The Formation fo Super Massive Black Holes From Pre-Galactic Fragments

David Yoshida

under the direction of
Dr. Edmund Bertschinger
Massachuusetts Institute of Technology

Research Science Institute
July 31, 2001

Abstract

This experiment sought to show through N-body simulation that an alternate path to super massive black hole formation is possible, in which a super massive black hole is formed from disorderly fragments of stellar black holes and dark matter. The results of the experiment show that a constant supply of particles must be streamed into a black hole binary system in order to effect a merger. This suggests that a model for super massive black hole formation in which pre-galactic fragments are constanly colliding could provide the necessary matter to effect stellar black hole merger, eventually with the result of a super massive black hole forming.

1 Introduction

The Hubble telescope has found that every nearby large galaxy contains a super massive black hole in its center [3]. These super massive black holes (SMBH) fall between the masses of 10^6 and 10^9 solar masses ($M_\odot$), compared to the much less massive stellar black holes (BH), which are approximately 10 $M_\odot$. Until recent discoveries by the Chandra X Ray Observatory of black holes with intermediate mass in the M82 starburst galaxy, all known black holes had a mass either less than 20 $M_\odot$ or greater than $10^6 M_\odot$, which indicates that two different black hole formation followed two distinct paths [4]. BHs are commonly understood to have formed by the collapse of massive stars, yet it is unknown how the larger SMBHs form.

Not only is there question regarding the formation of SMBHs still open, but also questions about the relationship between a galaxy and the SMBH it contains. The presence of a SMBH at the center of a galaxy depend on the presence of a galactic bulge. Also, the mass of a SBMH is dependent on the SMBH's galactic bulge, continually observed at .2% of the total mass of the bulge [3].

For a black hole to grow in mass until it becomes a SMBH, a black hole must progress in one of two ways, it must either continuously be fed by small bodies over a long period of time, or must merge with a relatively fewer number of larger bodies. The first process would require a steady formation of bodies whose orbits exactly coincide with the event horizon of a black hole, since the black hole would presumably be consuming each body as the body collided with the black hole. Any body whose orbit was the slightest bit off from the black hole's event horizon would not add to the mass of the black hole but would be caught in an orbit within the black hole's accretion disk. In a normal scenario, perhaps a few bodies would have the right trajectory to be consumed by a black hole, yet once they were consumed, the mass of the black hole would not increase. Thus there would have to be a continuous supply of bodies falling onto the black hole. This process is implausible because of its preconditions.

The other scenario would require a black hole to merge with other more massive bodies, such as other black holes. In this process, presumably a binary, perhaps a black hole binary, would form. For the two bodies of the binary to merge, their gravitational potential energy must become more negative, as the distance between the two binaries decreases. For the energy of the binary to decrease, the energy must be transferred out of the system. This can be accomplished through gravitational radiation and three body ejection. Gravitational radiation, which would transfer energy to emitted gravity waves, first requires a hard binary, in which the two bodies are rotating quickly and are close to one another already. In order to progress a binary to the point at which gravitational radiation can take over, a binary must transfer energy to other bodies of close proximity through a three body ejection, in which the binary transfers potential energy to a third body, resulting in the third body being thrown out of the system as its gravitational potential energy becomes more positive.

If the formation of a SMBH must involve mergers between a few more massive bodies, then the intermediate black holes (IMBHs) which have been afore mentioned may be the raw material for the creation of a SMBH. A model which uses IMBHs has been proposed, which utilizes hierarchical collisions of multiple IMBH to form a SMBH [1]. This model relies on dynamical friction to draw IMBH's to the center of a galaxy where they merge. After multiple collisions and mergers, a conglomerate of mass equivilant to a SMBH would be formed. That SMBH would then continue to sink to the center of a galaxy under the influence of dynamical friction. However, it has been pointed out that such collisions would result in a black hole binary which would not be able to merge into a single black hole, because there would not be enough stars surrounding the IMBH binary for a sufficient transfer of energy to occur so that gravitational radiation could effect a merger of the two IMBHs [2].

Another theory uses a gas accretion disk surrounding a black hole binary to bring about the merger of the binary. This gas accretion disk assisted merger would be hierarchially repeated, with the BHs of the first merger eventually merging to form a SMBH. This theory though assumes that SMBH formation succeeds galactic formation, and postulates the formation of a SMBH within a relaxed and orderly system, such as a galaxy or globular cluster.

This paper proposes another theory of SMBH formation, which assumes a time frame for SMBH formation which coincides with galactic formation. In this proposal, merging of BH binaries is effected through both gravitational radiation as well as three body interactions. Coalescing pregalactic fragments which continuously collide will provide the necessary matter for continuous three body ejections which will bring black hole binaries sufficiently close that they will be able to merge through gravitational radiation.

The accuracy of this proposal will be measured through a n-body simulation, in which given a set of preconditions, the effect of the forces of gravity on a system can be measured. The program, will simulate a pregalactic distribution of bodies, and will simulate the gravitational forces using a direct summation of gravitational forces. Thus what we seek in our results is data which shows that Black hole binaries can only be brought close enough to merge when there is a sufficiently large supply of matter to which energy can be transferred.

2 N-body simulation

The N-body imulations which were run each contained 100 bodies, two of which were initialized as black holes. The other 98 bodies in each simulation were initialized as relatively light bodies of dark matter, each of which was one order of magnitude less massive than the black holes. In each case, the positions of the bodies within the system were randomized within a two unit cube. The velocities of the bodies were also initially randomized between +.05[1] and -.05. The equation used to calculate the gravitational forces within the system was the non-relatavistic Newtonian gravitational equation.

The scheme of the N-body simulation used was modeled on a scheme first proposed by Haruo Yoshida in order to update calculations of gravity, velocity, and position while maintaining optimal accuracy. Thus the three calculations central to the simulation were

[1] velocity is a unit of 65.5928 meters per second

therefore divided into a type of leapfrog scheme in which the position was updated first from X_n to $X_{n+\frac{1}{2}}$ using V_n. g_n was next updated to g_{n+1} followed by an update from V_n to $V_{n+\frac{1}{2}}$ using g_{n+1}. Finally, $X_{n+\frac{1}{2}}$ was updated to X_{n+1} using V_{n+1}.

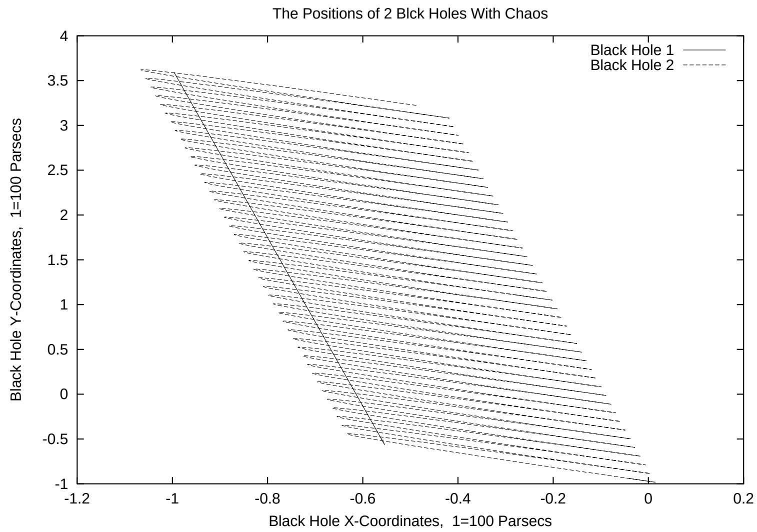
The precision of the conservation of energy check seemed to be dependent on the time-step taken. In order to measure the effect of the length of time-step on the precision of the calculations, several convergence tests were applied to the n-body problem, in which varied time-steps were taken, and the precision of the conservation of energy as well as linear and angular momentum measured.

ϵ	$\frac{\Delta E}{E_i}$	$\frac{\Delta PX}{PX_i}$	$\frac{\Delta PY}{PY_i}$	$\frac{\Delta PZ}{PZ_i}$	$\frac{\Delta LX}{LX_i}$	$\frac{\Delta LY}{LY_i}$	$\frac{\Delta LZ}{LZ_i}$
.1	-895465	-5.9289	-261.856	-261.856	117.594	132.006	59.6251
.01	-7605.69	-5.9289	-261.856	-145.687	106.154	113.352	37.5667
.001	- 9.04651e-07	-5.9289	-261.856	-64.4902	80.0893	40.0535	15.6393
.0001	-3.73613e-08	-5.9289	-24.7737	-24.4488	54.6416	29.7032	12.946

Table 1: Convergence

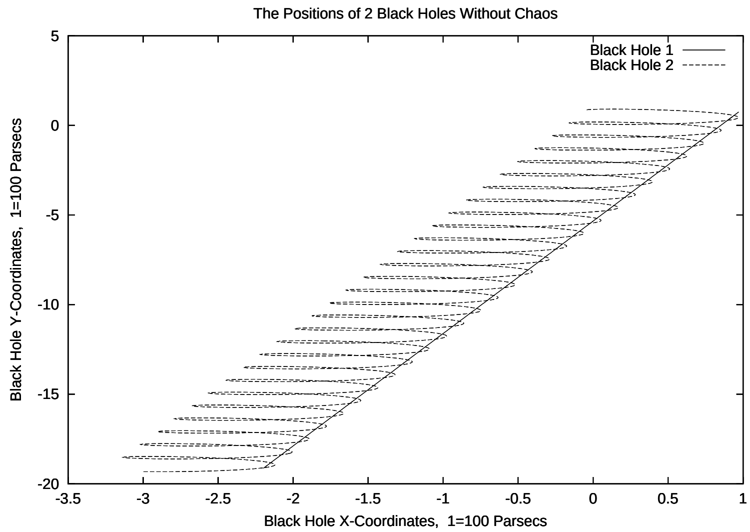
It was found that the convergence of Δe, P and L converged with ϵ to a constant in the case of each different convergence test.

Although the precision of the tests seem to converge as the time-step taken becomes smaller, as hard binaries begin to form with increasingly short periods, even the smallest reasonable initial time-step becomes too large to accurately measure the gravitational forces as the two black holes orbit. The options to solve this problem would be to soften the forces Thus in the actual experiments, a variable time-step was used, which decreased as the gravitational potential energy of the system decreased. Potential energy was used to determine the time-step, because as the period orbit of the black hole binary decreased, the potential energy of the system decreased as the potential energy of the black holes was transformed to the kinetic energy of the lighter bodies in the system. Thus the decrease of potential energy of the system was an accurate measure for the relative hardness of the binary.

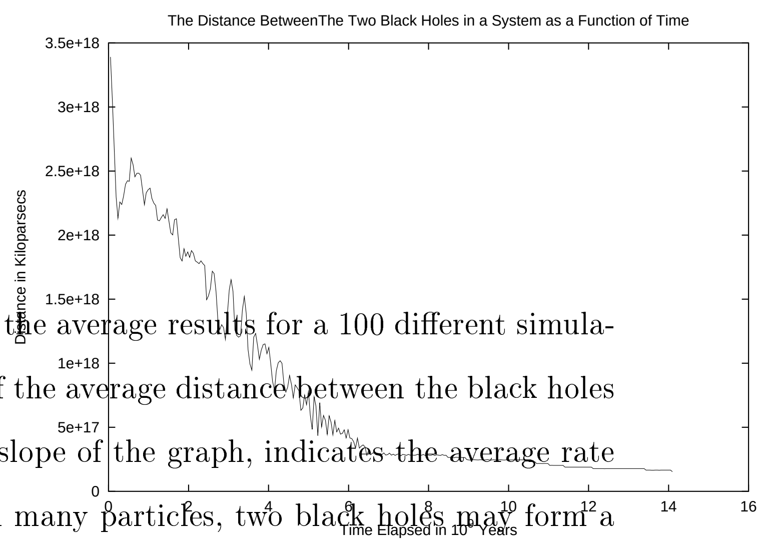


It must be noted that the N-body simulation used did model a chaotic system as shown in figures 1 and 2. In these two diagrams, the positions of the two black holes were tracked. The starting conditions were identical for both systems, except that one of the black holes in the second figure had its initial X coordinate shifted by 10^{-6}. The positions of the two pairs of black holes diverged rapidly as a result of this slight initial change. The reason why this system was made so chaotic, the signature of which is the sensitivity to the most minute change, was the finite precision of the floating point computer operations. In order for the system to be made non-chaotic and therefore more predictable, the the number of digits used by the computer in the calculations would have had to grow linearly with time as the simulation progressed in order to retain all the significant digits of the calculation.

One limitation imposed by this simulation which must be noted is the mass distribution in the system. In an more realistic system, the M_{BH} to M_{star} ratio should be 10^3. However, in this simulation, the mass ration is 10. The reason that a more realistic ratio was not used, was the limit imposed by computational power. For the mass of the blck hole to not dominate the system, the simulation would have required 10^3 bodies. However, this could not be simulated in a reasonable amount of time.



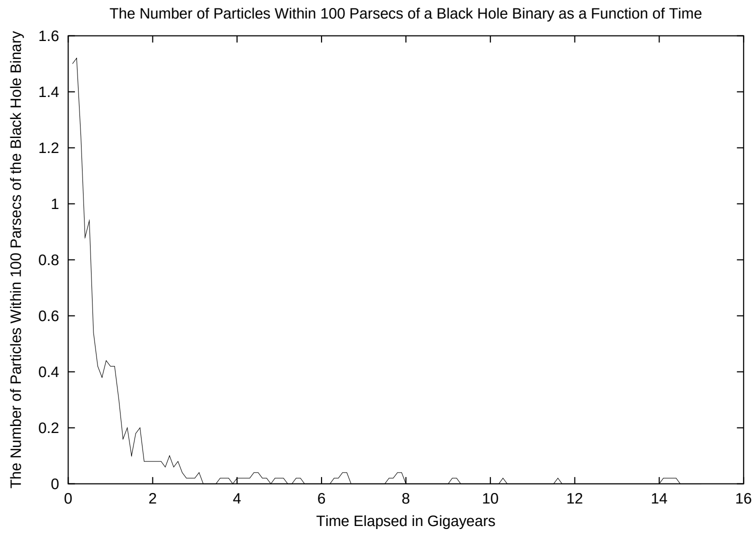
3 Data



4 Analysis

The measurements of the simulation represent the average results for a 100 different simulations. The first graph represents the findings of the average distance between the black holes in each simulation as a function of time. The slope of the graph, indicates the average rate at which, in a colliding galactic fragment with many particles, two black holes may form a hard binary.

The second graph shows the average rate at which stars are ejected from a cluster as their orbit brings them near to a black hole binary and they receive sufficient kinetic energy to be ejected from the vicinity of the black hole binary. The rate of ejection of lighter bodies from the vicinity of the black hole was found to often expell all particles in the system within the first billion years of the simulation. In this case, a unit sphere footnote1 unit is equal



to 100 parsecs was constructed around the center of the black hole binary to determine the number of particles in the vicinity of the black holes. The radius of the sphere used to collect the data was arbitrarily chosen as an indicator of the relative positions of the bodies in the simulation to the lack holes.

Taken together the two graphs show a direct correlation between the ejection of a body from the system, and the decrease of distance between the two black holes in this simulated system of colliding fragments. Extrapolation of the two curves indicate that a steady supply of bodies falling into, and being ejected by the black hole binary is necessary for the binary to continue to merge into a single more massive black hole.

5 Conclusion

The experiment confirms the initial hypothesis that a pregalactic formation for a supermassive black hole from non-relaxed fragments of black holes and dark matter is indeed a plausible theory for SMBH formation. Furthermore, the data which shows that a constant supply of dark matter is necessary for a black hole binary to continue to harden until it coaleses into a single black hole. Thus the Ebisuzaki model, in which a few collisions between relaxed globular clusters with finite ammounts of dark matter cause hierarcical merging may require further study. Furthermore, a single gass accretion disk with finite mass may also be inadequate to facilitate a black hole merger.

The experiment could be further developed through experimentation with a larger number of bodies and dark matter of varying masses, which would more accurately model a disorderly pregalactic fragment. Also, the effect of the number of bodies in the simulation nf the time required for the pair of black holes to form a binary also maybe ,easured.

6 Acknowledgments

I would like to thank the generosity of the Center for Excellence in Education for extending to me the opportunity to attend the 2001 Research Science Institute. However, much credit for this paper also belongs to my mentor, Dr. Bertschinger who proposed and suppoted my research on this project. I would also like to recognize Alexander Shirokov, whose technical assistance helped to guide my programming. Finally, I would like to thank Lisa Powell, my tutor who endured all my whining on this adventure.

References

[1] T. Ebisuzaki et al. The "Runaway" Path to Supermassive Black Holes. astro-ph/0106252;

[2] A. Gould & H Rix. Astrophys. J.,532 :L29-L32 (2000).

[3] J. Kormendy & D. Richstone 1995. ARA& A, 33 581.

[4] H. Matsumoto & T. Tsuru. 1999 PASJ, 51, 321.

[5] D. Merritt, L. Ferrarese, C. Joseph, No Supermassive Black Hole in M33?, Science 2001.

[6] H. Yoshida 1990, Phys. Lett. A150, 262.

```
\begin{singlespace}
#include <iostream.h>
#include <math.h>
#include <fstream.h>
#include <stdio.h>
#include<stdlib.h>
#include <string.h>

struct avx {
  double Vx;
  double Vy;
  double Vz;
  double x;
  double y;
  double z;
  double gx;
  double gy;
  double gz;
  double mass;
};


double rnd()
{
  return( 2.0*(double (rand())/(double)RAND_MAX) - 1.0);
}
```

```
double sqr(double x)
{
  return x*x;
}

void initialize_rand (avx *body, double ve, int N)
{
  for (int i=0; i<N; i++)
   {
    if (i<2)
      body[i].mass = 10.0;
    else
      body[i].mass = 1.0;
    body[i].x = rnd();
    body[i].y = rnd();
    body[i].z = rnd();

    body[i].Vx = ve * 0.5 * (double)rnd();
    body[i].Vy = ve * 0.5 * (double)rnd();
    body[i].Vz = ve * 0.5 * rnd();
  }
}

void initialize_circ (avx *body, double ve, int N)
{
  if(N!=2)
```

```
    {
      cout<<"Error: not two bodies in the initialize_cir\n\n";
      exit(1);
    }
  body[0].mass = 0.000000001;
  body[0].x =0.0;
  body[0].y =1.0;
  body[0].z =0.0;
  body[0].Vx =1.0;
  body[0].Vy =0.0;
  body[0].Vz =0.0;

  body[1].mass = 1.0;
  body[1].x =0.0;
  body[1].y =0.0;
  body[1].z =0.0;
  body[1].Vx =0.0;
  body[1].Vy =0.0;
  body[1].Vz =0.0;
}


void gravity (avx *body, int N, double *system)
{
  int i, j;

  double Dx, Dy, Dz, r, r3, ePE;
```

```
  ePE=0.0;
  for (i=0; i<N; i++){
    body[i].gx = 0.0;
    body[i].gy = 0.0;
    body[i].gz = 0.0;
  }

  for (i=0; i<N; i++){
    for (j=0; j<i; j++){
Dx = body[j].x - body[i].x;
Dy = body[j].y - body[i].y;
Dz = body[j].z - body[i].z;

r = sqrt(Dx*Dx + Dy*Dy + Dz*Dz);
r3 = pow(r, 3.0);

body[i].gx += (body[j].mass * Dx) / r3;
body[i].gy += (body[j].mass * Dy) / r3;
body[i].gz += (body[j].mass * Dz) / r3;
ePE -= body[i].mass * body [j].mass/r;
     }
  }
  system[2]=ePE;

}

void gravity (avx *body, int N)
```

```
{
  int i, j;

  double Dx, Dy, Dz, r, r3, ePE;
  ePE=0.0;
  for (i=0; i<N; i++){
    body[i].gx = 0.0;
    body[i].gy = 0.0;
    body[i].gz = 0.0;
  }

  for (i=0; i<N; i++){
    for (j=0; j<i; j++){
Dx = body[j].x - body[i].x;
Dy = body[j].y - body[i].y;
Dz = body[j].z - body[i].z;

r = sqrt(Dx*Dx + Dy*Dy + Dz*Dz);
r3 = pow(r, 3.0);

body[i].gx += (body[j].mass * Dx) / r3;
body[i].gy += (body[j].mass * Dy) / r3;
body[i].gz += (body[j].mass * Dz) / r3;
ePE -= body[i].mass * body [j].mass/r;
     }
  }
```

```
}

void total_mechanical_energy (avx *body, int N, double *system){
  int i, j;

  static int n=0;
  n++;



  double KE, PE, Dx, Dy, Dz, r;

  KE=0.0;
  PE=0.0;

  for (i=0; i<N; i++){
    KE+= 0.5 * (sqr(body[i].Vx)+sqr(body[i].Vy)+sqr(body[i].Vz))*body[i].mass;
  }
  system [0]= (2.0 * KE) / fabs(PE);
  system [1]= KE+system [2];

  if(n==0)printf("2.0*KE/|PE|= %g\nKE+PE     = %g\nKE       = %g\n",
 system[0], system[1], system[2]);

}

void total_linear_momentum (avx *body, int N, double *slm){
  int i;
```

```
  double l_m_x=0.0, l_m_y=0.0, l_m_z=0.0;

  for (i=0; i<N; i++){
    l_m_x+=body[i].Vx*body[i].mass;
    l_m_y+=body[i].Vy*body[i].mass;
    l_m_z+=body[i].Vz*body[i].mass;
  }

  slm[0] = l_m_x;
  slm[1] = l_m_y;
  slm[2] = l_m_z;
}

void total_angular_momentum (avx *body, int N, double *am){

  int i;
  double a_m_x=0.0, a_m_y=0.0, a_m_z=0.0;
  for (i=0; i<N; i++){
    a_m_x+=body[i].y * body[i].Vz - body[i].z * body[i].Vy;
    a_m_y+=body[i].z * body[i].Vx - body[i].x * body[i].Vz;
    a_m_z+=body[i].x * body[i].Vy - body[i].y * body[i].Vx;
  }
  am [0] = a_m_x;
  am [1] = a_m_y;
  am [2] = a_m_z;
}
```

```
void move(int N,  avx * body, double *system, double dt){
  int  i;

     for (i=0; i<N; i++) {
      body[i].x += body[i].Vx * dt;
      body[i].y += body[i].Vy * dt;
      body[i].z += body[i].Vz * dt;
    }

    gravity (body, N, system);
   for (i=0; i<N; i++) {
     body[i].Vx += body[i].gx * dt * .5;
     body[i].Vy += body[i].gy * dt * .5;
     body[i].Vz += body[i].gz * dt *.5;
      }

    total_mechanical_energy ( body, N, system);
    for (i=0; i<N; i++) {
      body[i].Vx += (body[i].gx * dt * .5);
      body[i].Vy += (body[i].gy * dt * .5);
      body[i].Vz += (body[i].gz * dt * .5);
    }

    for (i=0; i<N; i++) {
      body[i].x += body[i].Vx * dt;
      body[i].y += body[i].Vy * dt;
```

```
      body[i].z += body[i].Vz * dt;
    }

}




void main ()
{

  int    N=100;

  avx    body   [N];
  avx    bodyc  [N];
  double system_i  [3];
  double system    [3];
  double l_momentum[3];
  double a_momentum  [3];
  double l_momentum_i[3];
  double a_momentum_i  [3];

  double te, vf, ei,  lmix, lmiy, lmiz,   amix, amiy, amiz,   tec;

  int i, n1, k, m;
  srand(4567);
  double err;
  double dx, dy, dz, r12;
```

```
double dt, dt1;
double t_mass=double (N-2+(2 * 10.0));
double m5;
m5 =pow(t_mass, 5.0);
double eps;
double tf=30;
char filename [200];
char ext [5]=".txt";
cout<<"Enter timestep"<<endl;
cin>>eps;
static int h=0;
for(k=0; k<10; k++)
  {
    initialize_rand (body, 1.0, N);
    for(m=0; m<N; m++)
      {
bodyc[m].x=body[m].x;
        bodyc[m].y=body[m].y;
        bodyc[m].z=body[m].z;
        bodyc[m].Vx=body[m].Vx;
        bodyc[m].Vy=body[m].Vy;
        bodyc[m].Vz=body[m].Vz;
        bodyc[m].gx=body[m].gx;
        bodyc[m].gy=body[m].gy;
        bodyc[m].gz=body[m].gz;
      }
    gravity (body, N, system_i);
```

```
system [2]=system_i[2];

total_mechanical_energy ( body, N, system_i);

ei=system [1];

total_linear_momentum   (body, N, l_momentum_i);

total_angular_momentum  (body, N, a_momentum_i);

te=0.0;

int nt=0;

char header [31] ="/tmp/run2/";

char coor [31] ="coord";

char bhcoor [31] ="bh_coords";

char tME [31] ="tM";

char withirad [31] ="within_rad";

char bhdist [31] ="bh_dist";

char conv[31] = "diff";

char filename[255] = "";

char conv1[31] = "eps2";

char conv2[31] = "eps4";

char conv3[31] = "ave";


cout<<k<<'\n';

//!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

ofstream fout;

sprintf(filename, "%s%s%i%s", header, coor, k, ext);

fout.open (filename);

//filename ="\";

ofstream fout1;

sprintf(filename, "%s%s%i%s", header, tME, k, ext);
```

```
    fout1.open(filename);
    //filename ="\0";
    fout<<N<<'\n';
    ofstream fout3;
    sprintf(filename, "%s%s%i%s", header, bhcoor, k, ext);
    fout3.open (filename);
    //filename ="\0";
    ofstream fout4;
    sprintf(filename, "%s%s%i%s", header, withirad, k, ext);
    fout4.open (filename);
    //filename ="\0";
    ofstream fout5;
    sprintf(filename, "%s%s%i%s", header, bhdist, k, ext);
    fout5.open (filename);
    //filename ="\0";
    //!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
    while (te<tf)
{
  dt=eps*sqrt(-m5/(pow(system[2],3)));
  if (tf-te<dt) dt=tf-te;
  dt1=dt/(2.0-pow(2.0,1.0/3.0));
  move(N, body, system, dt1);
  move(N, body, system, dt-2.0*dt1);
  move(N, body, system, dt1);
  te += dt;
  nt++;
  err=(system[1]-system_i[1])/system_i[1];
```

```
  fout1<<te<<' '<<err<<'\n';
  if (nt%10 == 0|| te==tf)
    {
      for(i=0; i<N; i++)
       fout<<body[i].x<<' '<<body[i].y<<' ';
      fout<<'\n';
      n1=0;
      for (i=2; i<N; i++)
{
  dx = body[i].x-0.5 * (body [0].x+ body [1].x);
  dy = body[i].y-0.5 * (body [0].x+ body [1].x);
  dz = body[i].z-0.5 * (body [0].x+ body [1].x);
  r12=sqrt (dx*dx + dy*dy + dz*dz);
  if (r12<1.0)
  n1++;
}
      dx = body[0].x - body[1].x;
      dy = body[0].y - body[1].y;
      dz = body[0].z - body[1].z;
      r12=sqrt (dx*dx + dy*dy + dz*dz);
      fout3<<body[0].x<<' '<<body[0].y<<' '<<body[1].x<<' '<<body[1].y<<'\n';
      fout4<<te<<' '<<n1<<'\n';

      fout5<<te<<' '<<r12<<'\n';



      total_linear_momentum   (body, N, l_momentum);
```

```
      total_angular_momentum  (body, N, a_momentum);


      fout1<<te<<' '<<system[1]<<'\n';



    }
}
//|||||||||||||||||||||||check||||||||||||||||||||||||||||||||||||||||||||||||
/**/  te=0.0;
/**/  gravity (bodyc, N, system);
/**/  nt=0;
/**/ while (te<tf)
/**/   {
/**/    dt=eps*sqrt(-m5/(pow(system[2],3)));
/**/      if (tf-te<dt) dt=tf-te;
  dt1=dt/(2.0-pow(2.0,1.0/3.0));
  move(N, bodyc, system, dt1);
  move(N, bodyc, system, dt-2.0*dt1);
  move(N, bodyc, system, dt1);
  te += dt;
          total_linear_momentum   (body, N, l_momentum);
          total_angular_momentum  (body, N, a_momentum);
}



       fout.close();
       fout1.close();
```

```
       fout3.close();

       fout4.close();

       fout5.close();


        ofstream fout2;

        sprintf(filename, "%s%s%i%s", header, conv, k, ext);

fout2.open(filename);

        double eps2=eps * eps;

double eps4= eps2*eps2;

        double d_l_x,  d_l_y, d_l_z, d_a_x, d_a_y,  d_a_z, d_e, d_p;

        double a_d_l_x=0,  a_d_l_y=0, a_d_l_z=0, a_d_a_x=0, a_d_a_y=0,  a_d_a_z=0, a_d_e

d_p= body [0].x-bodyc [0].x;

        d_e=system[1]    - system_i[1];

        d_l_x=l_momentum_i [0]- l_momentum[0];

d_l_y=l_momentum_i [1]- l_momentum[1];

d_l_z=l_momentum_i [2]- l_momentum[2];

        d_a_x=a_momentum_i[0]- a_momentum[0];

        d_a_y=a_momentum_i[1]- a_momentum[1];

d_a_z=a_momentum_i[2]- a_momentum[2];

        a_d_l_x += d_l_x;

        a_d_l_y += d_l_y;

        a_d_l_z += d_l_z;

        a_d_a_x += d_a_x;

        a_d_a_y += d_a_y;

        a_d_a_z += d_a_z;

a_d_e += d_e;

        a_d_p += d_p;
```

```
        fout2<<d_p<<' '
<<d_e<<' '


<<d_l_x<<' '
<<d_l_y<<' '
<<d_l_z<<' '
<<d_a_x<<' '
<<d_a_y<<' '
<<d_a_z<<'\n';
        ofstream fout6;
        sprintf(filename, "%s%s%i%s", header, conv1, k, ext);
fout6.open(filename);
        fout6<<d_p/eps2<<' '
<<d_e/eps2<<' '


<<d_l_x/eps2<<' '
<<d_l_y/eps2<<' '
<<d_l_z/eps2<<' '
<<d_a_x/eps2<<' '
<<d_a_y/eps2<<' '
<<d_a_z/eps2<<'\n';


        ofstream fout7;
        sprintf(filename, "%s%s%i%s", header, conv2, k, ext);
fout7.open(filename);
         fout7<<d_p/eps2<<' '
```

```
<<d_e/eps2<<' '

<<d_l_x/eps4<<' '
<<d_l_y/eps4<<' '
<<d_l_z/eps4<<' '
<<d_a_x/eps4<<' '
<<d_a_y/eps4<<' '
<<d_a_z/eps4<<'\n';

        ofstream fout8;
        sprintf(filename, "%s%s%i%s", header, conv3, k, ext);
fout8.open(filename);
        fout8<<a_d_p/eps2<<' '
<<a_d_e/eps2<<' '

<<a_d_l_x/eps2<<' '
<<a_d_l_y/eps2<<' '
<<a_d_l_z/eps2<<' '
<<a_d_a_x/eps2<<' '
<<a_d_a_y/eps2<<' '
<<a_d_a_z/eps2<<'\n';

fout2.close();
        fout6.close();
fout7.close();
```

```
fout8.close();
  }
}
\end{singlespace}
```