

Группа М3201

К работе допущен _____

Студенты Ткачук С.А. и Чуб Д.О.

Работа выполнена _____

Преподаватель Шоев В.И.

Отчет принят _____

Отчет по моделированию № 1.1

Абсолютно упругое взаимодействие

1. Теоретическая часть

В данной работе изучается абсолютно упругое соударение двух тел. Используются законы сохранения импульсов тел для моделирования движения этих тел. В данной работе используется модель абсолютно упругого столкновения, которая предполагает, что энергия и импульс сохраняются во время столкновения.

2. Используемые формулы:

$$m_1 v_1 + m_2 v_2 = m_1 v'_1 + m_2 v'_2 \quad (1) - \text{закон сохранения импульса}$$

$$\frac{m_1 v_1^2}{2} + \frac{m_2 v_2^2}{2} = \frac{m_1 v'^2_1}{2} + \frac{m_2 v'^2_2}{2} \quad (2) - \text{закон сохранения энергии}$$

Используя эти формулы, можем получить измененные скорости:

$$v'_1 = \frac{(m_1 - m_2)v_1 + 2m_2 v_2}{m_1 + m_2}$$

$$v'_2 = \frac{(m_2 - m_1)v_2 + 2m_1 v_1}{m_1 + m_2}$$

3. Численный алгоритм для решения уравнений

Используется метод Эйлера для решения уравнения движения. Мы обновляем значения параметра координаты x на каждом временном шаге $\Delta t = 0,016$ и аппроксимируем траекторию движения.

4. Программный код на языке python:

```
import pygame

pygame.init()

window_height = 280
window_width = 1000
screen = pygame.display.set_mode((window_width, window_height))

font = pygame.font.Font(None, 28)

class Cube:
    def __init__(self, top_left_x, side_length, color, speed, mass):
        self.side_length = side_length
        self.top_left_x = top_left_x
        self.top_left_y = window_height - self.side_length
        self.color = color
        self.speed = speed
        self.mass = mass

cubel_speed = 0
cube2_speed = 100
cubel_mass = 1
cube2_mass = 1000

cubel = Cube(400, 30, (255, 255, 0), cubel_speed, cubel_mass)
cube2 = Cube(600, 60, (255, 0, 0), -cube2_speed, cube2_mass)

hits_counter_text = font.render('Количество ударов:', True, (0, 0, 0))

running = True
hits_count = 0

while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

    screen.fill((255, 255, 255))

    if cubel.top_left_x <= 0:
        hits_count += 1
        cubel.speed = -cubel.speed

    if cubel.top_left_x + cubel.side_length >= cube2.top_left_x:
        hits_count += 1
        new_speed0 = (((cubel.mass - cube2.mass) * cubel.speed) + 2 *
cube2.mass * cube2.speed) / (cubel.mass + cube2.mass)
        new_speed1 = (((cube2.mass - cubel.mass) * cube2.speed) + 2 *
cubel.mass * cubel.speed) / (cubel.mass + cube2.mass)
        cubel.speed = new_speed0
        cube2.speed = new_speed1
```

```

cubel.top_left_x = cubel.top_left_x + cubel.speed * 0.016
cube2.top_left_x = cube2.top_left_x + cube2.speed * 0.016

if cube2.top_left_x <= cubel.side_length:
    pygame.draw.rect(screen, cubel.color, (0, cubel.top_left_y,
cubel.side_length, cubel.side_length))
    pygame.draw.rect(screen, (0, 0, 0), (0, cubel.top_left_y,
cubel.side_length, cubel.side_length), 1)
    pygame.draw.rect(screen, cube2.color,
                        (cubel.side_length, cube2.top_left_y,
cube2.side_length, cube2.side_length))
    pygame.draw.rect(screen, (0, 0, 0), (cubel.side_length,
cube2.top_left_y, cube2.side_length, cube2.side_length), 1)
else:
    pygame.draw.rect(screen, cubel.color,
                        (cubel.top_left_x, cubel.top_left_y,
cubel.side_length, cubel.side_length))
    pygame.draw.rect(screen, (0, 0, 0), (cubel.top_left_x,
cubel.top_left_y, cubel.side_length, cubel.side_length), 1)
    pygame.draw.rect(screen, cube2.color,
                        (cube2.top_left_x, cube2.top_left_y,
cube2.side_length, cube2.side_length))
    pygame.draw.rect(screen, (0, 0, 0), (cube2.top_left_x,
cube2.top_left_y, cube2.side_length, cube2.side_length), 1)

screen.blit(hits_counter_text, (20, 20))
hits_count_text = font.render(str(hits_count), True, (0, 0, 0))
screen.blit(hits_count_text, (250, 20))

pygame.display.flip()
pygame.display.update()

pygame.time.Clock().tick(120)

pygame.quit()

print("Количество ударов:", hits_count)

```

5. Вывод

Представленный код демонстрирует применение численного метода Эйлера для моделирования столкновения двух кубов с использованием базовых принципов физики. Метод обеспечивает аппроксимацию движения объектов, используя законы сохранения энергии и импульса.