Отчёт по четвертому этапу группового проекта

Образование планетной системы

Абакумов Егор, Сухарев Кирилл, Калинина Кристина, Еременко Артем

Содержание

Цель работы

Провести моделирование одного из этапов эволюции Вселенной - образование некой «солнечной» системы из межзвездного газа.

Цель этапа

Провести коллективное обсуждение результата проекта, подвести итоги работы, сделать выводы.

Определение значимых для модели свойств объекта

Для всестороннего моделирования планетарной системы нами были выбраны следующие характеристики:

- Положение тел в пространстве
- Масса
- Радиусы
- Скорость
- Ускорение
- Потенциальная энергия

Механизмы взаимодействия

Движение частиц будет вычисляться согласно II закону Ньютона:

$$F_i = m_i \frac{d^2 r_i}{dt^2}$$

Потенциальное энергия взаимодействия частицы со всеми остальными описывается следующим уравнением:

$$U_i = \sum_{i \neq j} \frac{\gamma m_j m_i}{r_{ij}}$$

Сила отталкивания между двумя частицами равна:

$$F^r(b) = k((\frac{a}{b})^8 - 1)$$

А сила трения вычисляется по формуле:

$$F^f = \mu_1 \mu_2 F^r(b)$$

Построение алгоритма

По сути алгоритм сошелся к нахождению векторной суммы всех сил, действующих на частицу, а затем к просчету ее новых координат согласно следующим законам движения:

- Координаты:

$$x_{n+1} = x_n + v_n dt + \frac{a_n dt^2}{2}$$

- Скорости:

$$v_{n+1} = v_n + \frac{a_{n+1} + a_n}{2} dt$$

Также необходимо учитывать, что при сильном сближении частицы слипаются. Их параметры в таком случае примут следующий вид.

$$r = \frac{m_i r_i + m_j r_j}{m_i + m_j}$$

$$v = \frac{m_i v_i + m_j v_j}{m_i + m_j}$$

$$R = \sqrt[3]{R_i^3 + R_j^3}$$

Программная реализация

```
58      for i in range(N):
59          d[i] = np.sqrt((x - x[i]) ** 2 + (y - y[i]) ** 2)
60          dx[i] = x - x[i]
61          dy[i] = y - y[i]
62      dx = np.divide(dx, d, where = d != 0)
63      dy = np.divide(dy, d, where = d != 0)
64
65      nax = np.zeros(N)
66      nay = np.zeros(N)
67
68      for i in range(N):
69          for j in range(N):
70              if (d[i][j] != 0):
71                  rs = r[i] + r[j]
72
73                  # Gravity force
74                  if (d[i][j] > rs):
75                      gravity_value = G * m[i] * m[j] / d[i][j] ** 2
76                      nax[i] += dx[i][j] * gravity_value
77                      nay[i] += dy[i][j] * gravity_value
78                  else:
79                      # Repulsive force
80                      repulsive_value = k * ((rs / d[i][j]) ** 8 - 1)
81                      nax[i] += dx[j][i] * repulsive_value
82                      nay[i] += dy[j][i] * repulsive_value
83                      # Friction force
84                      friction_value = repulsive_value * mu[i] * mu[j]
85                      nax[i] += -1 * dy[i][j] * friction_value
86                      nay[i] += dx[i][j] * friction_value
87
88      nax /= m
89      nay /= m
```

Figure 1: Нахождение векторной суммы всех сил

```
    while i < N - 1:
        j = i + 1
        while j < N:
            if (i == j):
                continue
            rs = r[i] + r[j]
            if (d[i][j] < alpha * rs):
                m_c = m[i] + m[j]
                x[i] = (x[i] * m[i] + x[j] * m[j]) / m_c
                y[i] = (y[i] * m[i] + y[j] * m[j]) / m_c
                vx[i] = (vx[i] * m[i] + vx[j] * m[j]) / m_c
                vy[i] = (vy[i] * m[i] + vy[j] * m[j]) / m_c
                ax[i] = (ax[i] * m[i] + ax[j] * m[j]) / m_c
                ay[i] = (ay[i] * m[i] + ay[j] * m[j]) / m_c
                r[i] = (r[i] ** 3 + r[j] ** 3) ** (1 / 3)
                m[i] = m_c
                m = np.delete(m, j)
                x = np.delete(x, j)
                y = np.delete(y, j)
                vx = np.delete(vx, j)
                vy = np.delete(vy, j)
                ax = np.delete(ax, j)
                ay = np.delete(ay, j)
                r = np.delete(r, j)
                d = np.delete(d, j, 0)
                d = np.delete(d, j, 1)
                dx = np.delete(dx, j, 0)
                dx = np.delete(dx, j, 1)
                dy = np.delete(dy, j, 0)
                dy = np.delete(dy, j, 1)
                j -= 1
                N -= 1
            j += 1
        i += 1
```

Figure 2: Слипание частиц

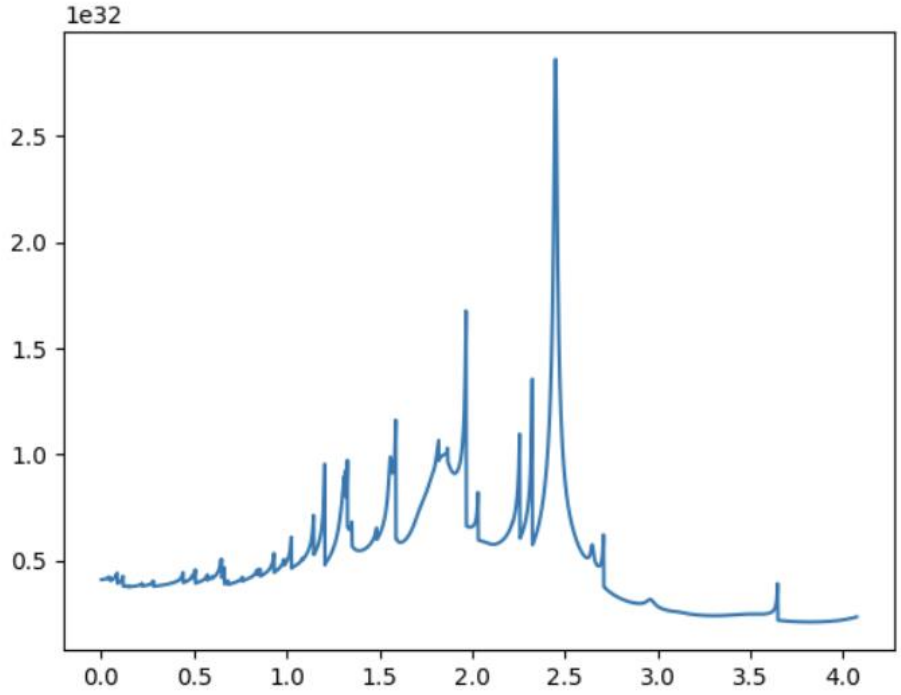

Figure 3: График потенциальной энергии

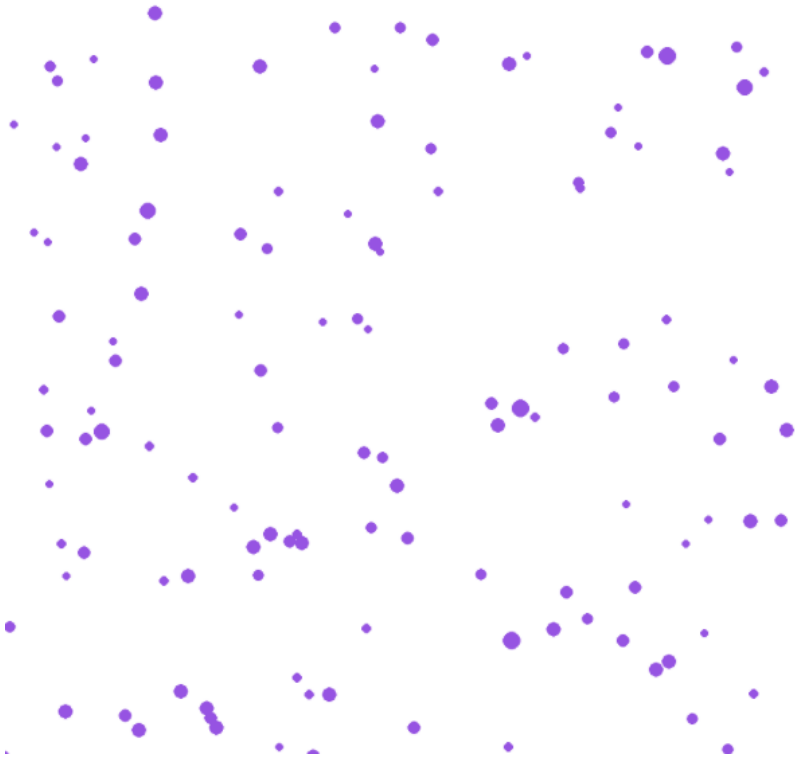

Figure 4: Графическое моделирование

Figure 5: Графическое моделирование

Figure 6: Графическое моделирование

Выводы

В ходе работы была разработана и реализована в программном коде модель некой «солнечной» системы из межзвездного газа. Проведены все математические расчеты и подготовлено теоретическое обоснование.

Оценка модели

Наш подход продемонстрировал следующие достоинства и недостатки системы:

-

  - Модель получилась объемной, охватывающей множество частиц
-
  - Модель учитывает воздействие на частицы всех значимых сил
-
  - Модель предусматривает слипание частиц и их отталкивание
-
  - Модель соотносится с реальными условиями, частицы ведут себя естественно
- – Из-за значительной вычислительной сложности пришлось ограничить масштабы модели несколькими сотнями частиц
- – Модель двумерна
- – Константы и коэффициенты взаимодействия некоторых частиц не всегда соотносятся с реальными, так как размер частицы на экране технически ограничен количеством пикселей, невозможно подобрать действительные коэффициенты

Самооценка

Свою работу наша группа оценивает положительно, так как все основные аспекты моделируемого объекта были учтены, необходимые практические результаты были получены и продемонстрированы. Работа была тщательно проанализирована, ошибки учтены и исправлены, выводы по результатам сделаны, а оставшиеся недостатки обусловлены лишь техническими ограничениями.