2nd Assignment-I

- Write a program that generates random realization of a Plummer sphere (positions only).

- Write separately a routine that generates radial coordinates, and another that generates random points on the unit sphere. Combine the results to generate cartesian coordinates for each point.

- Make a plot of the theoretical radial density profile *vs*. a histogram of the radial distribution of particles, normalized to the volume of the spherical shell used for each radial bin.

- Determine the global *rms* deviations between both curves as a function of the size of the random realization.

2nd Assignment-I

Let us work with a dimensionless version of this model:

$$\delta_P(\eta) = (1+\eta^2)^{-5/2}, \quad \text{where} \quad \eta \equiv r/r_o, \quad \text{and} \quad \delta_P(\eta) = \rho_P(\eta)/(3M/4\pi r_o^3)$$

The cumulative distribution function can be obtained integrating $\delta_P(\eta)$ in spherical shells:

$$4\pi \int_0^\eta s^2 \delta_P(s)\,ds = \frac{4\pi}{3}\frac{\eta^3}{(1+\eta^2)^{3/2}} \rightarrow \frac{4\pi}{3}, \quad \text{when} \quad \eta \rightarrow \infty$$

So we define the normalized cumulative distribution function as:

$$\mu_P(\eta) \equiv \frac{\eta^3}{(1+\eta^2)^{3/2}}$$

The inverse function is simply,

$$\eta = \sqrt{\frac{\mu_P^{2/3}}{1-\mu_P^{2/3}}}$$

2nd Assignment-I

$$\eta = \sqrt{\frac{\mu_P^{2/3}}{1 - \mu_P^{2/3}}}$$

Main program (f77)

```
*------------ Plummer model positions ------------

      open(1,file='plummer.dat',form='formatted',status='new') !open output file

      do i=1,Np                      !particle do-loop
         aux = rand()                !random number between 0 & 1
         aux = aux**(2./3.)
         aux = aux/(1. - aux)        !inverse cumulative function for r
         r = ro*Sqrt(aux)            !random radial coordinate for Plummer
         call randangle(x,r)         !random direction

         write(1,20) i,(x(j),j=1,3),r !dump particle info
 20      format(1x,i6,4x,3(1pe10.3,1x),4x,1pe10.3)
      end do

      close(1)                       !close output file
```

2nd Assignment-I

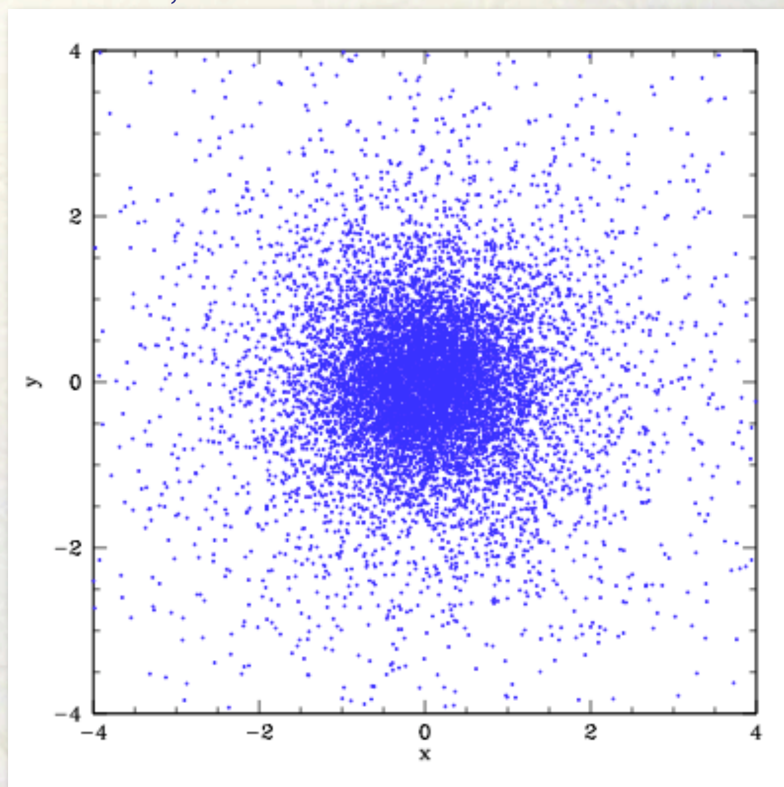
Randangle subroutine (f77)

```
      Subroutine randangle(x,r)
*
* It generates the 3 coordinates x(3), of a random direction vector of user-supplied modulus r.
* Method: 2 random numbers are generated: z & theta.
* The 1st is the z-component, the 2nd, the azimuth (angle of x-y projection with x-axis)
• of a unit vector.
• The final cartesian coordinates: x(3) are obtained by simple scaling by r.
* Notice the use of z, instead of the other spherical angle to avoid a non-uniform
• Coverage of the unit sphere.
*
*          Luis A. Aguilar                                   4/20/92
*
       PARAMETER (pi=3.14159265)    !pi
       REAL r
       REAL x(3)
       REAL z,theta,aux

       z = 2.*rand() - 1.           !random number between -1 y +1.
       theta = 2.*pi*ran1(iseed)    !random number between 0 & 2*pi.
       aux = Sqrt(1. - z*z)         !XY-projection of unit vector of z-coordinate
       x(1) = r*Cos(theta)*aux      !cartesian coordinates
       x(2) = r*Sin(theta)*aux
       x(3) = r*z

            return
            end
```

2nd Assignment-I

Random realization of Plummer model in *x-y* projection.

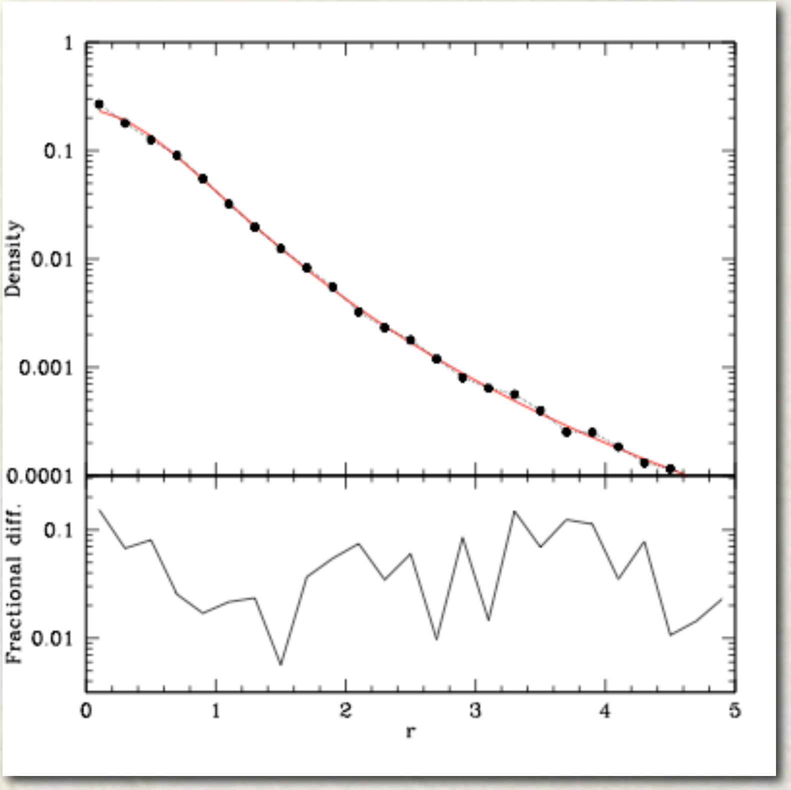
N = 10,000



2nd Assignment-I

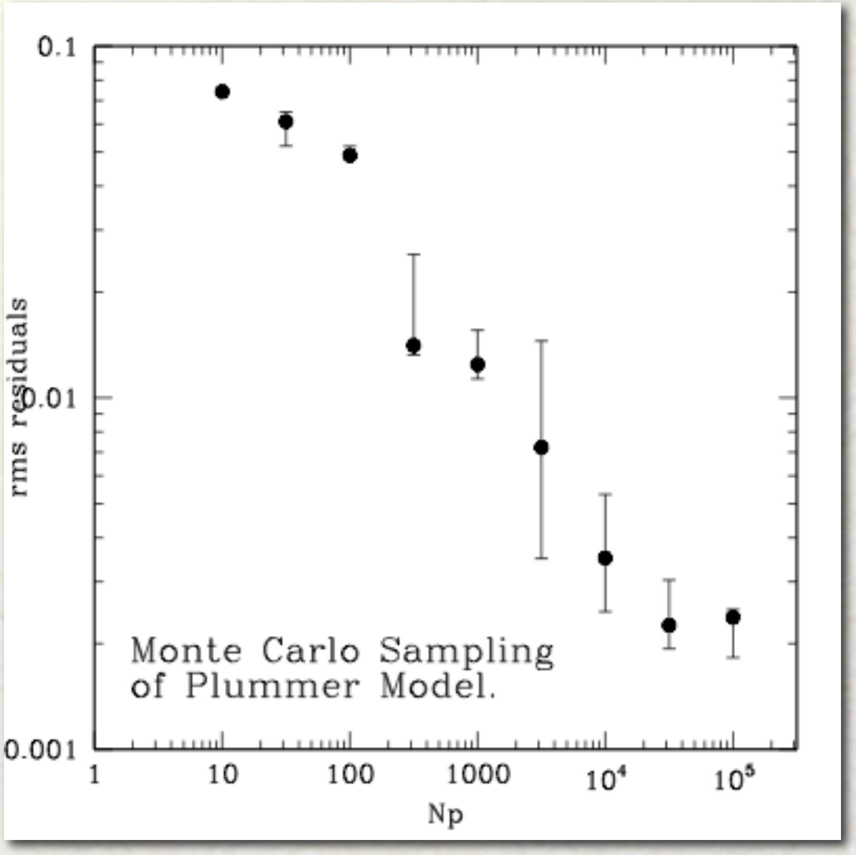
Density profile and residuals of random realization of Plummer model.

N = 10,000



2nd Assignment-I

Overall *rms* residuals of random realizations of plummer models of increasing size.



Homework

2nd Assignment - II

- Write a program that approximates the value of the integral:

$$A = \frac{1}{\sqrt{2\pi}} \iint_{x^2+y^2 \le 1} \exp[-(x^2+y^2)/2]\, dxdy$$

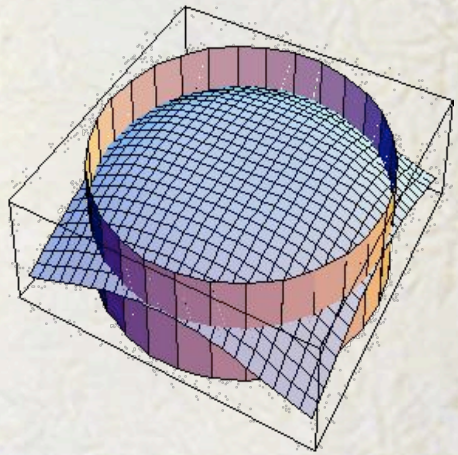
using the von Neumann Rejection Technique.

- List the approximated values A_n obtained using n points, as well as the fractional errors: $(A_n - A_o)/A_o$; where A_o is the exact value.
- Make a plot of the fractional error as a function of n.

2nd Assignment-II

```
      Program Aintegral
*
*     It uses the von Neumann Rejection Technique to approximate
*     the value of the integral:
*
*     A = Int (1/sqrt(2Pi))*Exp[-(x^2 + y^2)/2] dxdy,
*     within the domain: (x^2 + y^2) < 1
*     The exact value is: sqrt(2Pi)*[1-(1/sqrt(e))] ~ 0.986281
*
*     Luis A. Aguilar
*     (aguilar@astrosen.unam.mx)
*     May/2008
*
      parameter (Ao=0.986281) !exact value
      real A                  !Approximate value
      real rmax               !radius of integration domain
      real error              !absolute fractional error
      integer n               !size of sample

      rmax = 1.
      n = 10                  !initial sample size
      do i=1,10
         call Integ(n,A,rmax) !von Neumann integration
         error = Abs(A-Ao)/Ao !absolute fractional error
         write(6,10) n,A,error!write result
 10   format(1x,i9,3x,1pE11.5,2x,1pE9.4)
         n = 2*n              !duplicate sample size
      end do

      stop
      end
```

2nd Assignment-II



```
      subroutine Integ(n,A,rmax)
*     von Neumann integration routine:
*                 n:     number of points to use
*                 A:     returned approximate value of integral
*                 rmax:  radius of integration domain
*

      integer n,m
      real A,rmax,fmx
      real xrand,yrand,r2,frand,fraction
      external func             !function to be integrated

      fmx = func(0.,0.)        !maximum value of 'func' within domain

      m = 0                     !initialize successes counter
      do i=1,n
 10      xrand = ran()         !random x within (0,1)
         yrand = ran()         !random y within (0,1)
         r2 = xrand*xrand + yrand*yrand   !respective modulus squared
         if (r2.gt.1.) go to 10          !try again if point outside domain
         frand = fmx*ran()                !random f within (0,fmx)
         if (frand.lt.func(xrand,yrand)) m = m + 1 !add to counter if good
      end do

      fraction = float(m)/float(n)  !ratio of sampled to under func volumes
      A = 3.14159265*fraction       !multiply by sampled volume

      return
```