research supported in part by an Office of Naval Research Grant No. NR044-440 and in part by National Aeronautics and Space Administration Contract No. NAS7-100.

References

1. W. J. Cody and K. E. Hillstrom, "Chebyshev Approximations for the Coulomb Phase Shift", Math. Comp. Vol. 24, Pp. 671-678. (1970)

2. H. Kuki, "Algorithm 421 - Complex Gamma Function with Error Control", Comm. ACM Vol. 15, Pp. 262-267, 271-272. (1972)

3. C. W. Lucas and C. W. Terrill, "Algorithm 404 - Complex Gamma Function", Comm. ACM Vol. 14, Pp. 48-49. (1971)

4. Y. L. Luke, "Evaluation of the Gamma Function by Means of Padé Approximations", SIAM J. Math. Anal., Vol. 1, Pp. 266-281. (1970)

5. R. Spira, "Calculation of the Gamma Function by Stirling's Formula", Math. Comp. Vol. 25, Pp. 317-322. (1971)

6. N. Nielsen, Die Gammafunction. Band I. Handbuch der Theorie der Gammafunktion. Band II. Theorie des Integrallogarithmus und verwandter Transzendenten, Chelsea, N. Y. (1965)

7. NBS, Handbook of Mathematical Functions, Appl. Math. Series #55, eds. M. Abramowtiz and I. Stegun, Chap. 6. (1964)

8. E. T. Whittaker and G. N. Watson, "A Course of Modern Analysis", 4th ed., Cambridge Univ. Press, N. Y. (1962)

9. Wimp, J., "On Recursive Computation", Aerospace Research Laboratories Report ARL 69-0186, November 1969, Clearinghouse, U. S. Department of Commerce, Springfield, Virginia 22151.

NASA — JPL — Coml., L.A., Calif.