Контролна работа OpenMP. Пример: *класическата* задача за много тела.

Курс „Паралелно програмиране“

ИНСТИТУТ *за* СЪВРЕМЕННИ
ФИЗИЧЕСКИ ИЗСЛЕДВАНИЯ

Стоян Мишев

План

Определяне на задачата.

При дадени повече от две материални точки i с маси m_i и начални координати r_i и скорости v_i, взаимодействащи със сила f_{ij} между тях, да се определят координатите и скоростите им във всеки следващ момент.

Определяне на задачата.

Гравитационна сила:

$$\mathbf{f}_{i,j}(t) = -\frac{Gm_i m_j}{||\mathbf{r}_i(t) - \mathbf{r}_j(t)||^3}(\mathbf{r}_i(t) - \mathbf{r}_j(t))$$

Гравитационна сила:

$$\mathbf{f}_{i,j}(t) = -\frac{Gm_i m_j}{||\mathbf{r}_i(t) - \mathbf{r}_j(t)||^3}(\mathbf{r}_i(t) - \mathbf{r}_j(t))$$

Общата сила, действаща на дадена точка в момент t

$$\mathbf{F}_i(t) = \sum_{\substack{j=0 \\ j \neq i}}^{n-1} \mathbf{f}_{i,j}(t)$$

$$= -Gm_i \sum_{\substack{j=0 \\ j \neq i}}^{n-1} \frac{m_j}{||\mathbf{r}_i(t) - \mathbf{r}_j(t)||^3}(\mathbf{r}_i(t) - \mathbf{r}_j(t))$$

Движението $r(t)$ се определя от $\mathbf{F}_i(t) = m_i \ddot{\mathbf{r}}_i(t)$ (Нютон).

Движението $r(t)$ се определя от $\mathbf{F}_i(t) = m_i \ddot{\mathbf{r}}_i(t)$ (Нютон).

$$\ddot{\mathbf{r}}_i(t) = -G \sum_{\substack{j=0 \\ j \neq i}}^{n-1} \frac{m_j}{||\mathbf{r}_i(t) - \mathbf{r}_j(t)||^3} (\mathbf{r}_i(t) - \mathbf{r}_j(t))$$

За улеснение разглеждаме движение в R^2.

“Дискретизираме” времето $t = N\Delta t$

Require: input data
1: **for** each (constant) timestep **do**
2: **for** each particle i **do**
3: Compute $\mathbf{F}_i(t)$.
4: Update $\mathbf{r}_i(t)$ (and $\dot{\mathbf{r}}_i(t) := \mathbf{v}_i(t)$).
5: **end for**
6: **end for**
7: **return** $\mathbf{r}_i(T)$ for each particle i.

Require: input data

1: **for** each (constant) timestep **do**
2: **for** each particle i **do**
3: Compute $\mathbf{F}_i(t)$.
4: Update $\mathbf{r}_i(t)$ (and $\dot{\mathbf{r}}_i(t) := \mathbf{v}_i(t)$).
5: **end for**
6: **end for**
7: **return** $\mathbf{r}_i(T)$ for each particle i.

Pseudocode to compute the $\mathbf{F}_i(t)$ might look like

1: **for** each particle $j \neq i$ **do**

```
2:    dx = r[i][x] - r[j][x];
3:    dy = r[i][y] - r[j][y];
4:    d = sqrt(dx*dx+dy*dy);
5:    d3 = d*d*d;
6:    F[i][x] -= G*m[i]*m[j]/d3*(r[i][x]-r[j][x]);
7:    F[i][y] -= G*m[i]*m[j]/d3*(r[i][y]-r[j][y]);
```

8: **end for**

Примерен код с OpenMP

$f_{ij} = -f_{ji}$

```
1: for each particle i do
2:     F_i(t) = 0;
3: end for
4: for each particle i do
5:     for each particle j > i do
6:         F_i(t) += f_{i,j}(t);
7:         F_j(t) -= f_{i,j}(t);
8:     end for
9: end for
```

$$\dot{\mathbf{y}}(t) = \mathbf{f}(t, \mathbf{y}), \quad \mathbf{y}(0) = \mathbf{y}_0, \quad t > 0$$
$$\mathbf{y}_k = \mathbf{y}_{k-1} + \Delta t\, \mathbf{f}(t_{k-1}, \mathbf{y}_{k-1}), \quad k = 1, 2, \ldots, N$$

$$\dot{\mathbf{y}}(t) = \mathbf{f}(t, \mathbf{y}), \quad \mathbf{y}(0) = \mathbf{y}_0, \quad t > 0$$

$$\mathbf{y}_k = \mathbf{y}_{k-1} + \Delta t\, \mathbf{f}(t_{k-1}, \mathbf{y}_{k-1}), \quad k = 1, 2, \ldots, N$$

$$\ddot{x}(t) = -x(t)$$

$$y_1(t) = x(t), \qquad y_2(t) = \dot{x}(t)$$

$$\dot{\mathbf{y}}(t) = \begin{bmatrix} \dot{y}_1(t) \\ \dot{y}_2(t) \end{bmatrix} = \begin{bmatrix} y_2(t) \\ -y_1(t) \end{bmatrix}$$

$$\dot{\mathbf{r}}_i(t) = \mathbf{v}_i(t),$$

$$\dot{\mathbf{v}}_i(t) = \mathbf{F}_i(t)/m_i$$

```
1: r[i][x] += dt*v[i][x];
2: r[i][y] += dt*v[i][y];
3: v[i][x] += dt*f[i][x]/m[i];
4: v[i][y] += dt*f[i][y]/m[i];
```

Сериен код за енергия

```
for(i=0; i<nbodies; ++i)
  for(j=i+1; j<nbodies; ++j) {
    d2 = 0.0;
      for(k=0; k<3; ++k) {
        rij[k] = pos[i][k] - pos[j][k];
        d2 += rij[k]*rij[k];
      }
      if (d2 <= cut2) {
         d = sqrt(d2);
         d3 = d*d2;
         for(k=0; k<3; ++k) {
            double f = -rij[k]/d3;
           forces[i][k] += f;
            forces[j][k] -= f;
         }
         ene += -1.0/d;
      }
```

OpenMP код за енергия

```
#pragma omp parallel for private(i,j,k,rij,d,d2,d3)
      reduction(+:ene) schedule(guided)
 for(i=0; i<nbodies; ++i)
  for(j=i+1; j<nbodies; ++j) {
   d2 = 0.0;
   for(k=0; k<3; ++k) {
    rij[k] = pos[i][k] - pos[j][k];
    d2 += rij[k]*rij[k];
   }
   if (d2 <= cut2) {
     d = sqrt(d2);
     d3 = d*d2;
     for(k=0; k<3; ++k) {
       double f = -rij[k]/d3;
#pragma omp atomic
       forces[i][k] += f;
#pragma omp atomic
       forces[j][k] -= f;
   }
   ene += -1.0/d;
}
```

Горният код е от
`https://hpc-old.cineca.it/content/training-openmp`

Задача за контролна работа:

Използвайки разгледания код, както и фрагмента от слайд 9, определете динамиката на системата от частици с данните от `https://hpc-old.cineca.it/content/exercise-5-0`. В частност, определете местоположението на частиците след 5 времеви интервала.