National Bureau of Standards
Statistical Engineering Division
Gaithersburg, MD 20899

N-86-2
October 1986

An Algorithm for Computing the Gamma C.D.F. to a Specified Accuracy

Charles P. Reeve

The cumulative distribution function (c.d.f.) of the gamma distribution (also known as the incomplete gamma ratio) is

$$P(x,\alpha) = \int_0^x t^{\alpha-1}e^{-t}dt/\Gamma(\alpha) \quad (x>0;\alpha>0) \tag{1}$$

where $\Gamma(\alpha) = \int_0^\infty t^{\alpha-1}e^{-t}dt$. [The left side of (1) is not standard statistical notation but is easier to work with in the derivations which follow]. The upper tail area of the distribution is

$$Q(x,\alpha) = \int_x^\infty t^{\alpha-1}e^{-t}dt/\Gamma(\alpha) \quad (x>0;\alpha>0).$$

$$= 1 - P(x,\alpha)$$

These functions are useful in computing the c.d.f.'s of the χ^2, noncentral χ^2, and Poisson distributions. For example,

$$Pr\{\chi^2(\nu)<x\} = P(x/2,\nu/2) \quad (x>0;\nu>0) \text{ and}$$

$$Pr\{Po(\lambda)<x\} = Q(\lambda,1+x) \quad (x=0,1,2,...;\lambda>0).$$

Two recent computer algorithms for computing $P(x,\alpha)$ are those of Bhattacharjee [2] and Lau [5]. The former uses a series expansion or continued fraction depending on the values of x and α. The latter uses a single series expansion for all x and α. In both cases the criterion for truncating the series is that the current term is less than a specified limit. As will be shown, this does not guarantee that the result will be accurate to that limit. Versions of these algorithms have been incorporated into software packages available at NBS [4,6,7].

In this note a modification of the Lau algorithm is presented which efficiently computes $P(x,\alpha)$ to a user-specified absolute accuracy. The algorithm makes use of the recurrence relation (see equation 6.5.21 of Abramowitz and Stegun [1])

$$P(x,\alpha) = x^\alpha e^{-x}/\Gamma(\alpha+1) + P(x,\alpha+1) \tag{2}$$

which relates the lower tail areas of the $\Gamma(\alpha)$ and $\Gamma(\alpha+1)$ distributions. If (2) is applied n times then

$$P(x,\alpha) = \sum_{i=1}^{n} x^{\alpha+i-1}e^{-x}/\Gamma(\alpha+i) + P(x,\alpha+n). \tag{3}$$

Given x, α, and an absolute accuracy limit $\varepsilon>0$, there will be a non-negative integer n such that $P(x,\alpha+n)<\varepsilon$ (the proof is left to the reader). The summation of n terms will then be an acceptable approximation to $P(x,\alpha)$.

An upper bound on $P(x,\alpha+n)$ is readily obtained by more applications of (2), hence

$$\begin{aligned} P(x,\alpha+n) &= \sum_{i=n+1}^{\infty} x^{\alpha+i-1}e^{-x}/\Gamma(\alpha+i) \\ &= \frac{x^{\alpha+n}e^{-x}}{\Gamma(\alpha+n+1)}\left[1 + \sum_{i=1}^{\infty} \frac{x^i}{(\alpha+n+1)(\alpha+n+2)\ldots(\alpha+n+i)}\right] \\ &< \frac{x^{\alpha+n}e^{-x}}{\Gamma(\alpha+n+1)}\left[1 + \sum_{i=1}^{\infty} \frac{x^i}{(\alpha+n+1)^i}\right] \\ &< \frac{x^{\alpha+n}e^{-x}}{\Gamma(\alpha+n+1)[1-x/(\alpha+n+1)]} \end{aligned} \tag{4}$$

when $x<\alpha+n+1$. When n becomes large enough so that the right hand side (RHS) of (4) is $<\varepsilon$, the series in (3) is truncated.

The above algorithm produces a sequence of gamma distributions whose lower tail areas steadily decrease when x is fixed, thus it can be called the "lower tail" method. When $x<\alpha$ relatively few terms may be required for convergence, but when $x>>\alpha$ an exceedingly large number of terms may be required. The Lau algorithm uses the above procedure for all x except that truncation occurs when $x^{\alpha+n}e^{-x}/\Gamma(\alpha+n+1) < \varepsilon$. This value is considerably smaller than the RHS of (4) when x is near (but less than) $\alpha+n+1$.

The next question to consider is the feasibility of applying the reverse procedure when $x>\alpha$, that is, producing a sequence of gamma distributions whose upper tail areas steadily decrease when x is fixed. This algorithm, called the "upper tail" method, makes use of the recurrence relation

$$Q(x,\alpha) = x^{\alpha-1}e^{-x}/\Gamma(\alpha) + Q(x,\alpha-1) \tag{5}$$

which is easily derived from (2). If (5) is applied m times then

$$Q(x,\alpha) = \sum_{i=1}^{m} x^{\alpha-i}e^{-x}/\Gamma(\alpha-i+1) + Q(x,\alpha-m) \tag{6}$$

provided that $\alpha-m>0$. This restriction limits the application of this algorithm to cases where $Q(x,\alpha-M)<\varepsilon$ where M is an integer such that $0<\alpha-M<1$. Olver [8] gives the bound

$$Q(x,\alpha-m) < \frac{x^{\alpha-m}e^{-x}}{\Gamma(\alpha-m)[x-\max(\alpha-m-1,0)]}, \tag{7}$$

thus

$$Q(x,\alpha-M) < x^{\alpha-M-1}e^{-x}/\Gamma(\alpha-M). \tag{8}$$

If the RHS of (8) is $<\varepsilon$ then the "upper tail" method can be used. Terms are summed as in (6) until the RHS of (7) is $<\varepsilon$ for some $m<M$. This method, when it can be used, is considerably faster than the "lower tail" method when $x>>\alpha$. To illustrate the behavior of the two methods, the c.d.f. was computed by both, when possible, for various combinations of x, α, and ε. The results are summarized in table 1. Note that when $x=\alpha$ the "upper tail" method converges a little faster.

These two algorithms have been implemented in the FORTRAN subroutine CDFGAM. If $x<0$ then 0 is returned. If $x>\alpha$ and convergence is determined to be possible, the "upper tail" method is used and $1-Q(x,\alpha)$ is returned. Otherwise the "lower tail" method is used and $P(x,\alpha)$ is returned. An exception occurs when x is in the extreme tails of the distribution. In order to prevent exponential underflow when computing the first term in (3) or (6), the exponent is tested to see if it is less than a pre-set underflow limit. If it is then $P(x,\alpha)$ is set to 0 or 1 as appropriate and neither method is used. An external routine, such as Reeve[9], must be provided for computing the log of the gamma function in double precision.

A listing of CDFGAM is an appendix to this note. It is invoked by

```
CALL CDFGAM(X,ALPHA,EPS,IFLAG,CDFX)
```

where the variable names are defined in the program documentation. The returned value of CDFX is valid only if IFLAG=0 on return. In passing ε (variable name EPS) to CDFGAM the user should realize that accuracy is limited by the number of digits carried in a single precision variable, and that roundoff error may affect the last one or two of these digits.

For further discussion of series expansions of the incomplete gamma function see Bouver and Bargmann [3].

Table 1

Comparison of the number of terms required for convergence of the gamma c.d.f. by the "lower tail" and "upper tail" methods [n and m respectively as in (3) and (6)]. A "-" indicates that neither method was used because x was in the extreme tails of the distribution, and a "*" indicates that the "upper tail" method does not produce convergence.

		$\alpha = 1$		$\alpha = 3$		$\alpha = 10$		$\alpha = 30$		$\alpha = 50$	
	x	n	m	n	m	n	m	n	m	n	m
	0.5	5	*	3	*	0	*	-	-	-	-
	1.0	6	*	4	*	0	*	-	-	-	-
	2.0	9	*	7	*	0	*	0	*	-	-
	3.0	11	*	9	*	2	*	0	*	-	-
	5.0	15	*	13	*	6	*	0	*	-	-
	7.0	19	*	17	*	10	*	0	*	0	*
$\varepsilon = 10^{-4}$	10.0	24	0	22	2	15	9	0	29	0	49
	15.0	31	0	29	0	22	7	2	27	0	47
	20.0	39	0	37	0	30	4	10	24	0	44
	30.0	52	0	50	0	43	0	23	18	3	38
	40.0	66	0	64	0	57	0	37	11	17	31
	50.0	78	0	76	0	69	0	49	4	29	24
	60.0	91	0	89	0	82	0	62	0	42	17
	80.0	-	-	-	-	106	0	86	0	66	1
	100.0	-	-	-	-	-	-	110	0	90	0

		$\alpha = 1$		$\alpha = 3$		$\alpha = 10$		$\alpha = 30$		$\alpha = 50$	
	x	n	m	n	m	n	m	n	m	n	m
	0.5	8	*	6	*	0	*	-	-	-	-
	1.0	11	*	9	*	2	*	-	-	-	-
	2.0	14	*	12	*	5	*	0	*	-	-
	3.0	17	*	15	*	8	*	0	*	-	-
	5.0	22	*	20	*	13	*	0	*	-	-
	7.0	26	*	24	*	17	*	0	*	0	*
$\varepsilon = 10^{-8}$	10.0	32	*	30	*	23	*	3	*	0	*
	15.0	41	*	39	*	32	*	12	*	0	*
	20.0	50	0	48	2	41	9	21	29	1	49
	30.0	65	0	63	0	56	5	36	25	16	45
	40.0	80	0	78	0	71	0	51	20	31	40
	50.0	94	0	92	0	85	0	65	14	45	34
	60.0	108	0	106	0	99	0	79	8	59	28
	80.0	-	-	-	-	126	0	106	0	86	15
	100.0	-	-	-	-	-	-	132	0	112	1

References

1. Abramowitz, Milton and Stegun, Irene A., Handbook of Mathematical Functions, NBS Applied Mathematics Series 55, 1970, p. 262.

2. Bhattacharjee, G.P., "The Incomplete Gamma Integral", Algorithm AS32, Applied Statistics, Vol. 19, No. 3, 1970, pp. 285-7.

3. Bouver, Hubert and Bargmann, Rolf E., "Numerical Solutions of the Incomplete Gamma Function", Computer Science and Statistics: Tenth Annual Symposium on the Interface, NBS Special Publication 503, 1978 pp. 302-7.

4. IMSL, Inc., Houston, TX. [MDGAM]

5. Lau, Chi-leung, "A Simple Series for the Incomplete Gamma Integral", Algorithm AS147, Applied Statistics, Vol. 29, No. 1, 1980, pp. 113-4.

6. NBS Core Math Library (CMLIB). [GAMI/GAMMA]

7. Numerical Algorithms Group (NAG), Downers Grove, IL. [G05DGE]

8. Olver, F.W.J., Asymptotics and Special Functions, Academic Press, 1974, p. 91.

9. Reeve, Charles P., "Accurate Computation of the Log of the Gamma Function", SED Note 86-1, October 1986.

```
      SUBROUTINE CDFGAM (X,ALPHA,EPS,IFLAG,CDFX)
C
C-----------------------------------------------------------------------
C   CDFGAM   WRITTEN BY CHARLES P. REEVE, STATISTICAL ENGINEERING
C            DIVISION, NATIONAL BUREAU OF STANDARDS, GAITHERSBURG,
C            MD  20899
C
C   FOR: COMPUTING THE CUMULATIVE DISTRIBUTION FUNCTION OF THE GAMMA
C        DISTRIBUTION (ALSO KNOWN AS THE INCOMPLETE GAMMA RATIO) TO A
C        SPECIFIED ACCURACY (TRUNCATION ERROR IN THE INFINITE SERIES).
C        THE ALGORITHM, DESCRIBED IN REFERENCE 2, IS A MODIFICATION OF
C        THE ALGORITHM OF REFERENCE 1.  THREE FEATURES HAVE BEEN ADDED:
C
C        1) A PRECISE METHOD OF MEETING THE TRUNCATION ACCURACY,
C        2) COMPUTATION OF THE UPPER TAIL AREA BY DECREMENTING ALPHA
C           WHEN THAT METHOD IS MORE EFFICIENT, AND
C        3) A CONSTANT UFLO >= THE UNDERFLOW LIMIT ON THE COMPUTER.
C
C   SUBPROGRAMS CALLED: GAMLOG (LOG OF GAMMA FUNCTION)
C
C   CURRENT VERSION COMPLETED OCTOBER 29, 1986
C
C   REFERENCES:
C
C   1) LAU, CHI-LEUNG, 'A SIMPLE SERIES FOR THE INCOMPLETE GAMMA
C      INTEGRAL', ALGORITHM AS 147, APPLIED STATISTICS, VOL. 29,
C      NO. 1, 1980, PP. 113-114.
C
C   2) REEVE, CHARLES P., 'AN ALGORITHM FOR COMPUTING THE GAMMA C.D.F.
C      TO A SPECIFIED ACCURACY', STATISTICAL ENGINEERING DIVISION
C      NOTE 86-2, OCTOBER 1986.
C-----------------------------------------------------------------------
C   DEFINITION OF PASSED PARAMETERS:
C
C       * X = VALUE AT WHICH THE C.D.F IS TO BE COMPUTED (REAL)
C
C   * ALPHA = PARAMETER OF THE GAMMA FUNCTION (>0) (REAL)
C
C     * EPS = THE DESIRED ABSOLUTE ACCURACY OF THE C.D.F (>0) (REAL)
C
C     IFLAG = ERROR INDICATOR ON OUTPUT (INTEGER)   INTERPRETATION:
C             0 -> NO ERRORS DETECTED
C             1 -> EITHER ALPHA OR EPS IS <= UFLO
C             2 -> NUMBER OF TERMS EVALUATED IN THE INFINITE SERIES
C                  EXCEEDS IMAX.
C
C      CDFX = THE C.D.F. EVALUATED AT X (REAL)
C
C   * INDICATES PARAMETERS REQUIRING INPUT VALUES
C-----------------------------------------------------------------------
C
      LOGICAL LL
      DOUBLE PRECISION DX,GAMLOG
```

```
      DATA IMAX,UFLO / 5000,1.0E-100 /
      CDFX = 0.0
C
C--- CHECK FOR VALIDITY OF ARGUMENTS ALPHA AND EPS
C
      IF (ALPHA.LE.UFLO.OR.EPS.LE.UFLO) THEN
         IFLAG = 1
         RETURN
      ENDIF
      IFLAG = 0
C
C--- CHECK FOR SPECIAL CASE OF X
C
      IF (X.LE.0.0) RETURN
C
C--- EVALUATE THE GAMMA P.D.F. AND CHECK FOR UNDERFLOW
C
      DX = DBLE(X)
      PDFL = SNGL(DBLE(ALPHA-1.0)*DLOG(DX)-DX-GAMLOG(ALPHA))
      IF (PDFL.LT.ALOG(UFLO)) THEN
         IF (X.GE.ALPHA) CDFX = 1.0
      ELSE
         P = ALPHA
         U = EXP(PDFL)
C
C--- DETERMINE WHETHER TO INCREMENT OR DECREMENT ALPHA (A.K.A. P)
C
         LL = .TRUE.
         IF (X.GE.P) THEN
            K = INT(P)
            IF (P.LE.REAL(K)) K = K-1
            ETA = P-REAL(K)
            BL = SNGL(DBLE(ETA-1.0)*DLOG(DX)-DX-GAMLOG(ETA))
            LL = BL.GT.ALOG(EPS)
         ENDIF
         EPSX=EPS/X
         IF (LL) THEN
C
C--- INCREMENT P
C
            DO 10 I = 0, IMAX
               IF (U.LE.EPSX*(P-X)) RETURN
               U = X*U/P
               CDFX = CDFX+U
               P = P+1.0
   10       CONTINUE
            IFLAG = 2
         ELSE
C
C--- DECREMENT P
C
            DO 20 J = 1, K
               P = P-1.0
```

```
            IF (U.LE.EPSX*(X-P)) GO TO 30
            CDFX = CDFX+U
            U = P*U/X
20        CONTINUE
30        CDFX = 1.0-CDFX
       ENDIF
    ENDIF
    RETURN
    END
```