

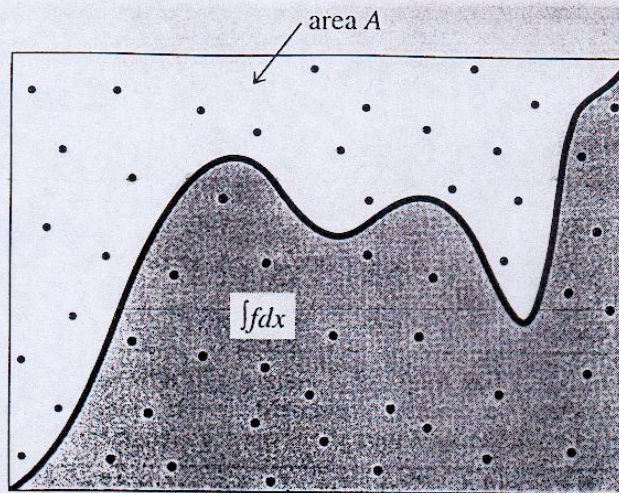


Figure 7.6.1. Monte Carlo integration. Random points are chosen within the area A . The integral of the function f is estimated as the area of A multiplied by the fraction of random points that fall below the curve f . Refinements on this procedure can improve the accuracy of the method; see text.

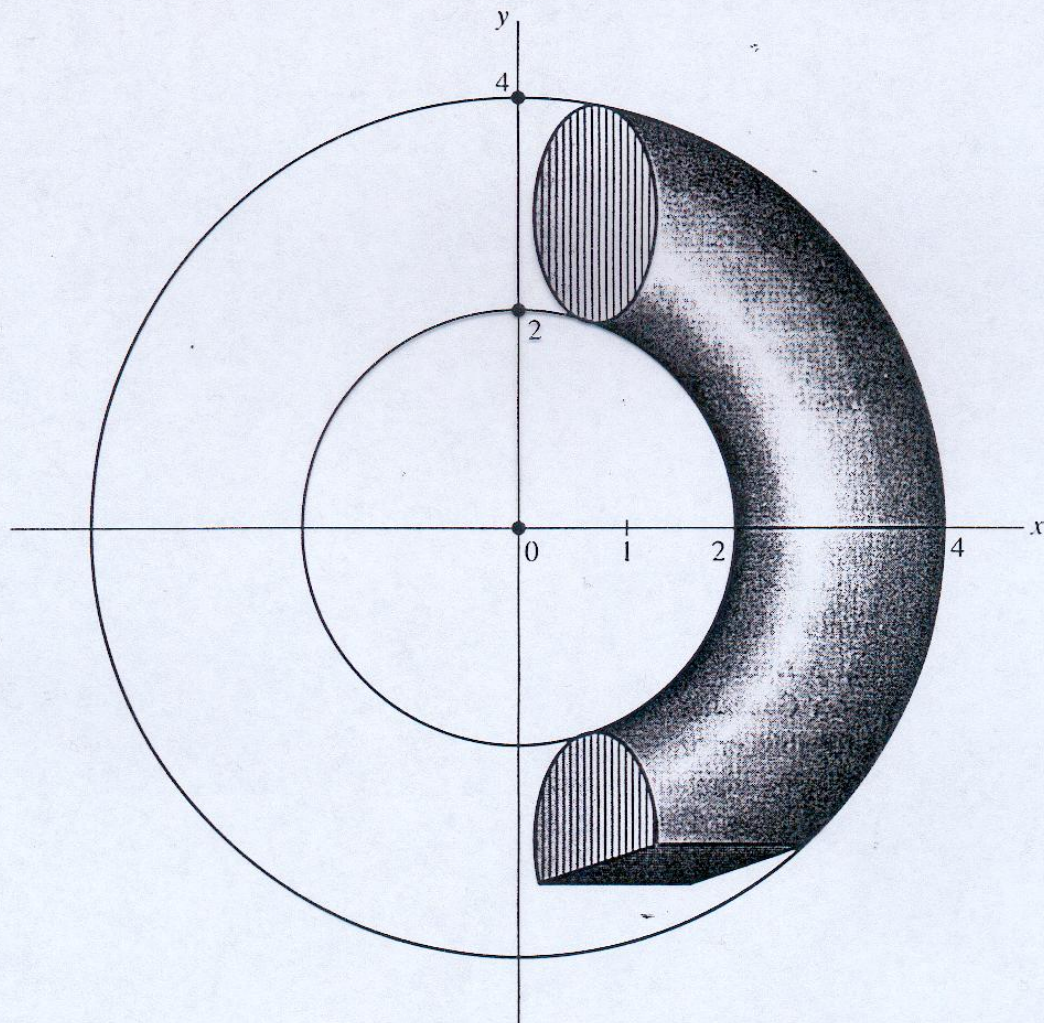


Figure 7.6.2. Example of Monte Carlo integration (see text). The region of interest is a piece of a torus, bounded by the intersection of two planes. The limits of integration of the region cannot easily be written in analytically closed form, so Monte Carlo is a useful technique.

```

FUNCTION ran2(idum)
  INTEGER idum,IM1,IM2,IMM1,IA1,IA2,IQ1,IQ2,IR1,IR2,NTAB,NDIV
  REAL ran2,AM,EPS,RNMX
  PARAMETER (IM1=2147483563,IM2=2147483399,AM=1./IM1,IMM1=IM1-1,
*           IA1=40014,IA2=40692,IQ1=53668,IQ2=52774,IR1=12211,
*           IR2=3791,NTAB=32,NDIV=1+IMM1/NTAB,EPS=1.2e-7,RNMX=1.-EPS)
  Long period ( $> 2 \times 10^{18}$ ) random number generator of L'Ecuyer with Bays-Durham shuffle
  and added safeguards. Returns a uniform random deviate between 0.0 and 1.0 (exclusive
  of the endpoint values). Call with idum a negative integer to initialize; thereafter, do not
  alter idum between successive deviates in a sequence. RNMX should approximate the largest
  floating value that is less than 1.
  INTEGER idum2,j,k,iv(NTAB),iy
  SAVE iv,iy,idum2
  DATA idum2/123456789/, iv/NTAB*0/, iy/0/
  if (idum.le.0) then
    idum=max(-idum,1)
    idum2=idum
    do 11 j=NTAB+8,1,-1
      k=idum/IQ1
      idum=IA1*(idum-k*IQ1)-k*IR1
      if (idum.lt.0) idum=idum+IM1
      if (j.le.NTAB) iv(j)=idum
    enddo 11
    iy=iv(1)
  endif
  k=idum/IQ1
  idum=IA1*(idum-k*IQ1)-k*IR1
  if (idum.lt.0) idum=idum+IM1
  k=idum2/IQ2
  idum2=IA2*(idum2-k*IQ2)-k*IR2
  if (idum2.lt.0) idum2=idum2+IM2
  j=1+iy/NDIV
  iy=iv(j)-idum2
  iv(j)=idum
  if(iy.lt.1)iy=iy+IMM1
  ran2=min(AM*iy,RNMX)
  return
END

```

Initialize.
Be sure to prevent idum = 0.

Load the shuffle table (after 8 warm-ups).

Start here when not initializing.
Compute idum=mod(IA1*idum,IM1) without overflows by Schrage's method.

Compute idum2=mod(IA2*idum2,IM2) likewise.

Will be in the range 1:NTAB.
Here idum is shuffled, idum and idum2 are combined to generate output.

Because users don't expect endpoint values.

C Demonstration der Bedienung der Routinen DATE_AND_TIME und
C RAN2

C

IMPLICIT NONE
REAL ZUFALL, RAN2
INTEGER I, IDUM, IDATE(8)
CHARACTER*12 DATE, TIME, TZONE

CALL DATE_AND_TIME (DATE, TIME, TZONE, IDATE)

C DATE: Datum (Characterform)

C TIME: Uhrzeit (Characterform)

C TZONE: Zeitunterschied zu UT (Characterform)

C IDATE(1): Jahr

C IDATE(2): Monat

C IDATE(3): Tag

C IDATE(4): Zeitdifferenz zu UT in Min.

C IDATE(5): Stunde der Uhrzeit

C IDATE(6): Minute

C IDATE(7): Sekunde

C IDATE(8): Millisekunden

C-----Erzeugen eines "zufaelligen" Startwertes

IDUM = -(60000 * IDATE(6) +1000*IDATE(7) +IDATE(8))

C-----und Initialisieren

RAND = RAN2(IDUM)

C-----Erzeugt ein Feld mit Zufallszahlen

SUM = 0.

DO I = 1, 1000

ZUFALL(I) = RAN2(IDUM)

END DO