Algorithm 736: Hyperelliptic Integrals and the Surface Measure of Ellipsoids

CHARLES F. DUNKL and DONALD E. RAMIREZ
University of Virginia

The algorithm for computing a class of hyperelliptic integrals and for determining the surface measure of ellipsoids is described in detail by Dunkl and Ramirez [1994]. An efficient implementation of their algorithm is presented here.

Categories and Subject Descriptors: G.1.4 [**Numerical Analysis**]: Quadrature and Numerical Differentiation—*multiple quadrature*; G.3 [**Mathematics and Computing**]: Probability and Statistics

General Terms: Algorithms

Additional Key Words and Phrases: Elliptic integral, expected radius, Lauricella's hypergeometric function, surface measure

1. DESCRIPTION AND PURPOSE

The algorithm for computing a class of hyperelliptic integrals and for determining the surface measure of ellipsoids is described in detail by Dunkl and Ramirez [1994]. An efficient implementation of their algorithm is presented here.

For a positive definite matrix $\sum$ with eigenvalues $\gamma_1 \geq \gamma_2 \geq \cdots \geq \gamma_n > 0$, the expected value $\mathscr{E}((X' \sum X)^{1/2}) \equiv |||\sum|||$ of the square root of the quadratic form $X' \sum X$, where X is the uniformly distributed variable on the unit sphere in $\mathbb{R}^n$, is useful in statistics and multivariate analysis. This expectation can be represented as a type of hyperelliptic integral.

When $n = 2$, the value of $|||\sum|||$ can be expressed as a complete elliptic integral of the second kind by

$$|||\sum||| = \frac{2}{\pi}\sqrt{\gamma_1}\,E\left(k, \frac{\pi}{2}\right) \qquad \text{with} \quad k^2 = 1 - \frac{\gamma_2}{\gamma_1}.$$

C. F. Dunkl's research was partially supported by NSF Grant DMS-88-02400. D. E. Ramirez's research was partially supported by the Center for Advanced Studies at the University of Virginia.
Authors' address: Department of Mathematics, University of Virginia, Charlottesville, VA 22903.

Accordingly, we shall call $|||\sum|||$ the *multivariate elliptic integral* for general n. In hyperspherical coordinates, it has the form

$$\sigma_n^{-1}\int_0^{2\pi}\int_0^{\pi}\cdots\int_0^{\pi}(\gamma_1 u_1^2+\cdots+\gamma_{n-1}u_{n-1}^2+\gamma_n u_n^2)^{1/2} \cdot(\sin\omega_1)^{n-2}(\sin\omega_2)^{n-3}\cdots(\sin\omega_{n-2})\,d\omega_1\cdots d\omega_{n-2}\,d\omega_{n-1}$$

with

$$\begin{aligned} u_1 &= \cos\omega_1\\ u_2 &= \sin\omega_1\cos\omega_2\\ &\vdots\\ u_{n-1} &= \sin\omega_1\cdots\sin\omega_{n-2}\cos\omega_{n-1}\\ u_n &= \sin\omega_1\cdots\sin\omega_{n-2}\sin\omega_{n-1} \end{aligned} \tag{1}$$

($0\le\omega_i\le\pi$, $1\le i\le n-2$, and $0\le\omega_{n-1}<2\pi$) and with σ_n from formula (3).

In Dunkl and Ramirez [1994], it is noted that the above integral is a special case of Lauricella's hypergeometric function F_D. The results of Carlson [1963, p. 466] are applied to show that the multivariate elliptic integral can be expressed as a *single* variable integral, and so standard univariate integration methods can be applied. We chose to evaluate numerically the single variable integral by the Romberg algorithm of Dunkl [1962] (see, e.g., Davis and Rabinowitz [1984, p. 493]). We have implemented a modification of the Kahan [1980] substitution to provide for faster convergence (see, e.g., Davis and Rabinowitz [1984, p. 441]). We have found that the higher-order zero provided by $x=5u^4-4u^5$ is efficient for an automatic algorithm that tries to achieve specified precision.

We next address the long-standing problem of efficiently computing the surface measure of an ellipsoid. Let $0<\delta_1\le\delta_2\le\cdots\le\delta_n$ be the semiaxes for the ellipsoid $(x_1/\delta_1)^2+(x_2/\delta_2)^2+\cdots+(x_n/\delta_n)^2=1$ in $\mathbb{R}^n$. Dunkl and Ramirez [1994] have shown the relationship between the multivariate elliptic integral and the surface measure $\mathcal{S}(\delta_1,\ldots,\delta_n)$ of the ellipsoid to be given by

$$\mathcal{S}(\delta_1,\ldots,\delta_n)=\sigma_n\left(\prod_{i=1}^n\delta_i\right)|||\textstyle\sum|||, \tag{2}$$

with $\sum=\mathrm{diag}(1/\delta_1^2,\ldots,1/\delta_n^2)$ and

$$\sigma_n=\frac{2\pi^{n/2}}{\Gamma\left(\frac{n}{2}\right)}, \tag{3}$$

the surface measure of the unit sphere in $\mathbb{R}^n$. We put this forward as a modern and efficient algorithm for computing surface measures of ellipsoids.

Bounds for $|||\sum|||$ are

$$\frac{1}{n}\sum_{i=1}^{n}(\gamma_i)^{1/2} \le |||\sum||| \le \left(\frac{1}{n}\sum_{i=1}^{n}\gamma_i\right)^{1/2}, \tag{4}$$

which carry over to the surface measure formula (2). The call

CALL ELLPTI (NDIM, MAXDIM, GAMMA, ERRTOL, RESULT, NX, WORK, IER)

returns the computed multivariate elliptic integral (1) as RESULT(1) $\pm$ RESULT(4). The surface measure (2) is RESULT(5) $\pm$ RESULT(6). The lower and upper bounds (4) are returned in RESULT(2) and RESULT(3), respectively. The number of functions evaluations used is NX.

The internal INTEGER parameter MAXIT is set equal to 14. This controls the maximum number of function evaluations, $< 2^{14} = 16{,}384$.

The values for the surface measure (roughly, $\prod_{i=1}^{n}\delta_i$ in magnitude) in practice can be so large as to cause overflow in the REAL mode. The recommended implementation is DOUBLE PRECISION mode (i.e., 8 bytes) with relative error tolerance 1.0D-10. To convert the routines to double precision, the following changes should be made:

(1) Change REAL to DOUBLE PRECISION in each routine and

(2) change the real constants in the DATA statements to double-precision constants.

```
      SUBROUTINE ELLPTI(NDIM, MAXDIM, GAMMA, ERRTOL, RESULT, NX, WORK,
     +                  IER)
C
C     PROGRAM FINDS INTEGRAL OVER THE UNIT SPHERE IN R**N
C     OF SQRT(GAMMA(1)*X(1)**2+...+GAMMA(N)*X(N)**2)
C     THE EXPECTED VALUE OF SQRT(X''*A*X)
C     WHERE A HAS EIGENVALUES GAMMA(1),...,GAMMA(N) AND
C     X IS UNIFORMLY DISTRIBUTED ON THE SPHERE
C
C     EQUIVALENTLY, PROGRAM FINDS THE EXPECTED RADIUS
C     OF THE ELLIPSOID X'*B**(-1)*X = 1
C     WHERE B IS DIAG(DELTA(1)**2,...,DELTA(N)**2)
C     WITH AXES DELTA(1),...,DELTA(N)'
C     WHERE GAMMA(I)=1/DELTA(I)**2, 1<=I<=N
C
C     VERSION - 10/15/91
C
C     FORMAL PARAMETERS
C        NDIM     INTEGER          input:  the number of values in GAMMA.
C        MAXDIM   INTEGER          input:  the dimension of GAMMA in the
C                                          main program.
C
C        GAMMA    REAL array(*)    input:  the values of GAMMA.
C
C        ERRTOL   REAL             input:  on input the user's requested
C                                    &     relative error tolerance, and
C                                  output: on output the error tolerance
C                                          used by the program.
C
```

```
C       RESULT  REAL array(6)   output: RESULT(1) is the computed
C                                       multivariate elliptic integral,
C                                       RESULT(2) is the lower bound
C                                       estimate for the integral,
C                                       RESULT(3) is the upper bound
C                                       estimate for the integral,
C                                       RESULT(4) is the error estimate
C                                       for the integral,
C                                       RESULT(5) is the computed
C                                       surface measure, and
C                                       RESULT(6) is the error estimate
C                                       for the surface measure.
C
C       NX      INTEGER         output: number of function evaluations
C                                       used.
C
C       WORK    REAL array      input:  work space
C               (2*MAXDIM)
C
C       IER     INTEGER         output: the error flag.  See failure
C                                       indications below.
C
C    FAILURE INDICATIONS
C       If IER = 0, then no error was detected and successful
C          convergence was obtained.
C
C       If IER = 1, then the program did not converge to the
C          required tolerance.  The last value for the integral along
C          with the estimated error are reported, allowing the user
C          to evaluate the utility of the results.
C

      If IER = 2, then at least one value in GAMMA is
         nonpositive.  All nonpositive values are set equal to zero.
         This is a warning that the surface measure is undefined and
         RESULT(5) and RESULT(6) are set to zero.

      If IER = 3, then both of the above conditions for IER = 1
         and IER = 2 hold.

      If IER = 4, then there are no positive values in GAMMA and
         the program terminates.

      If IER = 5, then the dimension of GAMMA is incorrect and
         the program terminates.


     CONSTANTS
        SMALL is set to be 2**(-18) in REAL mode.  This should be
           changed to 2**(-40) for DOUBLE PRECISION mode.  These
           values are about 64 times machine epsilon.


     GLOBAL VARIABLES
     INTEGER IER, MAXDIM, NDIM, NX

     $$$ CHOOSE MODE $$$
     DOUBLE PRECISION ERRTOL, GAMMA(*), RESULT(6), WORK(2,*)
     REAL             ERRTOL, GAMMA(*), RESULT(6), WORK(2,*)
     LOCAL VARIABLES
     INTEGER MAXIT
     CONTROLS NUMBER OF FUNCTION EVALUTIONS
     PARAMETER (MAXIT = 14)

     $$$ CHOOSE MODE $$$
     DOUBLE PRECISION BETAIN, C10, C20, ERRQF, FACTOR, FACVAL,
```

```
      REAL               BETAIN, C10, C20, ERRQF, FACTOR, FACVAL,
     +                   FIVE, FOUR, FX, F4, H, INTGRL, LOWQF, MAXGAM,
     +                   ONE, PROD, Q(MAXIT), Q0, SMALL,
     +                   SUM, THREE, TRAP, TWO, UPQF, V, X,
     +                   XSUM, X3, ZERO
      INTEGER I, IMAX, IP1, J, N
      INTRINSIC ABS, MAX, SQRT
      EXTERNAL BETAIN, BOUNDS, FACTVL

      USE SMALL = 2**(-40) FOR DOUBLE PRECISION MODE
      USE SMALL = 2**(-18) FOR REAL MODE

      $$$ CHOOSE MODE $$$
      DATA SMALL / 9.094947017729D-13 /
      DATA ZERO, ONE, TWO, THREE, FOUR, FIVE, C10, C20 / 0.0D0, 1.0D0,
     +     2.0D0, 3.0D0, 4.0D0, 5.0D0, 10.0D0, 20.0D0 /
      DATA SMALL / 3.8146973E-06 /
      DATA ZERO, ONE, TWO, THREE, FOUR, FIVE, C10, C20 / 0.0E0, 1.0E0,
     +     2.0E0, 3.0E0, 4.0E0, 5.0E0, 10.0E0, 20.0E0 /

      START OF EXECUTABLE CODE

      INITIALIZE IER = 0
      IER = 0

      CHECK BOUNDS OF NDIM - ABORT IF OUT OF BOUNDS

      IF ((NDIM.LT.2).OR.(NDIM.GT.MAXDIM)) THEN
         IER = 5
         RETURN
      ENDIF
      MAXGAM = ZERO
      IMAX = 0
      DO 100, I=1,NDIM
C
C     CHECK VALUES OF GAMMA TO BE POSITIVE - WARNING ONLY
         IF (GAMMA(I).LE.ZERO) THEN
            IER = 2
            GAMMA(I) = ZERO
         ENDIF
C
C     MAKE GAMMA(1) THE MAXIMUM OF THE VALUES OF GAMMA
         IF (GAMMA(I).GT.MAXGAM) THEN
            MAXGAM = GAMMA(I)
            IMAX = I
         ENDIF
 100  CONTINUE
C     CHECK THAT AT LEAST ONE GAMMA VALUE IS POSITIVE - ABORT IF NOT
      IF (IMAX.EQ.0) THEN
         IER = 4
         RETURN
      ENDIF
      GAMMA(IMAX) = GAMMA(1)

      GAMMA(1) = MAXGAM
      DO 110, I=1,NDIM
         WORK(1,I) = (MAXGAM-GAMMA(I))/MAXGAM
C     WORK(2,I) = 1 - WORK(1,I)
         WORK(2,I) = GAMMA(I)/MAXGAM
 110  CONTINUE
C
C     NOTE: WORK(1,1) = ZERO
C     NOTE: WORK(2,1) = ONE
      FACTOR = C20*SQRT(MAXGAM)*BETAIN(NDIM)/NDIM
      ERRTOL = MAX(ERRTOL,SMALL)
      H = ONE
```

```
C
C     EVALUATE FUNCTION AT ONE
      FX = ONE
C     START I = 2 SINCE WORK(2,1) = ONE
      DO 120, I=2,NDIM
         FX = FX+WORK(2,I)
 120  CONTINUE
      FX = FX/SQRT(C10)
C
C     FUNCTION IS ZERO AT ZERO
      TRAP = FX/TWO
      NX = 1
C
C     LOOP FOR ROMBERG INTEGRATION
C     REFERENCE IS
C     DUNKL, C. F. (1962), ROMBERG QUADRATURE TO PRESCRIBED ACCURACY,
C        SHARE FILE NUMBER 7090-1481
      DO 130, N=1,MAXIT
         H = H/TWO
         SUM = ZERO
         NX = NX*2
         DO 140, J=1,NX-1,2
            X = J*H
C
C     IMPLEMENT MODIFICATION OF KAHAN SUBSTITUTION
C     W. M. KAHAN, "HANDHELD CALCULATOR EVALUATES INTEGRALS,"
C        HEWLETT-PACKARD JOURNAL,AUGUST 1980.
C        SET V = 5*X**4 - 4*X**5 TO REMOVE SINGULARITIES AT ZERO AND ONE
            X3 = X**3
            V = (FIVE-FOUR*X)*X*X3
            PROD = ONE/(X*(X*(FOUR*X+THREE)+TWO)+ONE)
            XSUM = ONE
            DO 150, I=2,NDIM
               FX = WORK(2,I)+V*WORK(1,I)
               XSUM = XSUM+(WORK(2,I)/FX)
               PROD = PROD*(V/FX)
 150        CONTINUE
            FX = X3*XSUM*SQRT(PROD)
            SUM = SUM+FX
 140     CONTINUE
         SUM = SUM*H
         TRAP = SUM+TRAP/TWO
         Q(N) = TWO*(TRAP+SUM)/THREE
         Q0 = Q(1)
         IF (N.GT.1) THEN
            F4 = FOUR
            DO 160, I=N-1,1,-1
               F4 = F4*FOUR
               IP1 = I+1
               Q(I) = Q(IP1)+(Q(IP1)-Q(I))/(F4-ONE)
 160        CONTINUE
            ERRQF = ABS(Q(1)-Q0)
            IF (ERRQF.LE.(ERRTOL*Q0)) GOTO 200
         ENDIF
 130  CONTINUE
C
C     PROGRAM DID NOT CONVERGE TO REQUIRED TOLERANCE - WARNING ONLY
      IER = IER+1
C
C     SUCCESSFUL CONVERGENCE
 200  CONTINUE
      INTGRL = FACTOR*Q(1)
      ERRQF = FACTOR*ERRQF
C
C     COMPUTE BOUNDS FOR INTEGRAL
      CALL BOUNDS(NDIM, GAMMA, LOWQF, UPQF)
C
```

```
C     PASS RESULTS BACK IN VECTOR RESULT(6)
      RESULT(1) = INTGRL
      RESULT(2) = LOWQF
      RESULT(3) = UPQF
      RESULT(4) = ERRQF
C
C     COMPUTE FACVAL = (PRODUCT OF DELTA'S)*(SURFACE MEASURE OF SPHERE)
C     AND SURFACE MEASURE ONLY IF ALL GAMMA VALUES ARE POSITIVE
      IF (IER.LE.1) THEN
         CALL FACTVL(NDIM, GAMMA, FACVAL)
         RESULT(5) = FACVAL*INTGRL
C
C     COMPUTE ERROR ESTIMATE FOR SURFACE MEASURE
         RESULT(6) = FACVAL*ERRQF

      ELSE
         RESULT(5) = ZERO
         RESULT(6) = ZERO
      ENDIF
C     EXIT
      RETURN
      END
***********************************************************************
      FUNCTION BETAIN(N)
C     COMPUTES BETA(1/2,(N+1)/2)**(-1)
C
C     GLOBAL VARIABLES
C
C     $$$ CHOOSE MODE $$$
C     DOUBLE PRECISION BETAIN
      REAL             BETAIN
      INTEGER N
C
C     LOCAL VARIABLES
      INTEGER I
C
C     $$$ CHOOSE MODE $$$
C     DOUBLE PRECISION HALF, PI, ONE
      REAL             HALF, PI, ONE
      INTRINSIC MOD
C
C     $$$ CHOOSE MODE $$$
C     DATA HALF, ONE, PI / 0.5D0, 1.0D0, 3.1415926535897932D0 /
      DATA HALF, ONE, PI / 0.5E0, 1.0E0, 3.1415926535897932E0 /
C
C     START OF EXECUTABLE CODE
C
C     N ODD
      IF (MOD(N,2).NE.0) THEN
         BETAIN = HALF
         DO 100, I=2,N-1,2
            BETAIN = (BETAIN*(I+1))/I
 100     CONTINUE
C
C     N EVEN
      ELSE
         BETAIN = ONE/PI
         DO 110, I=2,N,2
            BETAIN = (BETAIN*I)/(I-1)
 110     CONTINUE
      ENDIF
      RETURN
      END
***********************************************************************
      SUBROUTINE BOUNDS(NDIM, GAMMA, LOWQF, UPQF)
      INTEGER NDIM, I
C
C     $$$ CHOOSE MODE $$$
```

```
C     DOUBLE PRECISION GAMMA(*), LOWQF, UPQF, ZERO
      REAL             GAMMA(*), LOWQF, UPQF, ZERO
      INTRINSIC SQRT
C
C     $$$ CHOOSE MODE $$$
C     DATA ZERO / 0.0D0 /
      DATA ZERO / 0.0E0 /
C
C     START OF EXECUTABLE CODE
C
      LOWQF = ZERO
      UPQF = ZERO
      DO 100, I=1,NDIM
         LOWQF = LOWQF+SQRT(GAMMA(I))
         UPQF = UPQF+GAMMA(I)
 100  CONTINUE
      LOWQF = LOWQF/NDIM
      UPQF = SQRT(UPQF/NDIM)
      RETURN
      END
***********************************************************************
      SUBROUTINE FACTVL(NDIM, GAMMA, FACVAL)
      INTEGER I, NDIM
C
C     $$$ CHOOSE MODE $$$
C     DOUBLE PRECISION GAMMA(*), FACVAL, PI, PROD, TWO
      REAL             GAMMA(*), FACVAL, PI, PROD, TWO
      INTRINSIC MOD, SQRT
C
C     $$$ CHOOSE MODE $$$
C     DATA PI, TWO / 3.1415926535897932D0, 2.0D0 /
      DATA PI, TWO / 3.1415926535897932E0, 2.0E0 /
C
C     START OF EXECUTABLE CODE
C
C     NDIM ODD
      IF (MOD(NDIM,2).NE.0) THEN
         PROD = TWO
         DO 100, I=1,NDIM-1,2
             PROD = PROD*TWO*PI/I
 100     CONTINUE
C     NDIM EVEN
      ELSE
         PROD = TWO*PI
         DO 110, I=1,NDIM/2-1
            PROD = PROD*PI/I
 110     CONTINUE
      ENDIF
      FACVAL = GAMMA(1)
      DO 120, I=2,NDIM
         FACVAL = FACVAL*GAMMA(I)
 120  CONTINUE
      FACVAL = PROD/SQRT(FACVAL)
      RETURN
      END
```

REFERENCES

CARLSON, B. C. 1963. Lauricella's hypergeometric function F_D. *J. Math. Anal. Appl.* 7, 3, 452–470.

DAVIS, P. J. AND RABINOWITZ, P 1984. *Methods of Numerical Integration*. 2nd ed. Academic Press, Orlando, Fla.

DUNKL, C. F. 1962. Romberg quadrature to prescribed accuracy. SHARE File 7090-1481, TYQUAD.

DUNKL, C. F. AND RAMIREZ, D. E. 1994. Computing hyperelliptic integrals for surface measure of ellipsoids *ACM Trans. Math. Softw. 20*, 4. This issue.

KAHAN, W. M. 1980. Handheld calculator evaluates integrals. *Hewlett-Packard J. 31*, 8, 23–32.

Received October 1991; revised October 1992; accepted November 1992