This FORTRAN program is used to model diffraction through an infinite slit and is based on modified Huygen-Fresnel principle. The phase of an energy bundle upon its arrival to the observation screen is proportional to its optical length, and is summed with the phase of the others arriving at the same "bin" of the observation screen. The final sum of these phases at a is squared to provide the normalized intensity at that bin. Plotting this intensity of all bins yields a diffraction pattern. The subroutine rmarin is not provided here, as it can be found in Appendix A. The following defines the variables used in this program.

* NY is an array that divides the observation screen into strips. Each element in this array serves as a "bin" which keeps track of the number of bundles reaching that strip. This array is never actually used in the determination of the intensity pattern, but serves to verify that all of the bins are receiving the same number of energy bundles.
* LAMDA is the wavelength of the entering monochromatic radiation.
* W is the width of the rectangular slit.
* Z is the distance from the aperture to the observation screen.
* R is the change in vertical location of the entering energy bundle.
* DIST is the net distance traveled by energy bundle from aperture before being intercepted by observation screen.
* YO is the coordinate of the left hand bottom corner of the rectangular aperture.
* Y is the coordinate of the entering energy bundle in the plane containing the aperture.
* YSCREEN is the coordinate of the energy bundle when it strikes the observation screen. Note that the YSCREEN coordinate is such that YSCREEN=0 occurs at the aperture center.
* YSCMIN, YSCMAX determine the minimum and maximum values of Y for which the "number of energy bundles striking" at the observation screen will be recorded.
* H is the number of strips into which the observation screen will be divided.
* INCREM is the width of each of the strips on the observation screen.
* RANMAR calls a random number from the subroutine rmarin(ij, kl).
* K is the wavenumber, given by 2*PI/LAMDA.
* YDIV is the variable used to determine the "bin" into which to store energy bundles incident to the observation screen.

```
      PROGRAM SLITMONTECARLO
      dimension ny(10000), beta(10000)

      double precision lamda, w, z, y, r, yo, ranmar, pi, phidiff, yscreen, beta, k, yscmax, increm, dist,
     *     phimin, phimax, delphi,betaold,yscmin,obf
      integer n,ydiv,h,numrays,q,I,j
      open(01,file='out.dat',status='old')

      data lamda,w,z,pi/0.00000058, 0.0003,16.0,2.0,3.14159/
      yo=-w/2.0d0

*     Note that h must be evenly divisible by 2
      h=10000

*     Limit range of interest; specifying min/max y of interest.
      yscmax=0.009d0
      yscmin=-yscmax

      numrays=100000
      k=2.0d0*pi/lamda
```

```
*     Initialize the values in the matrix containing the number of energy bundles arriving at screen in a certain strip.
      do 3 j=1,h
       ny(j)=0.0d0
       beta(j)=0.0d0
3     continue

*     Initialize random number generator.
      i=1
      j=3
      increm=(yscmax-yscmin)/h
      call rmarin(i,j)


      do 5 q=1,numrays
*     Calculate the random point of entry of current energy bundle.  Note that the following restricts the direction of
*     emission so that all energy bundles will arrive at the observation screen, but maintains a diffuse emission where all
*     diffraction angles within the limited range will be equally probable.

       y=yo+ranmar()*w
       phimax=datan((yscmax)/z)
       phidiff=-phimax+2*phimax*ranmar()
       r=dtan(phidiff)*z
       yscreen=y+r
       dist=dsqrt(z**2+r**2)

*     Increment the number falling into a bin, keeping with the numbering scheme.
      if(abs(yscreen).le.yscmax)then
       if(yscreen.lt.0.0d0)then
       ydiv=int(abs(yscreen/increm))+1
       ydiv=(h/2)-ydiv+1
       else if (yscreen.ge.0.0d0) then
       ydiv=int(yscreen/increm)+1
       ydiv=ydiv+(h/2)
      end if
      ny(ydiv)=ny(ydiv)+1
*     Increment the phase of the appropriate bin.
      betaold=beta(ydiv)

*     Define the obliquity factor.
      obf=0.5d0*(1.0d0+dcos(phidiff))

*     Note that here dcos(2d0*dist*pi/lamda)) is used instead of dsin (2d0*dist*pi/lamda)), as defined by equation 4.33.
*     Either provide similar results, whereby the peaks fall within the envelope of the analytical solution, however use of
*     cos results in a pattern with the central oscillation at a peak, where use of sin results in the central oscillation at a
*     valley.
      beta(ydiv)=(obf*dcos(2d0*dist*pi/lamda))/(2d0*dist*pi/lamda)
      beta(ydiv)=beta(ydiv)+betaold
      end if
5     continue

      do 4 j=1,h
      write(01,*)sngl(atan((yscmin+increm*j)/z)),sngl(abs(beta(j)**2))
4     continue
      end
```