Számítógépes Szimulációs Labor

2. N-test szimuláció

Készítette:
Sztreha Balázs
Informatikus fizikus

October 15, 2007

1 Bevezetés

Az N-test probléma főleg nagy hatótávolságú erőknél okoz nagy problémát, mivel itt egy test az összes másik testtel kölcsönhatásban van. Ilyen kölcsönhatás például a gravitációs erő két test között vagy az elektromágneses kölcsönhatás két töltés között. Miután Newton megoldotta a 2-test problémát, matematikusok es numerikus problémákkal foglalkozó tudósok kezdték keresni a 3-test probléma egzakt megoldását. Matematikusok bebizonyították, hogy 3-test problémánál az integrálnak nincs zárt alakja, tehát a legtöbb trajektória kaotikus.

2 Feladatok

2.1 nbody_sh1.C program lefordítása, futtatása

A program a negyedrendű Hermite integrátor algoritmuson alapul, aminek a várható futási ideje O(N^2). A részecskék elhelyezkedését egy külső fileból adjuk meg, aminek az első sorában a részecskék száma szerepel, a második sorban a t_0, majd a többi sorban az részecskák adatai: tömeg, x, y, z, v_x, v_y, v_z. Ilyen file generálására írtam egy programot (Függelék 1.), ami a .plt file-t is megírja, így egyszerűbb több testnek az ábrázolása.
Ha a egy speciális elrendezést és sebességeket veszünk, és kicsi z irányú sebességet adunk, akkor a kimenetet az 1. ábrán láthatjuk.
Ha 10 testet rakunk a rendszerbe, eléggé kaotikus lesz a mozgás (2. ábra).

2.2 nbodypp.cpp program lefordítása, futtatása

A várható futási idő O(N^2) (3. ábra)

2.3 treecode.cpp program lefordítása, futtatása

A várható futási idő O(NlogN) (4. ábra)

2.4 fastm.cpp program lefordítása, futtatása

A várható futási idő O(N) (5. ábra)
Ahogy növeltem a részecskék számát, az illesztésnél úgy csökkent a hibaszázalék, ebből lehet következtetni, hogy tart a lineáris futási időhöz.

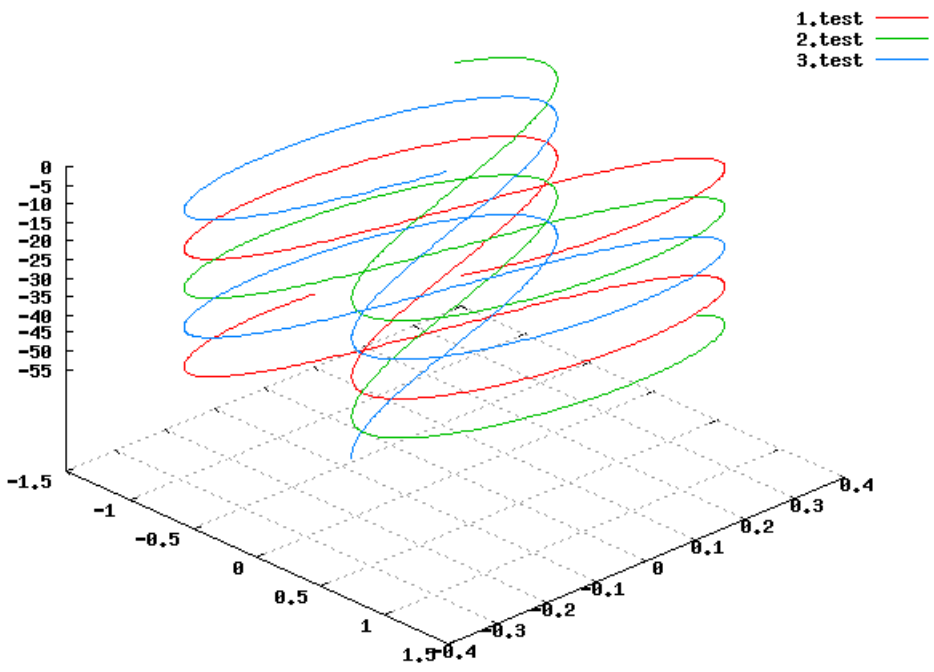

Figure 1: 3-test, enyhe z irányú sebességgel

3 Részecske-részecske kód és a tree-code összehasonlítása

A 6. ábrán az nbodypp és a treecode programok által kiszámolt komplex energiák valós részének különbsége látható a multipol együtthatók függvényében.

4 Gyorsulás számolása

A gyorsulás Newton törvényből és a térerősségből számolható az alábbi egyenletek alapján:

$$\mathbf{F} = m\mathbf{a}, \mathbf{F} = q\mathbf{E}$$

$$\mathbf{E}(z) = \sum_{i=1}^{N} \frac{q_i}{z - z_i}$$

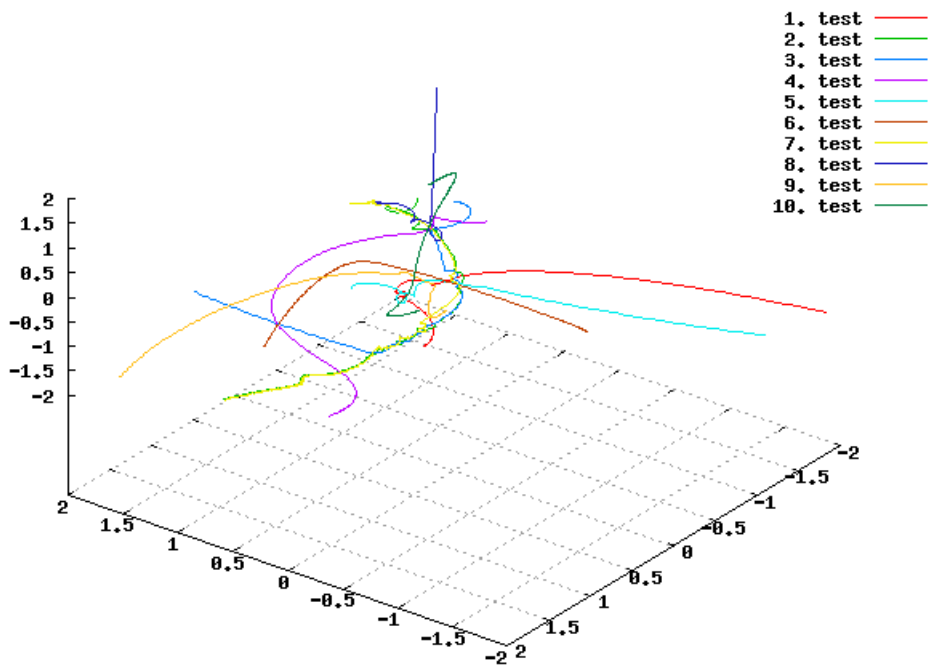

Figure 2: 10-test

Tehát a k-ik részecske gyorsulása:

$$a_k = \frac{q_k}{m_k} \sum_i \frac{q_i}{z_k - z_i}$$

A következő függvény számolja ki:

```
complex<double> computeAcceleration(int k){
complex<double> a = complex<double>(0,0);
for(int i=0;i<N;i++){
  if(i!=k) a+=q[i]/(z[k]-z[i]);
  }
  return q[k]*a;
}
```

A várakozásoknak megfelelően az eredmények egyeztek a két program esetében.

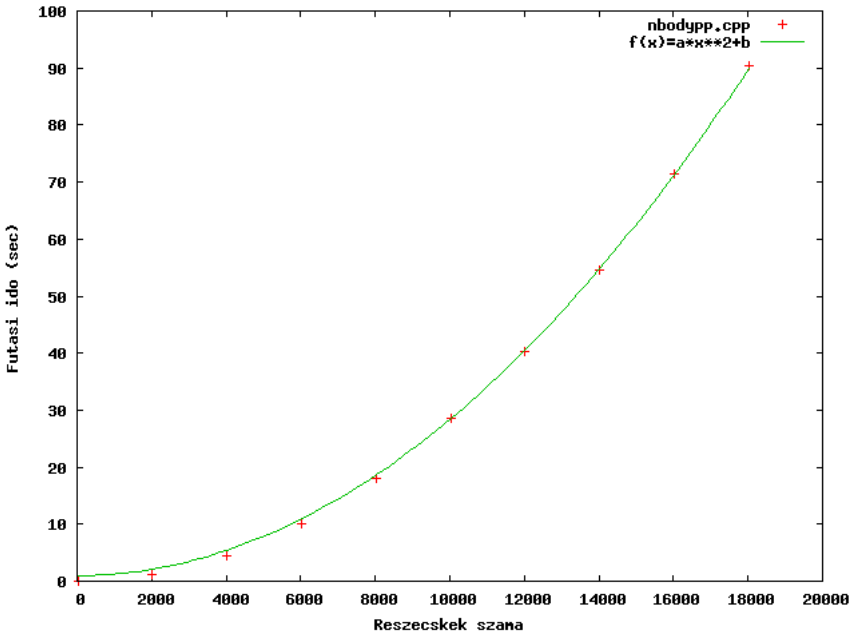

Figure 3: nbodypp.cpp futási ideje

5 Függelék

5.1 nbody_sh1.C programhoz bemenet generáló program

```
#include <cmath>
#include <cstdlib>
#include <fstream>
#include <iostream>
#include <string>
using namespace std;
int main(int argc, char *argv[]) {
    int N = atoi(argv[1]);
    double m = 1.0;
    double x,y,z,vx,vy,vz;
    int usg=1;
    ofstream file("input.data");
    ofstream plotra("plot.data");
    ofstream plt("rajzol.plt");
    file << N << '\n' << 0 << '\n';
    //plt << "set xrange[-2:2] \n set yrange[-2:2] \n set zrange[-2:2] \n";
```

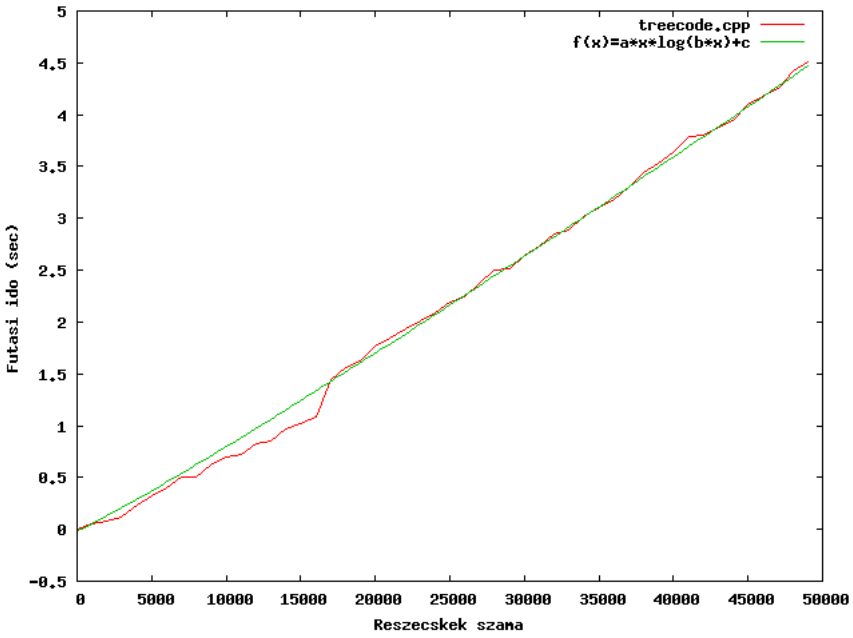

Figure 4: treecode.cpp futási ideje

```
    plt << "splot 'figure.plot' ";
    for (int i = 1; i < N+1; i++) {
      x=(2 * rand() / double(RAND_MAX));
      y=(2 * rand() / double(RAND_MAX));
      z=(2 * rand() / double(RAND_MAX));
      vx=(5 * rand() / double(RAND_MAX));
      vy=(1 * rand() / double(RAND_MAX));
      vz=(1 * rand() / double(RAND_MAX));
if(vx<0) vx=(-1)*vx;
      file << m << '\t' << x << '\t' << y << '\t' << z << '\t' << vx  << '\t' <<
      plotra << x << '\t' << y << '\t' << z << '\n';

      if(i>1) plt << "''";
      plt << " u " << usg << ":" << usg+1 << ":" << usg+2 << " w l title '" << i
      if(i<N) plt << ", ";
      usg+=3;
    }
    file.close();
    plotra.close();
    plt.close();
}
```

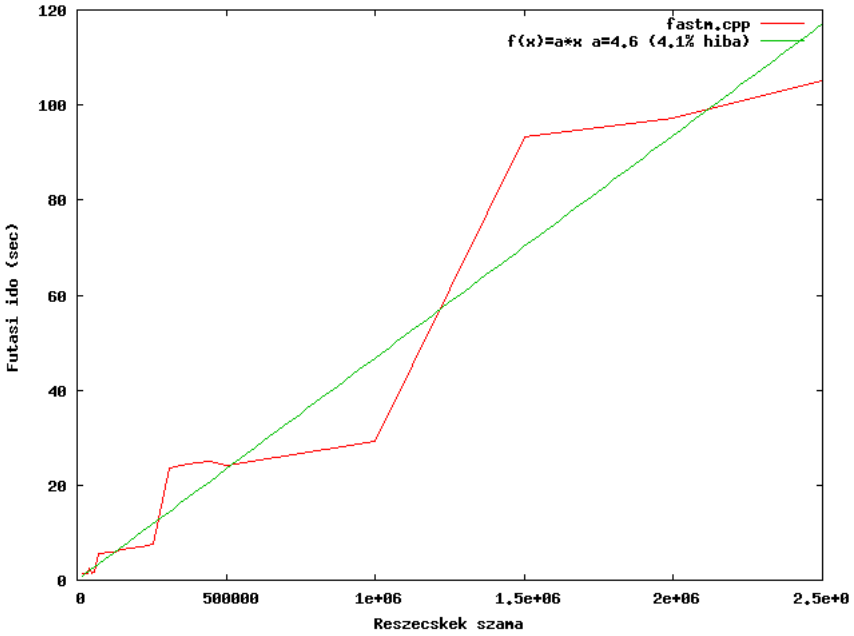

Figure 5: fastm.cpp futási ideje

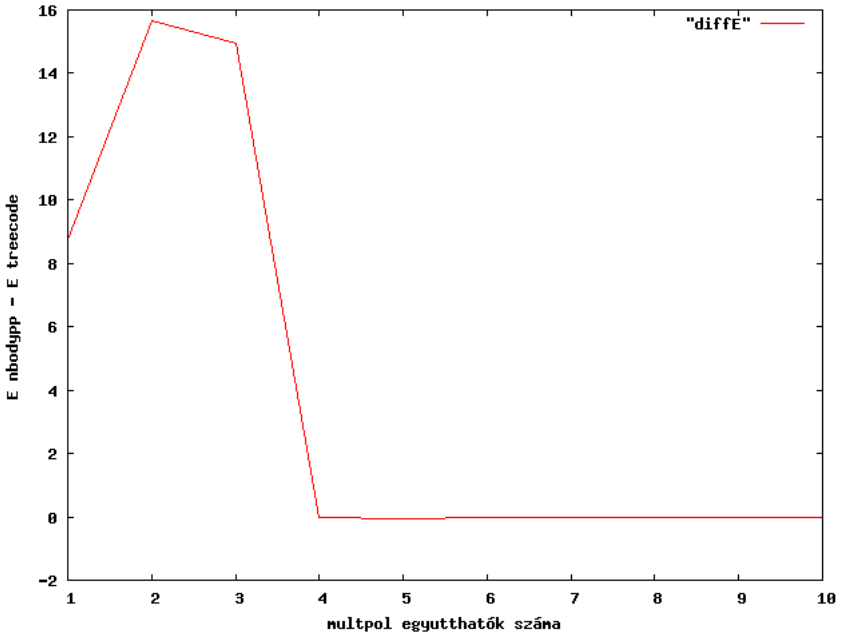

Figure 6: nbodypp és a treecode (komplex energiák valós részének különbsége a multipol együtthatók függvényében)