

Санкт-Петербургский политехнический университет Петра Великого
Институт физики, нанотехнологий и телекоммуникаций
Высшая инженерно-физическая школа

КУРСОВОЙ ПРОЕКТ
«Визуализация бильярда
в среде MATLAB»

по дисциплине «Моделирование физических задач в среде MATLAB»
направление 03.03.02 — «Физика»

Выполнил

студент гр. 3430302/70801

А. И. Бегун

Руководитель

доцент

В. П. Каасик

«___» _____ 2020 г.

Санкт-Петербург
2020

Оглавление

1. Постановка задачи.	3
2. Физическая постановка задачи.	3
3. Выбор метода решения.	5
4. Разработка алгоритма.	11
5. Составление программы.	12
6. Решение задачи и анализ результатов.	15
7. Заключение.	19
Список использованной литературы.	20

1. Постановка задачи.

Условие задачи:

Представьте и изобразите бильярд, в котором из любого положения шар можно закатить в любую из шести луз одним ударом кия по шару.

Итак, в качестве исходных данных пользователь указывает положение шара и лузу, в которую требуется его закатить. На выходе он получает схематичную визуализацию бильярдного стола (в виде ограниченной координатной сетки) и траекторий движения шара от его исходного положения до заданной лузы после одного удара кия по шару.

2. Физическая постановка задачи.

Будем решать задачу в предположении, что:

1. Шар и лузы точечные, то есть их размерами можно пренебречь.
2. Шар и поверхность бильярдного стола идеальны в том смысле, что их коэффициенты трения пренебрежимо малы.
3. Столкновения шара с бильярдными стенками – упругие, то есть проекция импульса шара на направление вдоль стенки и энергия шара сохраняются.

Для решения поставленной задачи введём двумерную декартову систему координат так, чтобы начало координат совпадало с левым нижним углом бильярдного стола, а оси были направлены вдоль нижней и левой боковой стенок (см. рис. 1).

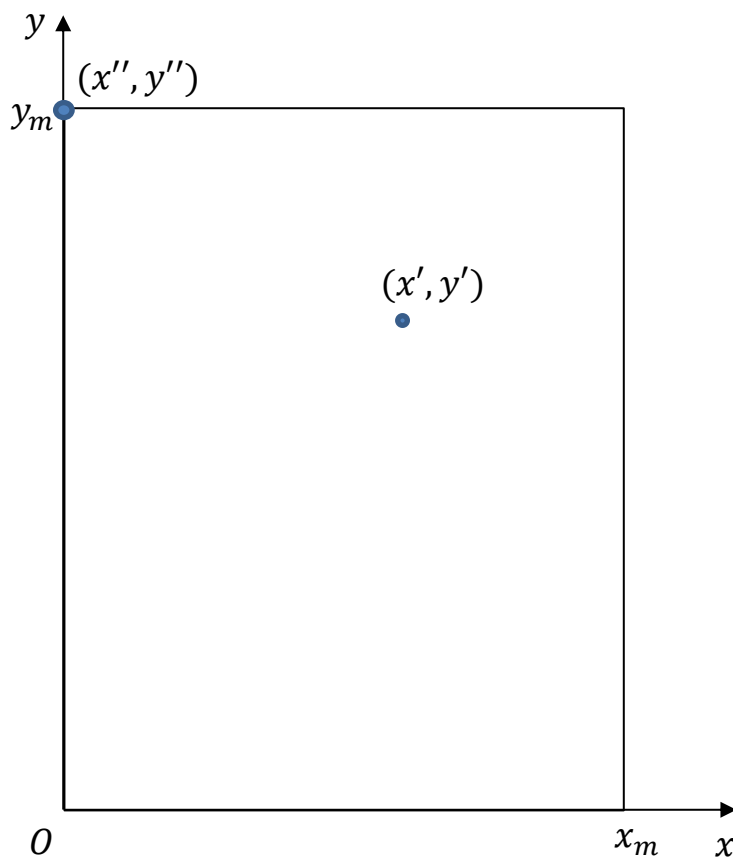


Рисунок 1. Система координат xOy и координаты шара и лузы.

В рамках сделанных предположений определим исходные данные (см. рис. 1):

1. Координаты исходного положения шара: (x', y') .
2. Координаты лузы: (x'', y'') .

Для простоты будем рассматривать случай, когда шар закатывается в лузу, при этом испытывая не больше одного соударения со стенкой бильярда.

Результат решения задачи должен представлять собой набор траекторий движения шара $y_i = y_i(x)$ при закатывании его в одну из шести луз одним ударом кия.

Понятно, что значения координат исходного положения шара, луз, движущегося шара ограничены размерами бильярдного стола (x_m, y_m) (см. рис. 1).

3. Выбор метода решения.

На рис. 2 показаны возможные траектории движения шара. Всегда реализуется ситуация, при которой шар движется от его исходного положения до лузы по прямой. Сразу можно записать уравнение этой траектории, как уравнение прямой, проходящей через две точки (x', y') и (x'', y'') :

$$\frac{y_1 - y''}{y' - y''} = \frac{x - x''}{x' - x''} \Leftrightarrow y_1(x) = \frac{y' - y''}{x' - x''} \cdot x + \frac{y''x' - y'x''}{x' - x''}.$$

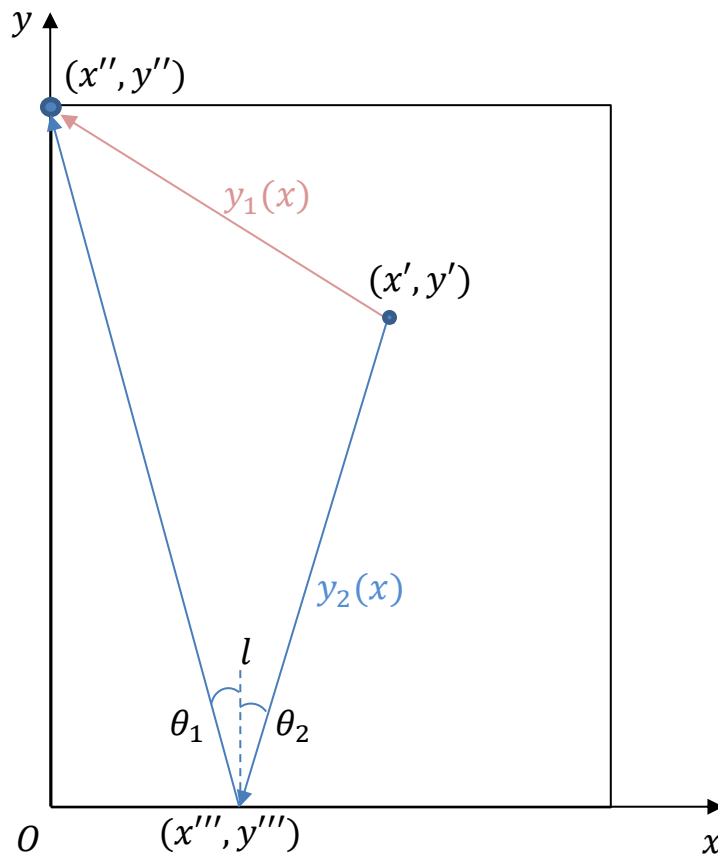


Рисунок 2. Возможные траектории движения шара при закатывании в лузу.

Также существуют траектории, реализуемые в результате столкновения шара со стенкой бильярда. Введём углы θ_1 и θ_2 . Так как удар упругий, выполняется закон сохранения энергии:

$$\frac{p_1^2}{2m} = \frac{p_2^2}{2m} \Rightarrow p_1 = p_2 = p,$$

где p_1 и p_2 – величины импульса шара до и после столкновения соответственно. Таким образом, величина импульса сохраняется.

Из закона сохранения проекции импульса на ось x :

$$-p \cdot \sin \theta_1 = -p \cdot \sin \theta_2 \Rightarrow \theta_1 = \theta_2.$$

То есть угол падения равен углу отражения. Следовательно, прямая l – биссектриса угла при вершине с координатами (x''', y''') . А задача о нахождении траектории сводится к нахождению уравнений двух прямых – прямой, проходящей через точки (x', y') и (x''', y''') :

$$\begin{aligned} \frac{y - y'''}{y' - y'''} = \frac{x - x'''}{x' - x'''} &\Leftrightarrow y = \frac{y' - y'''}{x' - x'''} \cdot x + \frac{y'''x' - y'x'''}{x' - x'''} \Leftrightarrow \\ &(y' - y''') \cdot x + (x''' - x') \cdot y + (y'''x' - y'x'''). \end{aligned}$$

и прямой, проходящей через точки (x''', y''') и (x'', y'') :

$$\begin{aligned} \frac{y - y''}{y''' - y''} = \frac{x - x''}{x''' - x''} &\Leftrightarrow y = \frac{y''' - y''}{x''' - x''} \cdot x + \frac{y''x''' - y'''x''}{x''' - x''} \Leftrightarrow \\ &(y''' - y'') \cdot x + (x'' - x''') \cdot y + (y''x''' - y'''x''). \end{aligned}$$

Координаты (x''', y''') найдём из условия перпендикулярности биссектрисы к стенке бильярда. Запишем уравнение биссектрисы:

$$\begin{aligned} &\frac{(y''' - y'') \cdot x + (x'' - x''') \cdot y + (y''x''' - y'''x'')}{\sqrt{(y''' - y'')^2 + (x'' - x''')^2}} \\ &= \pm \frac{(y' - y''') \cdot x + (x''' - x') \cdot y + (y'''x' - y'x''')}{\sqrt{(y' - y''')^2 + (x''' - x')^2}}. \quad (1) \end{aligned}$$

Из условия того, что биссектриса перпендикулярна стенке, вдоль которой направлена ось x , следует, что абсцисса биссектрисы имеет одно значение $x = x'''$. Подставим $x = x'''$ в (1):

$$\begin{aligned} &\frac{(y''' - y'') \cdot x''' + (x'' - x''') \cdot y + (y''x''' - y'''x'')}{\sqrt{(y''' - y'')^2 + (x'' - x''')^2}} \\ &= \pm \frac{(y' - y''') \cdot x''' + (x''' - x') \cdot y + (y'''x' - y'x''')}{\sqrt{(y' - y''')^2 + (x''' - x')^2}} \Leftrightarrow \end{aligned}$$

$$\begin{aligned}
\frac{y'''x''' + (x'' - x''') \cdot y - y'''x''}{\sqrt{(y''' - y'')^2 + (x'' - x''')^2}} &= \pm \frac{-y'''x''' + (x''' - x') \cdot y + y'''x'}{\sqrt{(y' - y''')^2 + (x''' - x')^2}} \Leftrightarrow \\
\frac{(x'' - x''') \cdot (y - y''')}{\sqrt{(y''' - y'')^2 + (x'' - x''')^2}} &= \pm \frac{(x''' - x') \cdot (y - y''')}{\sqrt{(y' - y''')^2 + (x''' - x')^2}} \Leftrightarrow \\
\frac{(x'' - x''')^2}{(y''' - y'')^2 + (x'' - x''')^2} &= \frac{(x''' - x')^2}{(y' - y''')^2 + (x''' - x')^2} \Leftrightarrow \\
\frac{(y''' - y'')^2}{(x'' - x''')^2} &= \frac{(y' - y''')^2}{(x''' - x')^2} \Leftrightarrow \\
\frac{(y''' - y'')}{(x'' - x''')} &= \pm \frac{(y' - y''')}{(x''' - x')} \Leftrightarrow \\
x'''(y''' - y'') - x'(y''' - y'') &= \pm x''(y' - y''') \mp x'''(y' - y''') \Leftrightarrow \\
x''' = \frac{x'(y''' - y'') \pm x''(y' - y''')}{(y''' - y'') \pm (y' - y''')} &\Leftrightarrow \begin{cases} x''' = \frac{x'(y''' - y'') - x''(y''' - y')}{y' - y''} \\ x''' = \frac{x'(y''' - y'') + x''(y''' - y')}{2y''' - y'' - y'} \end{cases} \cdot (2)
\end{aligned}$$

Получили два уравнения биссектрисы, но она должна задаваться только одним. Чтобы выбрать правильное, подставим конкретные значения координат (см. рис. 3).

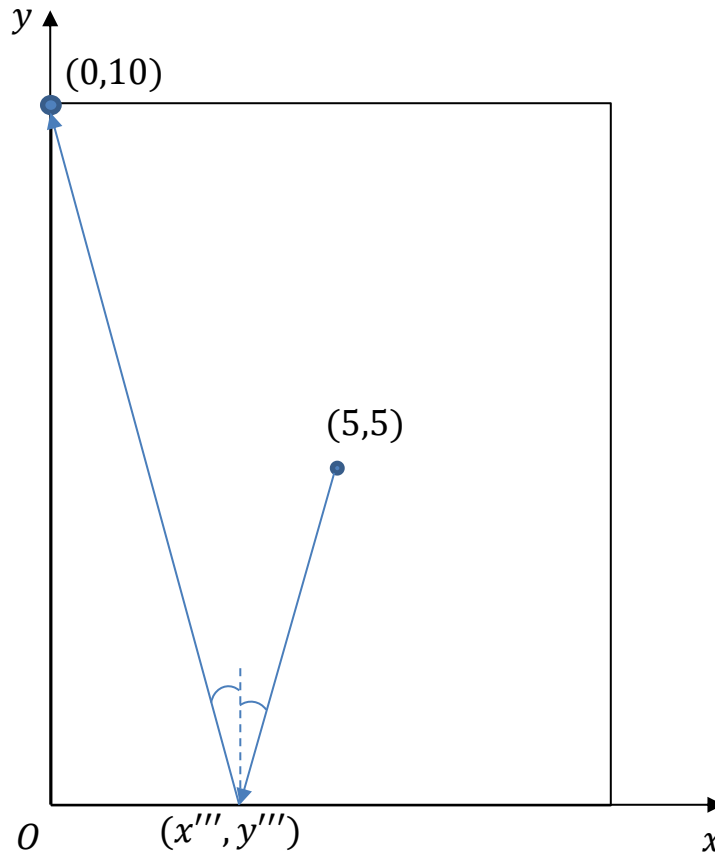


Рисунок 3. Конкретные значения координат шара и лузы.

Подстановка даёт:

$$\begin{cases} x''' = \frac{5 \cdot (-10)}{5 - 10} = 10 \\ x''' = \frac{5 \cdot (-10)}{-10 - 5} \approx 3,3 \end{cases} \quad (3)$$

Понятно, что в данном случае должно выполняться условие $x'' < x''' < x'$ и, следовательно, нам подходит второе значение в (3) и второе выражение в совокупности (2).

Итак, уравнение биссектрисы или абсцисса x''' точки соударения шара со стенкой задаётся выражением:

$$x''' = \frac{x'(y''' - y'') + x''(y''' - y')}{2y''' - y'' - y'}.$$

Причём это выражение описывает соударение с нижней стенкой, когда $y''' = 0$, и с верхней стенкой, когда $y''' = y_m$.

Чтобы получить выражения координат точек соударения с боковыми стенками, произведём замену $y \rightarrow x, x \rightarrow y$. Тогда ордината y''' точки соударения шара с боковой стенкой задаётся выражением:

$$y''' = \frac{y'(x''' - x'') + y''(x''' - x')}{2x''' - x'' - x'}.$$

Причём это выражение описывает соударение с левой стенкой, когда $x''' = 0$, и с правой стенкой, когда $x''' = x_m$.

Подведём итоги.

- 1) Шар из данного положения (x', y') может закатиться в заданную лузу (x'', y'') по прямой, задающейся уравнением:

$$y_1(x) = \frac{y' - y''}{x' - x''} \cdot x + \frac{y''x' - y'x''}{x' - x''}.$$

- 2) Шар из данного положения может закатиться в заданную лузу после одного соударения со стенкой бильярда. При этом он будет двигаться сначала по прямой, задаваемой уравнением

$$y = \frac{y' - y'''}{x' - x'''} \cdot x + \frac{y'''x' - y'x'''}{x' - x'''},$$

затем по прямой, задаваемой уравнением

$$y = \frac{y''' - y''}{x''' - x''} \cdot x + \frac{y''x''' - y'''x''}{x''' - x''}.$$

При этом, если

- a. Шар ударяется о нижнюю стенку, то

$$x''' = \frac{x'(y''' - y'') + x''(y''' - y')}{2y''' - y'' - y'}, y''' = 0.$$

- b. Шар ударяется о верхнюю стенку, то

$$x''' = \frac{x'(y''' - y'') + x''(y''' - y')}{2y''' - y'' - y'}, y''' = y_m.$$

с. Шар ударяется о левую стенку, то

$$y''' = \frac{y'(x''' - x'') + y''(x''' - x')}{2x''' - x'' - x'}, x''' = 0.$$

d. Шар ударяется о правую стенку, то

$$y''' = \frac{y'(x''' - x'') + y''(x''' - x')}{2x''' - x'' - x'}, x''' = x_m.$$

Понятно, с какими стенками будет происходить соударение при заданной лузе. Пронумеруем лузы, как показано на рисунке 4.

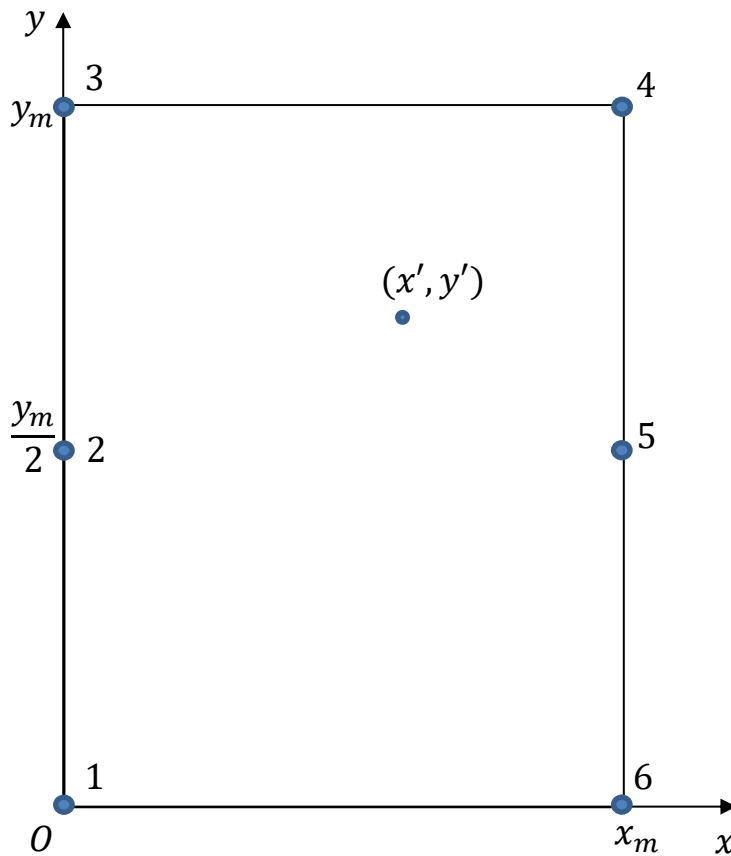


Рисунок 4. Нумерация луз.

Если шар требуется закатить в лузу:

1. №1 $((x'', y'') = (0,0))$, то шар, очевидно, может удариться о верхнюю стенку и правую стенку.

2. №2 ($(x'', y'') = (0, \frac{y_m}{2})$), то шар может удариться о верхнюю, нижнюю стенку и правую стенку.
3. №3 ($(x'', y'') = (0, y_m)$), то шар может удариться о правую стенку и нижнюю стенку.
4. №4 ($(x'', y'') = (x_m, y_m)$), то шар может удариться о левую стенку и нижнюю стенку.
5. №5 ($(x'', y'') = (x_m, \frac{y_m}{2})$), то шар может удариться о левую, нижнюю стенку и верхнюю стенку.
6. №6 ($(x'', y'') = (x_m, 0)$), то шар может удариться о левую стенку и верхнюю стенку.

4. Разработка алгоритма.

Для решения задачи создадим программу в виде функции `fun_billiard(xb, yb, n)`, которая будет получать в качестве аргументов значения координат исходного положения шара `xb, yb` и номер лузы, в которую нужно закатить шар – `n`.

В теле функции зададим размеры бильярдного стола `xm, ym`, шаг построения координатной сетки (влияет на точность построения) - `xh`. Далее построим прямоугольник, как схематичное изображение бильярдного стола, и оси. Также создадим массивы `x, y`, хранящие значения координат луз, и точками отметим лузы на координатной сетке.

Далее будет выполняться цикл `switch`, в котором в зависимости от значения номера лузы будут запускаться функции по построению траекторий в соответствии с положениями 1-6 из предыдущего раздела.

Функция

`straight_line(x(n), y(n), xm, ym, xh, xb, yb)` получает в качестве аргументов значения координат лузы, размеров бильярдного стола, шага построения координатной сетки, координат положения шара соответственно и выполняет построение прямой линии, соединяющей шар и лузу в соответствии с пунктом 1) предыдущего раздела. В теле функции также используется

функция `comet1(x1,y1,0)`, которая отличается от встроенной функции MATLAB цветом хвоста «кометы».

Функции `upper_wall(x(n),y(n),xm,ym,xh,xb,yb)`, `right_wall(x(n),y(n),xm,ym,xh,xb,yb)`, `bottom_wall(x(n),y(n),xm,ym,xh,xb,yb)`, `left_wall(x(n),y(n),xm,ym,xh,xb,yb)` получают в качестве аргументов те же значения, что и предыдущая функция и выполняют построение траекторий в соответствии с пунктом 2) предыдущего раздела.

5. Составление программы.

Ниже представлен детальный код основной программы-функции и пяти подпрограмм-функций.

```
function fun_billiard(xb,yb,n)

    %размеры бильярдного стола
    xm = 60; ym = 120;

    %шаг построения координатной сетки
    xh = 0.01;

    hold on
    rectangle('Position',[0 0 xm ym])
    axis([0 xm+10 0 ym+10])

    %массивы со значениями координат луз
    x = [0 0 0 xm xm xm];
    y = [0 ym/2 ym ym ym/2 0];

    plot(x,y,'o','Color',[0.47 0.52
0.8]','MarkerFaceColor',[0.47 0.52 0.8])

    switch n
    case 1
        straight_line(x(n),y(n),xm,ym,xh,xb,yb)
        upper_wall(x(n),y(n),xm,ym,xh,xb,yb)
        right_wall(x(n),y(n),xm,ym,xh,xb,yb)
    case 2
        straight_line(x(n),y(n),xm,ym,xh,xb,yb)
        upper_wall(x(n),y(n),xm,ym,xh,xb,yb)
        right_wall(x(n),y(n),xm,ym,xh,xb,yb)
```

```

        bottom_wall(x(n),y(n),xm,ym,xh,xb,yb)
    case 3
        straight_line(x(n),y(n),xm,ym,xh,xb,yb)
        right_wall(x(n),y(n),xm,ym,xh,xb,yb)
        bottom_wall(x(n),y(n),xm,ym,xh,xb,yb)
    case 4
        straight_line(x(n),y(n),xm,ym,xh,xb,yb)
        bottom_wall(x(n),y(n),xm,ym,xh,xb,yb)
        left_wall(x(n),y(n),xm,ym,xh,xb,yb)
    case 5
        straight_line(x(n),y(n),xm,ym,xh,xb,yb)
        upper_wall(x(n),y(n),xm,ym,xh,xb,yb)
        bottom_wall(x(n),y(n),xm,ym,xh,xb,yb)
        left_wall(x(n),y(n),xm,ym,xh,xb,yb)
    case 6
        straight_line(x(n),y(n),xm,ym,xh,xb,yb)
        upper_wall(x(n),y(n),xm,ym,xh,xb,yb)
        left_wall(x(n),y(n),xm,ym,xh,xb,yb)
    otherwise

end
end

```

Код функции straight_line(x(n),y(n),xm,ym,xh,xb,yb) :

```

function straight_line(x,y,xm,ym,xh,xb,yb)
    %кратчайшая траектория - прямая, соединяющая две
    точки
    if xb>x
        x1 = xb:-xh:x;
    else
        x1 = xb:xh:x;
    end
    y1 = (y-yb)/(x-xb)*x1 + (yb*x-y*xb)/(x-xb);
    comet1(x1,y1,0)
end

```

Код функции upper_wall(x(n),y(n),xm,ym,xh,xb,yb) :

```

function upper_wall(x,y,xm,ym,xh,xb,yb)
    %шар ударяется о верхнюю стенку бильярда
    yimp = ym;
    ximp = (x*(yimp-yb)+xb*(yimp-y))/(2*yimp-yb-y);
    if xb>ximp
        x41 = xb:-xh:ximp;
        x42 = ximp:-xh:x;
    end

```

```

else
    x41 = xb:xh:ximp;
    x42 = ximp:xh:x;
end
y41 = (yimp-yb)/(ximp-xb)*x41 + (yb*ximp-
yimp*xb)/(ximp-xb);
y42 = (y-yimp)/(x-ximp)*x42 + (yimp*x-y*ximp)/(x-
ximp);
x4 = [x41 x42];
y4 = [y41 y42];
comet2(x4,y4,0)
end

```

Код функции right_wall(x(n),y(n),xm,ym,xh,xb,yb) :

```

function right_wall(x,y,xm,ym,xh,xb,yb)
%шар ударяется о правую боковую стенку бильярда
ximp = xm;
yimp = (y*(ximp-xb)+yb*(ximp-x))/(2*ximp-xb-x);
if (yimp>=ym/2+1)||(yimp<=ym/2-1)
    x31 = xb:xh:xm;
    x32 = xm:-xh:x;
    y31 = (yimp-yb)/(ximp-xb)*x31 + (yb*ximp-
yimp*xb)/(ximp-xb);
    y32 = (y-yimp)/(x-ximp)*x32 + (yimp*x-
y*ximp)/(x-ximp);
    x3 = [x31 x32];
    y3 = [y31 y32];
    comet(x3,y3,0)
end
end

```

Код функции bottom_wall(x(n),y(n),xm,ym,xh,xb,yb) :

```

function bottom_wall(x,y,xm,ym,xh,xb,yb)
%шар ударяется о нижнюю стенку бильярда
yimp = 0;
ximp = (x*(yimp-yb)+xb*(yimp-y))/(2*yimp-yb-y);
if xb>ximp
    x21 = xb:-xh:ximp;
    x22 = ximp:-xh:x;
else
    x21 = xb:xh:ximp;
    x22 = ximp:xh:x;
end

```

```

        y21 = (yimp-yb)/(ximp-xb)*x21 + (yb*ximp-
yimp*xb)/(ximp-xb);
        y22 = (y-yimp)/(x-ximp)*x22 + (yimp*x-y*ximp)/(x-
ximp);
        x2 = [x21 x22];
        y2 = [y21 y22];
        comet2(x2,y2,0)
end

```

Код функции left_wall(x(n),y(n),xm,ym,xh,xb,yb) :

```

function left_wall(x,y,xm,ym,xh,xb,yb)
    %шар ударяется о левую боковую стенку бильярда
    ximp = 0;
    yimp = (y*(ximp-xb)+yb*(ximp-x))/(2*ximp-xb-x);
    if (yimp>=ym/2+1) || (yimp<=ym/2-1)
        x51 = xb:-xh:ximp;
        x52 = ximp:xh:x;
        y51 = (yimp-yb)/(ximp-xb)*x51 + (yb*ximp-
yimp*xb)/(ximp-xb);
        y52 = (y-yimp)/(x-ximp)*x52 + (yimp*x-
y*ximp)/(x-ximp);
        x5 = [x51 x52];
        y5 = [y51 y52];
        comet(x5,y5,0)
    end
end
end

```

6. Решение задачи и анализ результатов.

Приведём графическое решение задачи для нескольких случаев.

Размеры стола $(x_m, y_m) = (60, 120)$.

1. Положение шара $(x', y') = (30, 60)$ и луза №1. Результат работы программы см. на рис. 5.

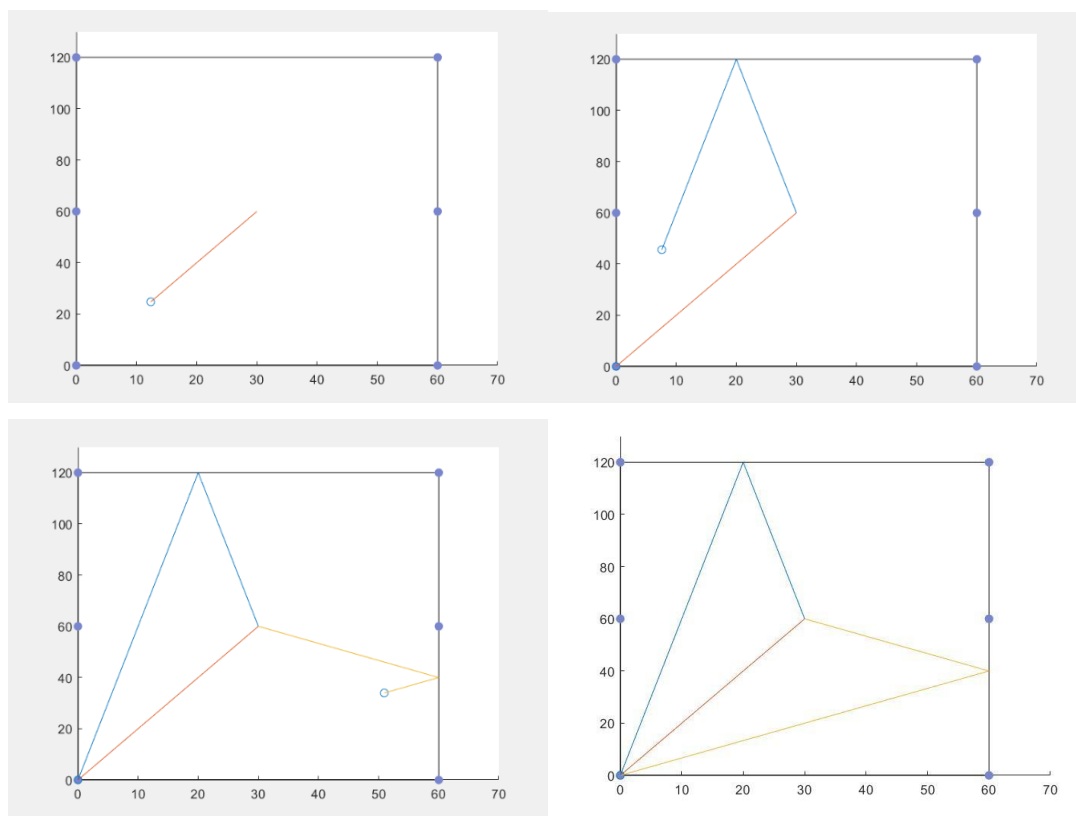


Рисунок 5. Процесс построения траекторий и конечный результат при попадании шара в лузу №1.

2. Положение шара $(x', y') = (30, 60)$ и луза №2. Результат работы программы см. на рис. 6.

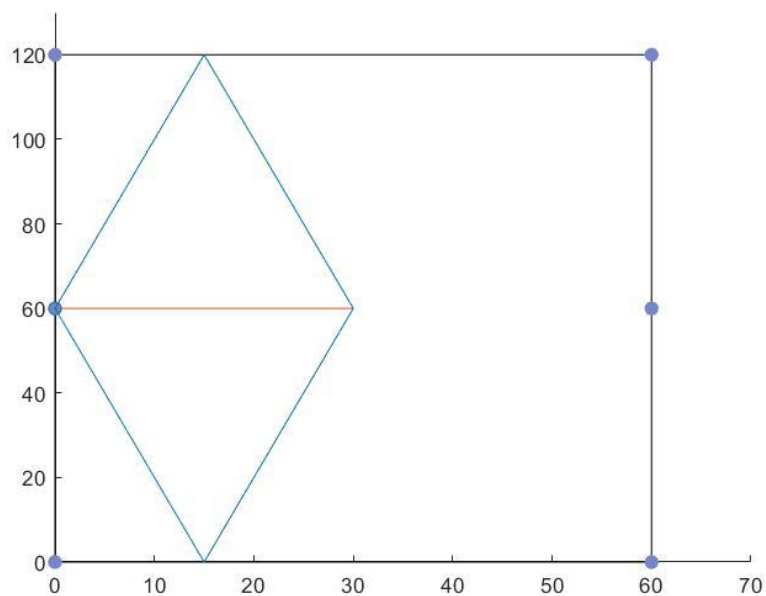


Рисунок 6. Траектории попадания шара в лузу №2.

3. Положение шара $(x', y') = (30, 40)$ и луза №3. Результат работы программы см. на рис. 7.

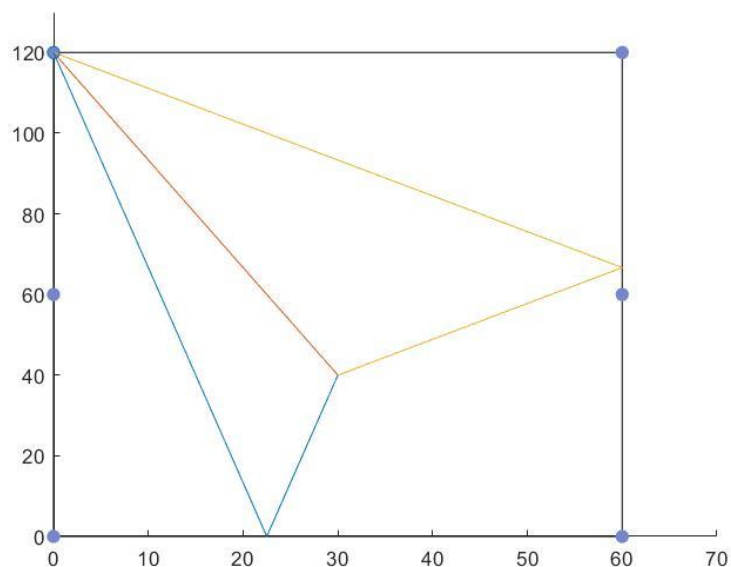


Рисунок 7. Траектории попадания шара в лузу №3.

4. Положение шара $(x', y') = (30, 30)$ и луза №4. Результат работы программы см. на рис. 8.

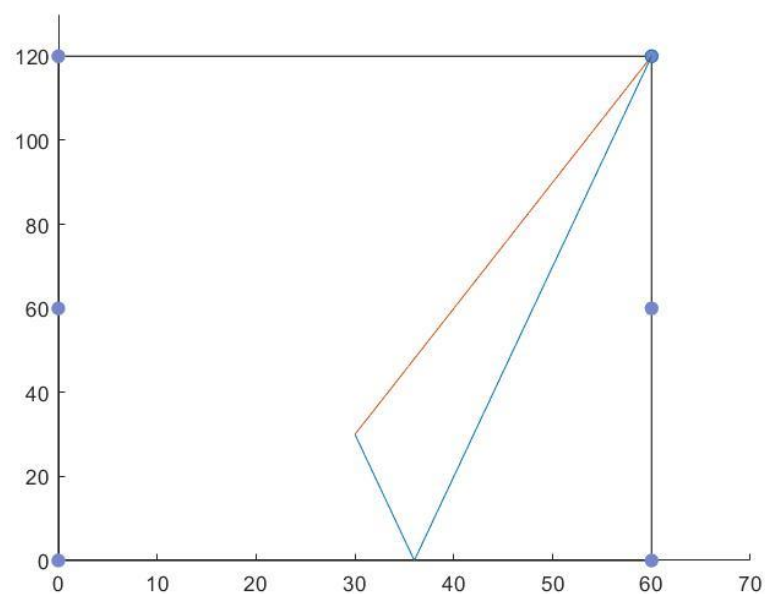


Рисунок 8. Траектории попадания шара в лузу №4.

Видно, что в данном случае не была построена траектория движения шара при соударении с левой стенкой, так как точка соударения совпала бы с лузой №2.

5. Положение шара $(x', y') = (30, 100)$ и луза №5. Результат работы программы см. на рис. 9.

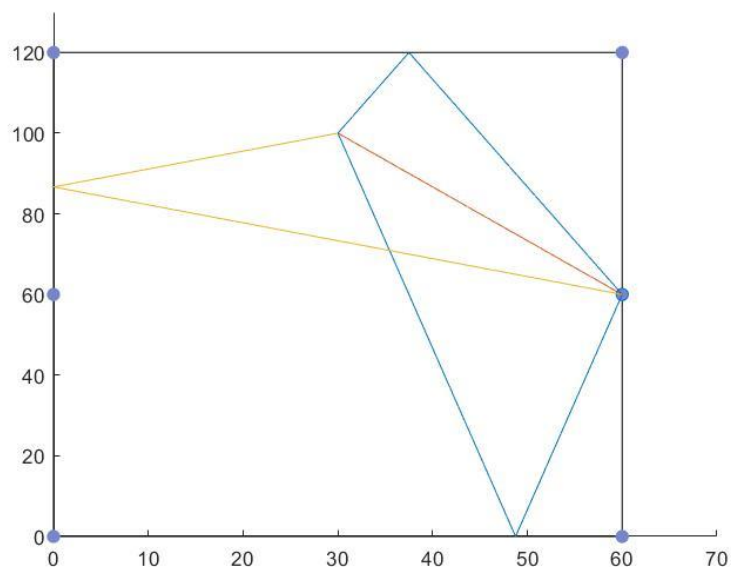


Рисунок 9. Траектории попадания шара в лузу №5.

6. Положение шара $(x', y') = (50, 10)$ и луза №6. Результат работы программы см. на рис. 10.

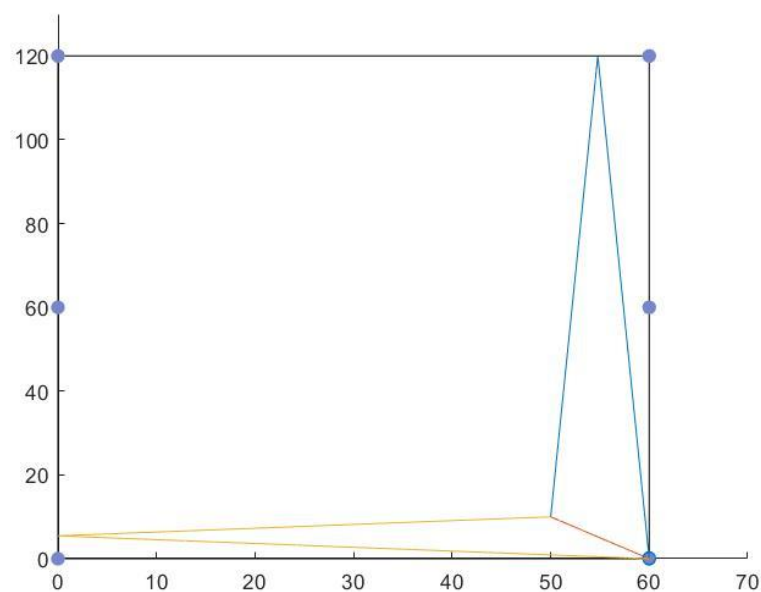


Рисунок 10. Траектории попадания шара в лузу №6.

Полученные результаты отвечают поставленной задаче.

7. Заключение.

В данной работе была решена задача о визуализации бильярда, в котором из любого положения шар можно закатить в любую из шести луз одним ударом кия по шару. Была написана программа MATLAB, которая позволила просчитать траектории движения шара и графически продемонстрировать их. Полученные результаты удовлетворяют поставленной задаче.

Список использованной литературы.

1. **Савельев И. В.** Курс общей физики. В 3 т. Т. 1. Механика. Молекулярная физика : учеб. пособие для втузов / И. В. Савельев. – 2-е изд., перераб. – М. : Наука, 2006. – 432 с.
2. **Потемкин В. Г.** Система MATLAB.// М.: Изд. “Диалог-МИФИ”, 1997.