

# Отчёт по четвертому этапу группового проекта

## Образование планетной системы

---

Абакумов Егор, Сухарев Кирилл, Калинина Кристина, Еременко  
Артем

# Цели

---

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

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

# Теоретическое обоснование

---

Для всестороннего моделирования планетарной системы нами были выбраны следующие характеристики:

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

Движение частиц будет вычисляться согласно II закону Ньютона:

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

Потенциальная 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:

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

А сила трения вычисляется по формуле:

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



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

---

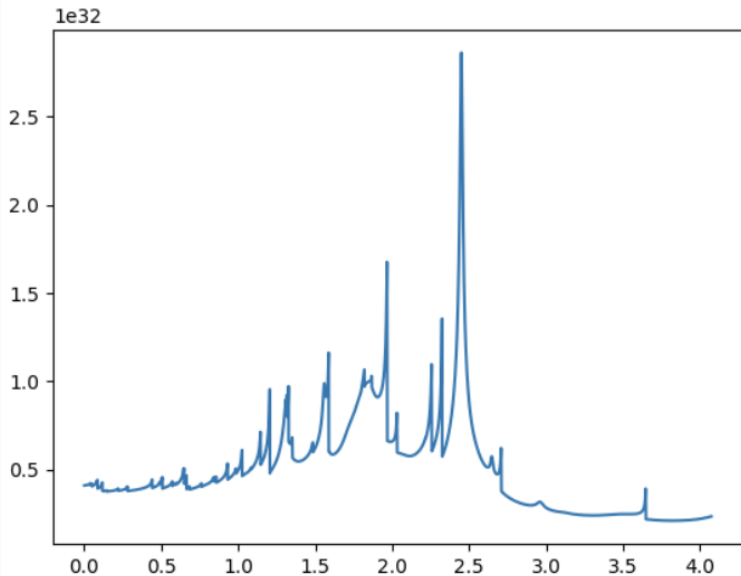
# Нахождение векторной суммы всех сил

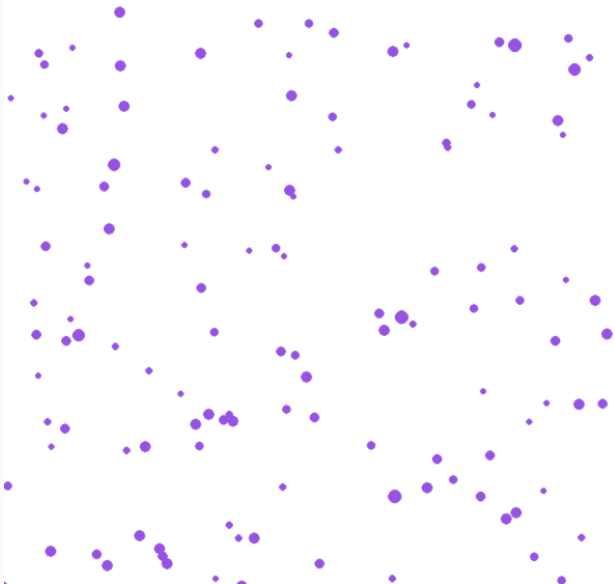
```
58 for i in range(N):
59     d[i] = np.sqrt((x - x[i]) ** 2 + (y - y[i]) ** 2)
60     dx[i] = x - x[i]
61     dy[i] = y - y[i]
62     dx = np.divide(dx, d, where = d != 0)
63     dy = np.divide(dy, d, where = d != 0)
64
65     nax = np.zeros(N)
66     nay = np.zeros(N)
67
68 for i in range(N):
69     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
```

# Слипание частиц

```
107 while i < N - 1:
108     j = i + 1
109     while j < N:
110         if (i == j):
111             continue
112         rs = r[i] + r[j]
113         if (d[i][j] < alpha * rs):
114             m_c = m[i] + m[j]
115             x[i] = (x[i] * m[i] + x[j] * m[j]) / m_c
116             y[i] = (y[i] * m[i] + y[j] * m[j]) / m_c
117             vx[i] = (vx[i] * m[i] + vx[j] * m[j]) / m_c
118             vy[i] = (vy[i] * m[i] + vy[j] * m[j]) / m_c
119             ax[i] = (ax[i] * m[i] + ax[j] * m[j]) / m_c
120             ay[i] = (ay[i] * m[i] + ay[j] * m[j]) / m_c
121             r[i] = (r[i] ** 3 + r[j] ** 3) ** (1 / 3)
122             m[i] = m_c
123             m = np.delete(m, j)
124             x = np.delete(x, j)
125             y = np.delete(y, j)
126             vx = np.delete(vx, j)
127             vy = np.delete(vy, j)
128             ax = np.delete(ax, j)
129             ay = np.delete(ay, j)
130             r = np.delete(r, j)
131             d = np.delete(d, j, 0)
132             d = np.delete(d, j, 1)
133             dx = np.delete(dx, j, 0)
134             dx = np.delete(dx, j, 1)
135             dy = np.delete(dy, j, 0)
136             dy = np.delete(dy, j, 1)
137             j -= 1
138             N -= 1
139         j += 1
140     i += 1
```

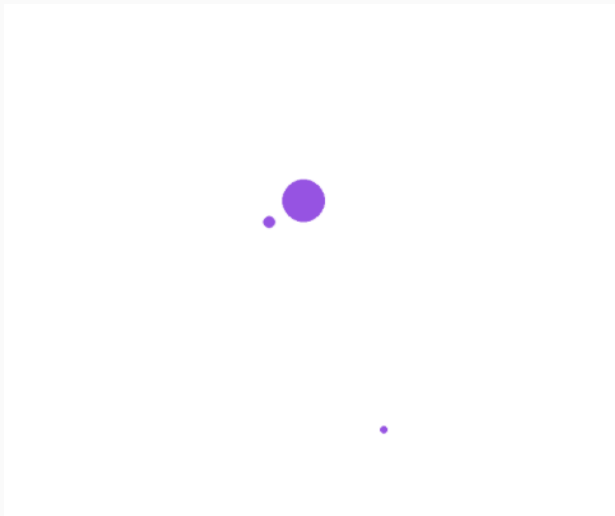
# График потенциальной энергии











## 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.

- 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

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.