ALGORITHM 403
CIRCULAR INTEGER PARTITIONING [A1]

M. W. COLEMAN AND M. S. TAYLOR (Recd. 30 June 1970)
Aberdeen Proving Ground, MD 21005

KEYWORDS AND PHRASES: partitions, combinatorics, statistical design of experiments
CR CATEGORIES: 5.39, 5.5

DESCRIPTION:

The partition, when expressed as a K-tuple $(X_1, \cdots, X_K)$, may be thought of as a K-digit number in the base V number system. The procedure *CIRPI* then functions as a counter which generates successive K-digit numbers in the base V number system. However, since all K-digit numbers do not correspond to circular partitions, it is possible to have the procedure generate only a subset of K-tuples for consideration, using the following criteria:

(a) The digits are constrained to sum to V, consequently, the K digits are not independent. Thus the procedure need only operate on the $K - 1$ most significant digits, the least significant digit being an easily computable function of the other $K - 1$ digits.

(b) Since the numbers are sequentially increasing, a given number is a cyclic permutation of a previously generated number if a cyclic rotation of its digits produces a number with a smaller value. Thus the most significant digit, X_1, provides an effective minimum value for any of the digits.

(c) Given that the digits must sum to V and the minimum value for any digit is X_1, the value $V - X_1 * (K - 1)$ provides an effective maximum for any digit.

(d) Since the maximum and minimum values depend on the most significant digit, X_1, the procedure is finished when X_1 has increased to the point where the minimum digit size exceeds the maximum digit size, i.e. when $X_1 > V - X_1 * (K - 1)$. This easily reduces to $X_1 > V/K$, providing an easy method for terminating the K-tuple generation as early as possible.

Therefore, the procedure efficiently generates the totality of circular partitions since it can greatly restrict the number of K-tuples that must be considered.

REFERENCES:

1. DAVID, H. A., AND F. W. WOLOCK. Cyclic designs. *Annals of Math. Stat. 36* (1965), 1526–1534.
2. NIVEN, I., *Mathematics of Choice*. Random House, New York, 1965, ch. 6.

ALGORITHM:

```
      SUBROUTINE CIRPI (V, K, X)
C
C   THIS SUBROUTINE GENERATES ALL K-TUPLES SUCH THAT.....
C A) THE SUM OF THE K ELEMENTS OF THF K-TUPLE IS V,
C B) EACH OF THE ELEMENTS IS AN INTEGER GREATER THAN 0, AND
C C) NO K-TUPLE IS A CYCLIC PERMUTATION OF ANY OTHER K-TUPLE.
C THE K-TUPLE IS STORED IN THE ARRAY X, WITH ONE ELEMENT
C PER ARRAY ELEMENT.  EACH K-TUPLE IS PROCESSED BY THE USER
C (USING THE SUBROUTINE 'PROCES') BEFORE THE NEXT K-TUPLE IS
C GENERATED.  THE SUBROUTINE 'PROCES' MUST NOT CHANGE THE
C CONTENTS OF THE ARRAY X.
C
      INTEGER X(K), V, V1, V2, C, SUM
        V1 = V-K+1
        V2 = V/K
        K1 = K-1
        K2 = K-2
        SUM = K1
C
C   INITIALIZE THE ARRAY X WITH THE FIRST K-TUPLE.
C
      DO 100 I = 1, K1
        X(I) = 1
  100 CONTINUE
        GO TO 115
C
C   GENERATE THE NEXT K-TUPLE WHICH SATISFIES THE GIVEN
C CONDITIONS, A) - C).
C
  110   C = 1
        SUM = X(1)
      DO 113 I = 1, K2
        I1 = K-I
        X(I1) = X(I1)+C
        IF (X(I1) .LT. V1) GO TO 111
          X(I1) = X(1)
          GO TO 112
  111   C = 0
  112   SUM = SUM+X(I1)
  113 CONTINUE
        IF (C .EQ. 0) GO TO 115
          X(1) = X(1)+1
          IF (X(1) .GT. V2) RETURN
      DO 114 I1 = 2, K1
        X(I1) = X(1)
  114 CONTINUE
        SUM = X(1)*K1
        V1 = V-SUM
  115   SUM = V-SUM
        IF (SUM .LT. X(1)) GO TO 110
          X(K) = SUM
C
C   CHECK TO SEE IF THE K-TUPLE IS A CYCLIC PERMUTATION OF
C ANY PREVIOUSLY GENERATED K-TUPLES.  IF IT IS, GENERATE THE
C NEXT CANDIDATE, OTHERWISE, CALL THE SUBROUTINE 'PROCES' TO
C PROCESS THE K-TUPLE BEFORE GENERATING THE NEXT ONE.
C
  120 DO 122 I = 2, K
        IF (X(I) .GT. X(1)) GO TO 122



        IF (X(I) .LT. X(1)) GO TO 110
        I1 = I+1
        DO 121 I2 = 2, K
          IF (I1 .GT. K) I1 = I1-K
          IF (X(I1) .GT. X(I2)) GO TO 122
          IF (X(I1) .LT. X(I2)) GO TO 110
          I1 = I1+1
  121   CONTINUE
        GO TO 130
  122 CONTINUE
  130 CALL PROCES (X, K)
        GO TO 110
      END
```

ALGORITHM 404
COMPLEX GAMMA FUNCTION [S14]

C. W. LUCAS JR.* AND C. W. TERRILL (Recd. 13 Feb. 1970 and 19 June 1970)
Physics Department, College of William and Mary in Virginia, Williamsburg, VA 23185

*William and Mary Predoctoral Fellow. This work was partly supported by the National Aeronautics and Space Agency, Contract NGL 47-006008.

KEY WORDS AND PHRASES: gamma function, poles of gamma function, Stirling's asymptotic series, recursion formula, reflection formula
CR CATEGORIES: 5.12

DESCRIPTION:

CGAMMA evaluates in single precision the gamma function for complex arguments. The method of evaluation is similar to the one employed by A. M. S. Filho and G. Schwachheim in evaluating the gamma function with arbitrary precision for real arguments [1]. First the real part of the argument of the gamma function is increased by some integer M, if necessary, so that Stirling's asymptotic series for the logarithm of the gamma function may be used with high precision and a small number of terms. Then the recursion formula for the gamma function

$$\Gamma(Z) = \Gamma(Z + 1)/Z$$

is used to step down to the original gamma function.

The conditions on the value of $T = Z + M$ used in Stirling's asymptotic series are:

1. $Real(T) > 10$
2. $Arg(T) = arctan(Imaginary(T)/Real(T)) \leq \pi/4$

This second condition ensures that the error incurred in using Stirling's asymptotic series with a finite number of terms is less than the value of the next term in the series [2].

The only condition on the argument Z is that it must not be too close to a pole of the gamma function, i.e. $Z = 0, -1, -2, \cdots$. A rough empirical relation was found between the number of significant figures obtained by Stirling's asymptotic series and the distance δ in the complex plane from Z to the nearest pole by approaching the poles at 0 and -1 from several directions. If $\delta = 10^{-n}$ (n an integer ≥ 3) this relation is *(minimum number of significant figures)* $= 7 - n$. With $\delta = 10^{-4}$, for instance, Stirling's asymptotic series gives three or more significant figures depending on the direction of Z from the pole. The upper limit on the size of Z for which *CGAMMA* will work is a function of the computer system. For the IBM 360 system where the largest size number that can be handled is about 10^{75} the upper limit for real Z is about ± 57, for Z on the line $Imaginary\ (Z) = \pm Real(Z)$ it is $(63 \pm 63i)$, for $Real(Z) > 0$ and $(-32 \pm 32i)$ for $Real(Z) < 0$, and for Z on the imaginary axis it is $\pm 107i$.

CGAMMA has been tested in several ways. The reflection formula

$$\Gamma(Z)\Gamma(1 - Z) = \frac{\pi}{sin(\pi Z)}$$

and the relation

$$\Gamma(n + 1) = n! \quad (n \text{ integer})$$

have been employed as checks. Also $log(gamma(Z))$ has been compared with tabulated valued in reference [2] for a number of values of Z. These tests lead us to conclude that *CGAMMA* gives four to five significant figures for Z outside disks of radius $\delta = 10^{-3}$ centered on the poles. If the subroutine is written in double precision, we have found that about eight more significant figures will be obtained everywhere for an IBM 360 system, and near the poles

$$(minimum\ number\ of\ significant\ figures) = 15 - n$$

where $\delta = 10^{-n}$. The range of the subroutine remains the same.

Acknowledgment. The authors wish to express their gratitude to Dr. Morton Eckhause of the William and Mary Physics Department for his continuing advice and to Shirley McCallum and Samuel Pettus Hoyle of the College of William and Mary Computer Center who frequently reduced the programming difficulties.

References:

1. Filho, Antonina Machado Souza and Schwachheim, Georges. Algorithm 309, Gamma function of arbitrary precision. *Comm. ACM 10* (Aug. 1967), 511.
2. US Dep. of Commerce, Amer. Nat. Stand. Inst. Table of the gamma function for complex arguments. Clearinghouse, Springfield, VA 22151 (1954), p. VIII.

Algorithm:

[*Warning.* System dependent constants are used in assigning values to IOUT, PI, TOL, SUM—L.D.F.]

```
      FUNCTION CGAMMA(Z)
      COMPLEX Z,ZM,T,TT,SUM,TERM,DEN,CGAMMA,PI,A
      DIMENSION C(12)
      LOGICAL REFLEK
C SET IOUT FOR PROPER OUTPUT CHANNEL OF COMPUTER SYSTEM FOR
C ERROR MESSAGES
      IOUT = 3
      PI = (3.141593,0.0)
      X = REAL(Z)
      Y = AIMAG(Z)
C TOL = LIMIT OF PRECISION OF COMPUTER SYSTEM IN SINGLE PRECISION
      TOL = 1.0E-7
      REFLEK = .TRUE.
C DETERMINE WHETHER Z IS TOO CLOSE TO A POLE
C CHECK WHETHER TOO CLOSE TO ORIGIN
      IF(X.GE.TOL) GO TO 20
C FIND THE NEAREST POLE AND COMPUTE DISTANCE TO IT
      XDIST = X-INT(X-.5)
      ZM = CMPLX(XDIST,Y)
      IF(CABS(ZM).GE.TOL) GO TO 10
C IF Z IS TOO CLOSE TO A POLE, PRINT ERROR MESSAGE AND RETURN
C WITH CGAMMA = (1.E7,0.0E0)
      WRITE(IOUT,900) Z
      CGAMMA = (1.E7,0.E0)
      RETURN
C FOR REAL(Z) NEGATIVE EMPLOY THE REFLECTION FORMULA
C                 GAMMA(Z) = PI/(SIN(PI*Z)*GAMMA(1-Z))
C AND COMPUTE GAMMA(1-Z).  NOTE REFLEK IS A TAG TO INDICATE THAT
C THIS RELATION MUST BE USED LATER.
10    IF(X.GE.0.0) GO TO 20
      REFLEK = .FALSE.
      Z = (1.0,0.0)-Z
      X = 1.0-X
      Y = -Y
C IF Z IS NOT TOO CLOSE TO A POLE, MAKE REAL(Z)>10 AND ARG(Z)<PI/4
20    M = 0
40    IF(X.GE.10.) GO TO 50
      X = X + 1.0
      M = M + 1
      GO TO 40
50    IF(ABS(Y).LT.X) GO TO 60
      X = X + 1.0
      M = M + 1
      GO TO 50
60    T = CMPLX(X,Y)
      TT = T*T
      DEN = T
C COEFFICIENTS IN STIRLING'S APPROXIMATION FOR LN(GAMMA(T))
      C(1) = 1./12.
      C(2) = -1./360.
      C(3) = 1./1260.
      C(4) = -1./1680.
      C(5) = 1./1188.
      C(6) = -691./360360.
      C(7) = 1./156.
      C(8) = -3617./122400.
      C(9) = 43867./244188.
      C(10) = -174611./125400.
      C(11) = 77683./5796.
      SUM = (T-(.5,0.0))*CLOG(T)-T+CMPLX(.5*ALOG(2.*3.14159),0.0)
      J = 1
70    TERM = C(J)/DEN
C TEST REAL AND IMAGINARY PARTS OF LN(GAMMA(Z)) SEPARATELY FOR
C CONVERGENCE.  IF Z IS REAL SKIP IMAGINARY PART OF CHECK.
      IF(ABS(REAL(TERM)/REAL(SUM)).GE.TOL) GO TO 80
      IF(Y.EQ.0.0) GO TO 100
      IF(ABS(AIMAG(TERM)/AIMAG(SUM)).LT.TOL) GO TO 100
80    SUM = SUM + TERM
      J = J + 1
      DEN = DEN*TT
C TEST FOR NONCONVERGENCE
      IF(J.EQ.12) GO TO 90
      GO TO 70
C STIRLING'S SERIES DID NOT CONVERGE.  PRINT ERROR MESSAGE AND
C PROCEDE.
90    WRITE(IOUT,910) Z
C RECURSION RELATION USED TO OBTAIN LN(GAMMA(Z))
C                LN(GAMMA(Z)) = LN(GAMMA(Z+M)/(Z*(Z+1)*...*(Z+M-1)))
C                             = LN(GAMMA(Z+M)-LN(Z)-LN(Z+1)-...-LN(Z+M-1)
100   IF(M.EQ.0) GO TO 120
      DO 110 I = 1,M
      A = CMPLX(I*1.-1.,0.0)
110   SUM = SUM-CLOG(Z+A)
C CHECK TO SEE IF REFLECTION FORMULA SHOULD BE USED
120   IF(REFLEK) GO TO 130
      SUM = CLOG(PI/CSIN(PI*Z))-SUM
      Z = (1.0,0.0) -Z
130   CGAMMA = CEXP(SUM)
      RETURN
900   FORMAT(1X,2E14.7,10X,49HARGUMENT OF GAMMA FUNCTION IS TOO CLOSE TO
     1 A POLE)
910   FORMAT(44H ERROR - STIRLING'S SERIES HAS NOT CONVERGED/14X,4HZ = ,
     12E14.7)
      END
```