# 2.3  Gamma and Log-Gamma Functions

## A.  Purpose

These subprograms compute values of the gamma function and the natural logarithm of the gamma function, [1] and [2]. The gamma function is defined by

$$\Gamma(x) = \int_0^\infty e^{-s} s^{x-1} ds.$$

For integer values of $n \geq 0$, it satisfies the relation

$$n! = \Gamma(n+1).$$

## B.  Usage

### B.1  Program Prototype, Single Precision

**REAL  SGAMMA, SLGAMA, X, Y**

Assign a value to X and obtain gamma or the natural logarithm of gamma, respectively, through use of the function statements:

| **Y =SGAMMA(X)** | **Y =SLGAMA(X)** |

### B.2  Argument Definition

**X**  [in] Argument at which function evaluation is desired. See Section E for restrictions on X.

### B.3  Modifications for Double Precision

For double precision usage, change the REAL type statement to DOUBLE PRECISION and change the function names from SGAMMA to DGAMMA and SLGAMA to DLGAMA.

### B.4  Program Prototype, Complex

**INTEGER  MODE**

**COMPLEX  CARG, CVAL**

**REAL  ERREST**

Assign the complex argument in CARG. Set MODE to 0 or 1.

> **CALL CGAM(CARG, CVAL, ERREST, MODE)**

The complex result will be in CVAL with an error estimate in ERREST.

### B.5  Argument Definitions

**CARG**  [in] Complex argument. The argument must not be zero or a negative real integer value.

**CVAL**  [out] Computed complex value of either log-gamma or gamma. The imaginary part of log-gamma, say $v$, will be standardized to satisfy $-\pi < v \leq \pi$.

**ERREST**  [out] On return ERREST gives an estimate of the absolute error (for log-gamma) or the relative error (for gamma) of the computed value.

**MODE**  [in] Set value to 0 for log-gamma and 1 for gamma.

### B.6  Program Prototype, Double Precision Complex

**INTEGER  MODE**

**DOUBLE PRECISION  CARG(2),  CVAL(2), ERREST**

Assign the complex argument in CARG(), with real part in CARG(1) and imaginary part in CARG(2). Set MODE to 0 or 1.

> **CALL ZGAM(CARG, CVAL, ERREST, MODE)**

The result will be in CVAL() with an error estimate in ERREST.

### B.7  Argument Definitions

**CARG()**  [in] Array of 2 values representing the complex argument, CARG(1) for the real part and CARG(2) for the imaginary part. The argument must not be zero or a negative real integer value.

**CVAL()**  [out] Array of 2 values representing the complex function value returned: CVAL(1) and CVAL(2) for real and imaginary parts, respectively. The imaginary part of log-gamma, say $v$, will be standardized to satisfy $-\pi < v \leq \pi$.

**ERREST**  [out] On return ERREST gives an estimate of the absolute error (for log-gamma) or the relative error (for gamma) of the answer.

**MODE**  [in] Set value to 0 for log-gamma and 1 for gamma.

Although the Fortran 77 standard does not support a double precision complex type, many vendors do provide for this using a declaration COMPLEX*16 or DOUBLE COMPLEX. With such a compiler one can probably use

ZGAM with its first two arguments declared as double precision complex.

## C.  Examples and Remarks

### C.1  Example

The gamma function satisfies a "duplication" identity which may be written as

$$z = \frac{2\sqrt{\pi}\Gamma(x)}{2^x \Gamma(x/2)\Gamma(x/2 + \frac{1}{2})} - 1 = 0.$$

The listing of DRSGAMMA and ODSGAMMA gives an example of using these subprograms to evaluate this identity. The program DRCGAM and output ODCGAM illustrate the use of CGAM.

### C.2  Remarks

Neither the Fortran 77 standard nor the Fortran 90 standard includes intrinsic functions for gamma or log-gamma; however, such functions with the names GAMMA, DGAMMA, LGAMMA, ALGAMA, and DL-GAMA are provided as intrinsic functions in some vendors' Fortran systems. In particular these latter functions are provided with the UNISYS ASCII Fortran and IBM VS-Fortran, but not with the VAX-11 Fortran. In the UNISYS and IBM systems GAMMA and LGAMMA are "generic" names and thus can be used with either single or double precision arguments.

In a system having DGAMMA or DLGAMA as intrinsics a reference to one of these function names will cause the vendor-supplied code to be used. If one wishes to override this and use the code from this library one must declare the function name to be EXTERNAL in the referencing program unit.

## D.  Functional Description

The gamma function is defined and takes positive values for all real $x > 0$, becoming unbounded both as $x \to 0$ and as $x \to +\infty$. As $x \to 0$, $\Gamma(x)$ is asymptotic to $1/x$, and as $x \to \infty$, $\Gamma(x)$ is asymptotic to

$$h(x) = (x/e)^x \sqrt{2\pi/x}.$$

As an estimate of the size of $\Gamma(x)$ for large positive real $x$, note that for $x > 9$, $\Gamma(x)$ satisfies $h(x) < \Gamma(x) < 1.01\, h(x)$. The gamma function is also defined for negative $x$ except at the negative integers where it has poles.

As a function of a complex variable the gamma function is analytic throughout the complex plane except for poles at zero and the negative real integer points. See [1] and [2] for further discussion of the gamma function.

The subprograms SGAMMA, DGAMMA, SLGAMA, and DLGAMA are based on subprograms developed by

W.J. Cody, Argonne National Laboratory, which were designed for $10^{-20}$ precision. These subprograms use rational minmax approximations on selected subintervals as well as asymptotic formulas and argument reduction techniques. The degrees of the rational approximations are not varied between the single precision and double precision subprograms or as a function of the host computer's precision.
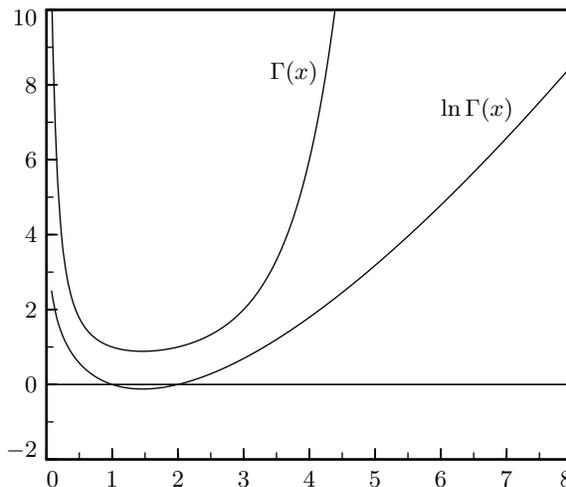


Subprograms CGAM and ZGAM are based on [3], making use of the Stirling asymptotic series and recursive formulas. The accuracy adjusts to the host machine up to about 17 significant decimal digits. CGAM and ZGAM reference I1MACH(10) to find the radix of the host system's floating point arithmetic.

**Accuracy tests**

Accuracy tests were run on a Univac 1108 in 1969 and on a Univac 1100 in 1983. The arithmetic precision of these systems is $\rho_1 = 2^{-27} \approx 0.745\mathrm{E}{-8}$ for single precision and $\rho_2 = 2^{-60} \approx 1.15\mathrm{E}{-18}$ for double precision.

Subprograms SGAMMA and SLGAMA were tested in 1983 on a Univac 1100 by comparison with DGAMMA and DLGAMA at 18,000 points. The test results may be summarized as follows:

| Function | Argument Interval | Max. Rel. Error |
|---|---|---|
| SGAMMA | [0., 1.] | $2.5\rho_1$ |
| | [1., 2.] | $1.4\rho_1$ |
| | [2., 10.] | $9.2\rho_1$ |
| | [10., 17.] | $100.0\rho_1$ |
| | [17., 30.] | $180.0\rho_1$ |
| SLGAMA | [0.0, 0.5] | $2.7\rho_1$ |
| | [0.5, 4.0] | $6.5\rho_1$ |
| | [4.0, 35.0] | $2.8\rho_1$ |

To test the double precision functions, and as an additional test of the single precision functions, the function

$z(x)$ defined in Section C and a logarithmic form of $z(x)$ defined as

$$w(x) = \text{LGAM}(x) + 0.5 \ \ln(2\pi) + (0.5 - x)\ln(2)$$
$$-\text{LGAM}(0.5x) - \text{LGAM}(0.5x + 0.5) = 0$$

were evaluated on a Univac 1100 in 1983 at about 100 points. These tests are summarized as follows ("Max. Error" is the maximum magnitude of the test function):

| Subprogram Used | Argument Interval | Test Function | Max. Error |
|---|---|---|---|
| SGAMMA | $[-20.75, -11.25]$ | $z$ | $114\rho_1$ |
| | $[-11.25, 11.50]$ | $z$ | $6\rho_1$ |
| | $[11.50, 34.00]$ | $z$ | $100\rho_1$ |
| DGAMMA | $[-20.75, 43.60]$ | $z$ | $130\rho_2$ |
| | $[43.60, 106.80]$ | $z$ | $384\rho_2$ |
| | $[106.80, 170.00]$ | $z$ | $1186\rho_2$ |
| SLGAMA | $[0.0625, 3.1]$ | $w$ | $3\rho_1$ |
| | $[3.1, 11.5]$ | $w$ | $8\rho_1$ |
| | $[11.5, 0.12\text{E}37]$ | $w/\text{SLGAMA}$ | $2.7\rho_1$ |
| DLGAMA | $[0.0625, 12.0]$ | $w$ | $110\rho_2$ |
| | $[12.0, 0.77\text{D}305]$ | $w/\text{DLGAMA}$ | $2.9\rho_2$ |

Subprogram ZGAM was tested in 1969 on a Univac 1108. The maximum relative error noted for complex arguments, $x$, satisfying $|x| < 30$, was $345\rho_2$ for gamma and $93\rho_2$ for log-gamma. The test function, $z$, was evaluated at a few complex points, $x$, of magnitude near one on a Univac 1100 in 1983. The maximum value of $|z|$ was $30\rho_2$.

### References

1. Milton Abramowitz and Irene A. Stegun, **Handbook of Mathematical Functions**, *Applied Mathematics Series 55*, National Bureau of Standards (1966).

2. J. F. Hart et al., **Computer Approximations**, J. Wiley and Sons, New York (1968) Section 6.6.

3. H. Kuki, *Complex gamma function with error control*, **Comm. ACM 15**, 4 (April 1972) 262–267 and 271–272.

## E.   Error Procedures and Restrictions

Subprograms SGAMMA and DGAMMA accept negative as well as positive arguments. These subprograms will issue an error message and return the largest machine number, $\Omega$, if $x$ is zero or a negative integer, or if $x \geq x_g$.

The subprograms SLGAMA and DLGAMA are designed only for positive arguments. These subprograms will issue an error message and return the value, $\Omega$, if $x \leq 0$, or if $x \geq x_\lambda$.

The complex gamma subroutines CGAM and ZGAM issue an error message and return the complex value $(\Omega, \Omega)$ if the argument is zero, a negative real integer, or a large argument that would cause overflow.

Let $\Omega$ denote the machine overflow limit. Let $x_g$ denote the value of $x$ for which $\Gamma(x) = 0.875 \ \Omega$, and let $x_\lambda$ denote the value of $x$ for which $\ln(\Gamma(x)) = 0.875 \ \Omega$. Some examples of these values follow.

| Computing System | $x_g$ | $x_\lambda$ |
|---|---|---|
| VAX 11/780, SP & DP | 34.8 | 1.8e36 |
| UNISYS (Sperry) 1100, SP | 34.5 | 1.8e36 |
| IEEE Math processor, SP | 35.0 | 3.6e36 |
| IBM Mainframe 3xxx, DP | 57.5 | 3.8e73 |
| UNISYS (Sperry) 1100, DP | 171.5 | 1.1e305 |
| IEEE Math processor, DP | 171.6 | 2.2e305 |
| Cray, SP & DP | 966.9 | 8.5e2461 |

The values listed above are computed on a first-time flag at run time making use of the overflow limit obtained from R1MACH(2) or D1MACH(2).

## F.   Supporting Information

| Entry | Required Files |
|---|---|
| **CGAM** | AMACH, CGAM, ERFIN, ERMSG, SERM1, SERV1 |
| **DGAMMA** | AMACH, DERM1, DERV1, DGAMMA, ERFIN, ERMSG |
| **DLGAMA** | AMACH, DERM1, DERV1, DGAMMA, DLGAMA, ERFIN, ERMSG |
| **SGAMMA** | AMACH, ERFIN, ERMSG, SERM1, SERV1, SGAMMA |
| **SLGAMA** | AMACH, ERFIN, ERMSG, SERM1, SERV1, SGAMMA, SLGAMA |
| **ZGAM** | AMACH, DERM1, DERV1, ERFIN, ERMSG, ZGAM |

Subprograms SGAMMA, DGAMMA, SLGAMA, and DLGAMA were developed by W.J. Cody, Argonne National Lab., 1982, and adapted to the JPL MATH 77 library by C. Lawson and S. Chiu, JPL, 1983.

Subroutines CGAM and ZGAM were developed by H. Kuki with the name CDLGAM ([3] above), adapted for JPL Univac 1108 usage by E.W. Ng, 1969, and adapted to the JPL MATH 77 library by C. Lawson and S. Chiu, JPL, 1983.

# DRSGAMMA

```
c        DRSGAMMA
c>> 1996-07-12 DRSGAMMA Krogh   Minor changes for conversion to C.
c>> 1994-10-19 DRSGAMMA Krogh   Changes to use M77CON
c>> 1992-06-11 DRSGAMMA Krogh   Changed "duplication" identity.
c>> 1991-10-21 DRSGAMMA CLL
c>> 1987-12-09 DRSGAMMA Lawson   Initial Code.
c
c--S replaces "?": DR?GAMMA, ?GAMMA, ?LGAMA
      external SGAMMA, SLGAMA
      real           SGAMMA, SLGAMA
      real           LGAM, GAM, PI, X(12), Z
      integer I
c
      data X /-1.5e0, -0.5e0, 0.5e0,  1.0e0,  1.5e0,   2.0e0,
     *           3.0e0, 4.0e0, 5.0e0, 10.0e0, 20.0e0, 30.0e0 /
      data PI / 3.1415926535897932385e0 /
c
      print '('' X'',9X,''SGAMMA'',10X,''SLGAMA'',10X,''Z''/1x)'
c
      do 10 I = 1, 12
        GAM = SGAMMA(X(I))
        Z = (2.e0 * SQRT(PI) * GAM) / ((2.0e0**X(I)) *
     1    SGAMMA(0.5e0*X(I)) * SGAMMA(0.5e0*X(I)+.5e0)) - 1.e0
        if(X(I) .lt. 0.0e0) then
          print '(1x,F4.1,3X,G15.8, 13x ,1X,E13.2)', X(I), GAM,   Z
        else
          LGAM = SLGAMA(X(I))
          print '(1x,F4.1,3X,G15.8,F13.5,1X,E13.2)', X(I), GAM, LGAM, Z
        endif
   10 continue
      end
```

# ODSGAMMA

| X | SGAMMA | SLGAMA | Z |
|---|---|---|---|
| −1.5 | 2.3632717 | | 0.00E+00 |
| −0.5 | −3.5449078 | | 0.00E+00 |
| 0.5 | 1.7724539 | 0.57236 | 0.00E+00 |
| 1.0 | 1.0000000 | −0.00000 | 0.00E+00 |
| 1.5 | 0.88622695 | −0.12078 | 0.00E+00 |
| 2.0 | 1.0000000 | 0.00000 | 0.00E+00 |
| 3.0 | 2.0000000 | 0.69315 | 0.00E+00 |
| 4.0 | 6.0000000 | 1.79176 | 0.00E+00 |
| 5.0 | 24.000000 | 3.17805 | 0.00E+00 |
| 10.0 | 362880.00 | 12.80183 | −0.12E−06 |
| 20.0 | 0.12164482E+18 | 39.33989 | −0.24E−05 |
| 30.0 | 0.88417885E+31 | 71.25703 | 0.64E−05 |

# DRCGAM

```
      program DRCGAM
c>> 2001-05-22 DRCGAM Krogh Minor change for making .f90 version.
c>> 1996-06-14 DRCGAM Krogh Changes for C conversion.
c>> 1994-08-17 CLL
c>> 1991-10-18 Demo driver for CGAM
c     ————————————————————————————————————————————————————
c—   D version uses DRZGAM, ZGAM
c—   S version uses DRCGAM, CGAM
c     ————————————————————————————————————————————————————
      real ARG(2),GAM(2),LGAM(2), X(2,8)
      real ERR0, ERR1
      integer I, J
      data ( ( X(I,J),I=1,2 ), J=1,8 ) /
     *  1.0E0,0.0E0,  1.0E0,0.5E0,  1.5E0,0.0E0,  1.5E0,0.5E0,
     *  0.0E0,0.5E0, −1.5E0,0.5E0,  −1.5E0,0.0E0,  −1.5E0,−0.5E0 /
c     ————————————————————————————————————————————————————
      print '(5x,''X            GAMMA           Err Est'', 11x,
     *  ''LGAMA          Err Est'')'
      do 10  J = 1,8
        ARG(1) = X(1,J)
        ARG(2) = X(2,J)
        call CGAM (ARG,GAM,ERR1,1)
        call CGAM (ARG,LGAM,ERR0,0)
        print
     *  '(1x,f5.1,g20.10,g10.2,g22.10,g10.2/1x,f5.1,g20.10,g32.10/)',
     *       ARG(1),GAM(1),ERR1, LGAM(1),  ERR0,
     *       ARG(2),GAM(2),        LGAM(2)
   10 continue
      end
```

# ODCGAM

| X | GAMMA | Err Est | LGAMA | Err Est |
|---|---|---|---|---|
| 1.0 | 0.9999998212 | 0.22E−05 | −0.2384185791E−06 | 0.22E−05 |
| 0.0 | 0.000000000 | | 0.000000000 | |
| | | | | |
| 1.0 | 0.8016943932 | 0.25E−05 | −0.1909457445 | 0.25E−05 |
| 0.5 | −0.1996398121 | | −0.2440582514 | |
| | | | | |
| 1.5 | 0.8862264752 | 0.23E−05 | −0.1207830906 | 0.23E−05 |
| 0.0 | 0.000000000 | | 0.000000000 | |
| | | | | |
| 1.5 | 0.7907390594 | 0.25E−05 | −0.2341866493 | 0.25E−05 |
| 0.5 | 0.2742525190E−01 | | 0.3466916084E−01 | |
| | | | | |
| 0.0 | −0.3992792964 | 0.26E−05 | 0.5022014380 | 0.26E−05 |
| 0.5 | −1.603387952 | | −1.814854503 | |
| | | | | |
| −1.5 | 0.9379166365 | 0.41E−05 | 0.8158981800E−03 | 0.41E−05 |
| 0.5 | 0.3492053747 | | 0.3564195633 | |
| | | | | |
| −1.5 | 2.363272190 | 0.44E−05 | 0.8600471020 | 0.44E−05 |
| 0.0 | −0.5165094876E−06 | | −0.000000000 | |
| | | | | |
| −1.5 | 0.9379166365 | 0.41E−05 | 0.8158981800E−03 | 0.41E−05 |

$-0.5$     $-0.3492053747$                    $-0.3564195633$