

# Выпускная квалификационная работа

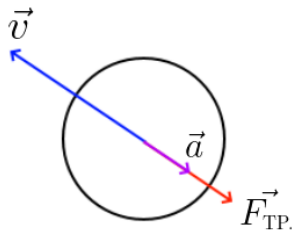
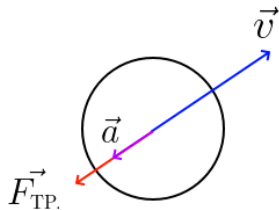
## Система физического моделирования на основе априорного подхода обнаружения столкновений

Владислав Прекель

ИКИТ СФУ  
КИ18-166

Красноярск  
20 июня 2022 г.

# Модель



# Формулы равноускоренного движения

Формулы для скорости (1) и положения тела (2):

$$\vec{v}(t) = \vec{v}_0 + \vec{a}t \quad (1)$$

$$x(t) = x_0 + v_{0x}t + \frac{a_x t^2}{2}, y(t) = y_0 + v_{0y}t + \frac{a_y t^2}{2} \quad (2)$$

где  $\vec{v}(t)$  – вектор скорости тела в момент времени  $t$ ;

$\vec{v}_0$  – вектор начальной скорости тела;

$\vec{a}$  – вектор ускорения тела;

$x(t)$  – координата тела в момент времени  $t$  по оси  $X$ ;

$x_0$  – координата начального положения тела по оси  $X$ ;

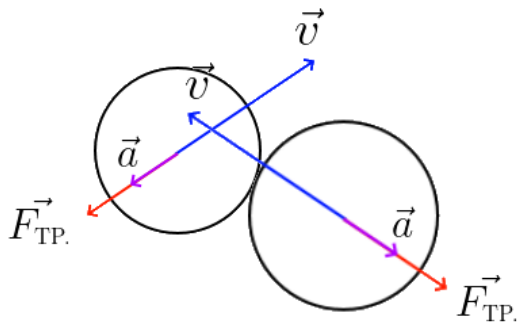
$y(t)$  – координата тела в момент времени  $t$  по оси  $Y$ ;

$y_0$  – координата начального положения тела по оси  $Y$ ;

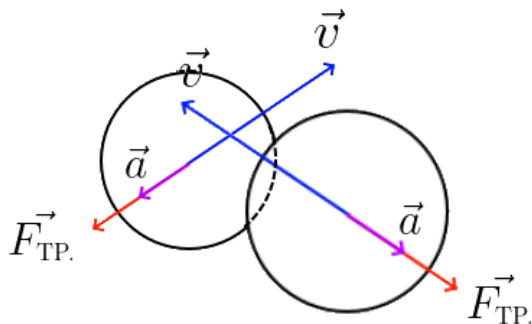
$a_x$  – проекция вектора ускорения тела  $\vec{a}$  на ось  $X$ ;

$a_y$  – проекция вектора ускорения тела  $\vec{a}$  на ось  $Y$ .

# Тела столкнулись



# Апостериорный подход



# Априорный подход

Основан на том, что можно найти время столкновения через уравнение (3):

$$distance(t) = r_1 + r_2 \quad (3)$$

где  $distance(t)$  – расстояние между центрами двух тел в момент времени  $t$ ;

$r_1$  – радиус первого тела;

$r_2$  – радиус второго тела.

**Целью выпускной квалификационной работы** является разработка физического движка, использующего априорный подход для обнаружения столкновений.

# Уравнение обнаружения столкновения двух тел

$$\sqrt{(x_1(t) - x_2(t))^2 + (y_1(t) - y_2(t))^2} = r_1 + r_2 \quad (4)$$

где  $x_1(t)$  – координата первого тела в момент времени  $t$  по оси  $X$ ;  
 $x_2(t)$  – координата второго тела в момент времени  $t$  по оси  $X$ ;  
 $y_1(t)$  – координата первого тела в момент времени  $t$  по оси  $Y$ ;  
 $y_2(t)$  – координата второго тела в момент времени  $t$  по оси  $Y$ ;  
 $r_1$  – радиус первого тела;  
 $r_2$  – радиус второго тела.



$$\begin{aligned}
& \frac{a_{x1}^2 - a_{x2}^2 + a_{y1}^2 - a_{y2}^2}{4} t^4 + \\
& + (v_{0x1} a_{x1} - v_{0x2} a_{x2} + v_{0y1} a_{y1} - v_{0y2} a_{y2}) t^3 + \\
& + (x_{01} a_{x1} - x_{02} a_{x2} + y_{01} a_{y1} - y_{02} a_{y2}) t^2 + \\
& + 2(x_{01} v_{0x1} - x_{02} v_{0x2} + y_{01} v_{0y1} - y_{02} v_{0y2}) t + \quad (5) \\
& + v_{0x1}^2 + x_{01}^2 - v_{0x2}^2 - x_{02}^2 + v_{0y1}^2 + y_{01}^2 - v_{0y2}^2 - y_{02}^2 - r_1^2 - 2r_1 r_2 - r_2^2 = 0
\end{aligned}$$

где  $x_{01}$  – начальная координата первого тела по оси  $X$ ;

$x_{02}$  – начальная координата второго тела по оси  $X$ ;

$y_{01}$  – начальная координата первого тела по оси  $Y$ ;

$y_{02}$  – начальная координата второго тела по оси  $Y$ ;

$v_{0x1}$  – проекция вектора начальной скорости I тела на ось  $X$ ;

$v_{0y1}$  – проекция вектора начальной скорости I тела на ось  $Y$ ;

$v_{0x2}$  – проекция вектора начальной скорости II тела на ось  $X$ ;

$v_{0y2}$  – проекция вектора начальной скорости II тела на ось  $Y$ ;

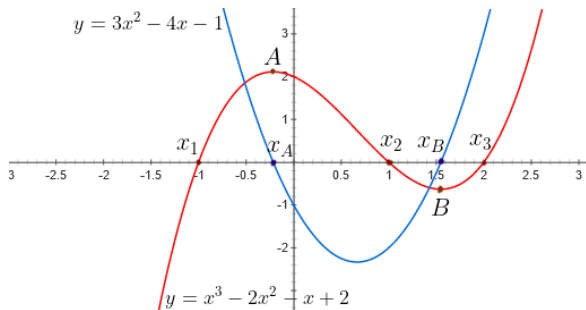
$a_{x1}$  – проекция вектора ускорения I тела на ось  $X$ ;

$a_{y1}$  – проекция вектора ускорения I тела на ось  $Y$ ;

$a_{x2}$  – проекция вектора ускорения II тела на ось  $X$ ;

$a_{y2}$  – проекция вектора ускорения II тела на ось  $Y$ .

# Метод численного решения алгебраических уравнений



Рассмотрим на примере уравнения  $x^3 - 2x^2 - x + 2 = 0$  (красное).  
Уравнение производной  $3x^2 - 4x - 1 = 0$  (синее).

Его корни:  $x_{A,B} = \frac{4 \pm \sqrt{28}}{6}$ .

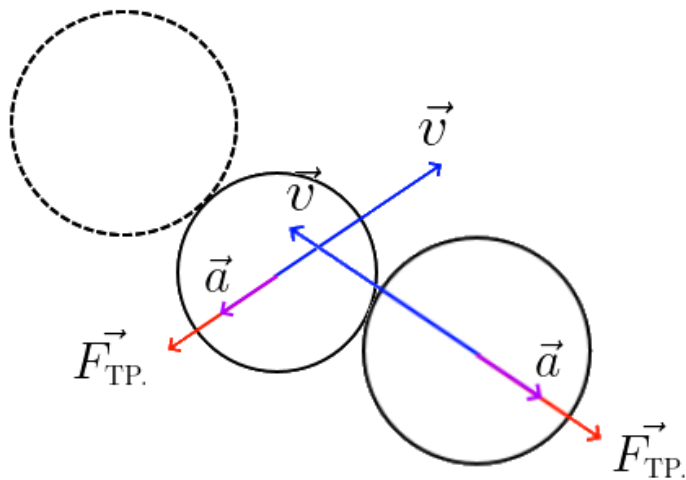
Тогда, корни исходного уравнения можно найти методом бисекции:

$x_1$  на промежутке  $(-\infty; \frac{4 - \sqrt{28}}{6}]$ , будет равен  $-1$

$x_2$  на промежутке  $[\frac{4 - \sqrt{28}}{6}; \frac{4 + \sqrt{28}}{6}]$ , будет равен  $1$

$x_3$  на промежутке  $[\frac{4 + \sqrt{28}}{6}; +\infty)$ , будет равен  $2$

# Выбор нужного корня



# Уравнение обнаружения столкновения с точкой

$$\sqrt{(x(t) - p_x)^2 + (y(t) - p_y)^2} = r \quad (6)$$

где  $x(t)$  – координата положения тела по оси  $X$ ;

$y(t)$  – координата положения тела по оси  $Y$ ;

$r$  – радиус тела;

$p_x$  – координата точки по оси  $X$ ;

$p_y$  – координата точки по оси  $Y$ .

# Уравнение обнаружения столкновения с прямой

$$\frac{|Ax(t) + By(t) + C|}{\sqrt{A^2 + B^2}} = r \quad (7)$$

где  $A, B, C$  – коэффициенты общего уравнения прямой;

$r$  – радиус тела;

$x(t), y(t)$  – координаты тела в момент времени  $t$ .

# Обработка ударов

$$\vec{v}'_1 = \vec{v}_1 - \frac{2m_2}{m_1 + m_2} \frac{\langle \vec{v}_1 - \vec{v}_2, \vec{r}_1 - \vec{r}_2 \rangle}{|\vec{r}_1 - \vec{r}_2|^2} (\vec{r}_1 - \vec{r}_2) \quad (8)$$

где  $\langle, \rangle$  – скалярное произведение векторов;

$\vec{v}'_1$  – вектор скорости первого тела после удара;

$\vec{v}_1$  – вектор скорости первого тела до удара;

$\vec{v}_2$  – вектор скорости второго тела до удара;

$\vec{r}_1$  – радиус-вектор положения первого тела;

$\vec{r}_2$  – радиус-вектор положения второго тела;

$m_1$  – масса первого тела;

$m_2$  – масса второго тела.

# Использованные технологии


- ▶ OCaml – язык программирования;
- ▶ Js\_of\_ocaml – компилятор OCaml в JavaScript;
- ▶ Lwt – библиотека для конкурентного программирования;
- ▶ Core – стандартная библиотека;
- ▶ Dream – web-фреймворк;
- ▶ ppx\_inline\_test, ppx\_expect – библиотеки юнит-тестирования;
- ▶ Sexplib – библиотека для сериализации и десериализации S-выражений;
- ▶ Bulma – CSS-фреймворк;
- ▶ Dune, opam – система сборки и пакетный менеджер;
- ▶ VS Code, OCaml Platform – среда разработки и плагин для работы с OCaml.

# Реализация движка

```
S.Model.init ~g:1.  
|> S.Engine.recv ~action:{ time = 0.  
    ; action =  
        AddBody { id = Some id1; x0 = 350.; y0 = 200.  
                ; r = 100.; mu = 1.; m = 10. }  
    ; until = { timespan = Some 0.; quantity = None }}  
|> S.Engine.recv ~action:{ time = 0.  
    ; action =  
        AddBody { id = Some id2; x0 = 700.; y0 = 200.  
                ; r = 100.; mu = 1.; m = 10. }  
    ; until = { timespan = Some 0.; quantity = None }}  
|> S.Engine.recv ~action:{ time = 0.  
    ; action = GiveVelocity { id = id2; v0 = -100., 0. }  
    ; until = { timespan = None; quantity = None }}
```



# Интерактивная демонстрация возможностей движка

chapgame 

Offline Online

**State**  

Copy/Paste Clear Stats

**Time** 30.28  

-1 0 +1

 30.27

**Calculation** ∞  

☒ Time 10.00  
☒ Quantity 25

**Speed** 1.00  

-0.1 -1x || 1x +0.1

 1.00

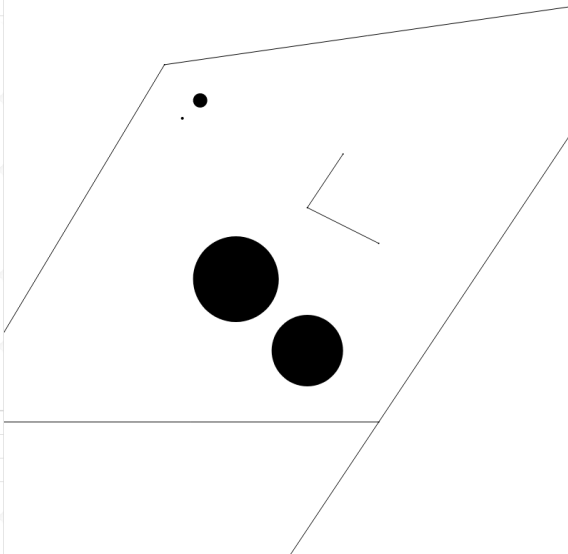
**Bodies**

Add


id		x	y	$\mu$	m	r
0		500.00	500.00	30.00	10800.00	60.00
1		600.00	600.00	20.00	7500.00	50.00
2		450.00	250.00	10.00	300.00	10.00
3		425.00	275.00	50.00	12.00	2.00

**Lines**

Add



<https://prekel.github.io/chapgame/>

chapgame 

Offline Online

**State**  

Copy/Paste Clear Stats

**Time** 38.80  

-1 0 +1

 38.78

**Calculation** 47.87  

☒ Time 10.00  
☒ Quantity 25

**Speed** 1.00  

-0.1 -1x || 1x +0.1

 1.00

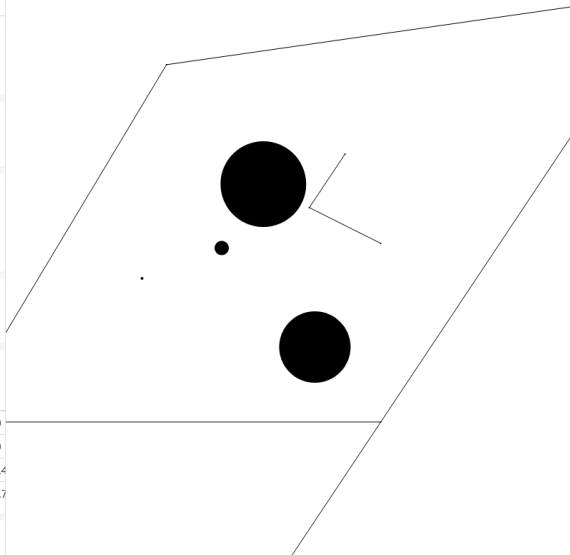
**Bodies**

Add

	$\mu$	$m$	$r$	$v_x$	$v_y$	$ v $
06	30.00	10800.00	60.00	0.00	0.00	0.00
08	20.00	7500.00	50.00	0.00	0.00	0.00
01	10.00	300.00	10.00	176.02	-220.92	282.4
04	50.00	12.00	2.00	534.77	-341.99	634.7

**Lines**

Add



# Решённые задачи

- ▶ определена модель и математическая база, требующуюся для моделирования;
- ▶ программно реализован физический движок и интерактивная демонстрация его работы.

# Выпускная квалификационная работа

## Система физического моделирования на основе априорного подхода обнаружения столкновений

Владислав Прекель

ИКИТ СФУ  
КИ18-166

Красноярск  
20 июня 2022 г.