

ЛАБОРАТОРНАЯ РАБОТА 13

КОМПЬЮТЕРНЫЕ МОДЕЛИ АНАЛИТИЧЕСКОЙ ГЕОМЕТРИИ. ГЕОМЕТРИЧЕСКИЙ ОБЪЕКТ «ПРЯМАЯ НА ПЛОСКОСТИ»

Компьютерная математика
БГУ, ММФ, 1 курс, КМ
доц. Малевич А.Э.,
доц. Щеглова Н.Л.
декабрь 2021

Постановка задачи

Требуется спроектировать и построить математическую систему, позволяющую решать задачи в области аналитической и компьютерной геометрии. Среда моделирования – символьный математический пакет *Mathematica*.

1 Внутреннее представление объекта «прямая на плоскости»

Задание 1. Спроектируйте внутреннее представление объекта «прямая на плоскости».

Выполнение задания 1

Поставим в соответствие математическому понятию «прямая» объект в *Mathematica* «прямая на плоскости».

В аналитической геометрии прямая l на плоскости может быть задана многими способами:

- 1) точкой $p_0 \in l$ и направляющим вектором $V \neq 0$ прямой l ;
- 2) координатами двух точек $a \in l, b \in l, a \neq b$;
- 3) точкой $p_0 \in l$ и нормальным вектором $N \neq 0$ прямой l ;
- 4) вектором параметров $F = [A, B, D]$ общего уравнения прямой $Ax + By + D = 0, A^2 + B^2 \neq 0$;
- 5) угловым коэффициентом k и точкой $p_0 \in l$;
- 6) длинами h_x, h_y отрезков (со знаками) в случае $h_x h_y \neq 0$, которые прямая l отсекает от координатных осей, если прямая l пересекает оси в точках $(h_x, 0), (0, h_y)$;
- 7) общим уравнением $Ax + By + D = 0, A^2 + B^2 \neq 0$;
- 8) каноническим уравнением $\frac{x - x_0}{m} = \frac{y - y_0}{n}$;
- 9) параметрическим уравнением $(x(t) = x_0 + V_x t, y(t) = y_0 + V_y t)$;
- 10) нормальным уравнением $x \cos \theta + y \sin \theta + p = 0$, где $(\cos \theta, \sin \theta)$ – координаты вектора нормали, p – длина перпендикуляра, опущенного на прямую из начала координат, θ – угол (измеренный в

положительном направлении) между положительным направлением оси Ox и направлением этого перпендикуляра;

11) уравнением с угловым коэффициентом $y = kx + b$;

12) уравнением прямой в отрезках $\frac{x}{h_x} + \frac{y}{h_y} = 1$, если $h_x, h_y \neq 0$;

13) уравнением в полярной системе координат.

Следует отметить, что в случаях 1) – 6) для задания прямой используется информация о других графических объектах: точке, векторе, отрезках, отсекаемых прямой. В случаях 7) – 13) информация о прямой линии находится в выражениях, которые мы называем уравнениями прямой.

Из всех возможных способов заданий прямой для представления объекта «прямая на плоскости» выберем тот способ, которые требует наименьшее количество информации об объекте, и для которого эта информация хранится в наиболее простой из возможных структур данных.

Таковыми данными о прямой располагает ее общее уравнение, а точнее – параметры A, B, C общего уравнения прямой $Ax + By + C = 0$. Кроме того, общее уравнение прямой известно еще со школьной программы, и еще его удобно использовать в алгебраических способах решения задач.

Внутренним представлением объекта «прямая на плоскости» в *Mathematica* будем называть выражение вида

$$\text{kmLine}[\text{km}, \text{id_String}, \{A_, B_, C_\}] \quad (1)$$

где $\{A, B, C\}$ – список, содержащий коэффициенты-параметры общего уравнения прямой $Ax + By + C = 0$.

Задание 2. Постройте булеву функцию **NamedQ**, позволяющую определить, имеет ли представленный объект «прямая» имя.

Выполнение задания 2

Задание 2 выполните **самостоятельно**. Функция должна тестировать структуру представления прямой (1) и возвращать булевы значения **True** или **False**.

2 Способы задания прямой

2.1 Выбор базовых конструкторов

Задание 3. Спроектируйте функции-конструкторы объекта «прямая на плоскости».

Выполнение задания 3

Проектируем конструкторы объекта «прямая на плоскости» следующим образом: выделим сначала базовые определения, а остальные

способы задания пропишем через базовые. Такой подход в том случае, если придется пере-/до- проектировать представление объекта, позволит ограничиться переделкой только базовых конструкторов.

Выявим способы, которыми мы будем задавать прямую. Задаваемой информацией может быть

- 1) имя прямой и ее общее уравнение;
- 2) общее уравнение прямой;
- 3) имя прямой, точка, ей принадлежащая, и направляющий вектор прямой;
- 4) точка, принадлежащая прямой, и направляющий вектор прямой;
- 5) имя прямой и две различные точки, ей принадлежащие;
- 6) две различные точки, принадлежащие прямой;
- 7) имя прямой, точка, ей принадлежащая, нормальный вектор к прямой;
- 8) точка, принадлежащая прямой, и нормальный вектор к прямой.

Построим левые части глобальных определений 1) – 8), формирующих представление объекта «прямая на плоскости». Введем обозначения, которые будем использовать в образцах: **id** – имя прямой, **equ** – общее уравнения прямой, **vars** – список переменных, где первый элемент указывает независимую переменную, второй – зависимую переменную для описания линейной функции, **P**, **P0**, **P1** – точки на плоскости, принадлежащие прямой, **Dir** – направляющий вектор прямой.

Описания образцов, соответствующих способам задания 1) – 8), сведем в таблицу 1.

Таблица 1

Образцы для задания объекта «прямая на плоскости»

	Именованный	Без имени
Общее уравнение прямой	1) <code>kmLine[id_String, A_. x_ + B_. y_ + C_. == 0, {x_Symbol, y_Symbol}]</code>	2) <code>kmLine[equ: (== 0), vars: {_Symbol, _Symbol}]</code>
Точка прямой и направление	3) <code>kmLine[id_String, P0_kmPoint, Dir_kmVector]</code>	4) <code>kmLine[P_kmPoint, Dir_kmVector]</code>
Две различные точки прямой	5) <code>kmLine[id_String, P1_kmPoint, P2_kmPoint]</code>	6) <code>kmLine[P1_kmPoint, P2_kmPoint]</code>
Точка прямой и нормаль	7) Написать самостоятельно	8) Написать самостоятельно

Выберем два базовых способа задания прямой: случай 1) и случай 3). Тогда способ 2) задания прямой сведем к способу 1), а способы задания 4)

– 8) будут выражены с использованием конструктора, соответствующего способу задания 3).

2.2 Прямая задана общим уравнением

Задание 4. Обеспечьте возможность задания объекта «прямая на плоскости», указывая имя прямой и ее общее уравнение.

Выполнение задания 4

Требуется построить конструктор **kmLine** объекта «прямая на плоскости» для случая, когда прямая задается именем и общим уравнением, записанным в стандартной математической нотации. Результатом работы конструктора должно быть представление прямой вида (1).

В общем уравнении прямой $Ax + By + C = 0$ коэффициенты A, B, C могут быть параметрами, и в этом случае *Mathematica* должна отличать символы-параметры от символов-переменных x и y . Именно поэтому при построении образца, левой части глобального правила, укажем три аргумента: имя прямой, общее уравнение прямой, список имен переменных в общем уравнении. Список содержит два аргумента: первый элемент списка – независимая переменная, второй – зависимая переменная линейной функции.

```
kmLine[id_String, A_. x_ + B_. y_ + C_. == 0,           (2)
  {x_Symbol, y_Symbol}] :=
kmLine[km, id, {A, B, C}] /;
  ¬ PossibleZeroQ[Abs@A + Abs@B]
```

Следует отметить, что мы предусмотрели возможность задания общего уравнения прямой относительно переменных, указываемых любыми символами. Разобраться, которые из символов играют роль переменных, и какова эта роль, позволяет третий входной аргумент – список, где первый элемент списка ассоциируется с независимой переменной x , второй – с переменной y общего уравнения прямой $Ax + By + C = 0$.

Убеждаемся в том, что правило работает адекватно нашей логике

```
kmLine["L1", 5 f - a == 0, {f, a}]
kmLine[km, L1, {5, -1, 0}]
```

```
kmLine["L1", 5 f - a == 0, {a, f}]
kmLine[km, L1, {-1, 5, 0}]
```

Продолжим тестировать конструктор. Следует проверить также неполные уравнения, так называемые вырожденные случаи. Зададим прямую, лежащую на оси OY

```
kmLine["L2", 5 f == 0, {f, a}]
kmLine[L2, 5 f == 0, {f, a}]
```

В этом случае конструктор не отработал. Это произошло потому, что уравнение прямой, указанное в качестве второго аргумента конструктора (2), не соответствует описанному образцу. Убеждаемся в этом непосредственно

```
MatchQ[5 f == 0, A_. x_ + B_. y_ + C_. == 0]
False
```

Интересно отметить, что как только мы совершим параллельный перенос прямой, то ее уравнение будет соответствовать образцу, описанному в (2) (**объясните, почему**)

```
MatchQ[5 f + 9 == 0, A_. x_ + B_. y_ + C_. == 0]
True
```

но конструктор снова не строит представление прямой (**объясните, почему**)

```
kmLine["L2", 5 f + 7 == 0, {f, a}]
kmLine[L2, 7 + 5 f == 0, {f, a}]
```

Опишем вырожденные случаи отдельными правилами (**объясните, почему** в описании третьего аргумента правила именован только один элемент списка)

```
kmLine[id_String, A_. x_ + C_. == 0,                                     (3)
  {x_Symbol, _Symbol}] := kmLine[km, id, {A, 0, C}];
kmLine[id_String, B_. y_ + C_. == 0,
  {_Symbol, y_Symbol}] := kmLine[km, id, {0, B, C}];
```

и проверим выполнение введенных правил

```
kmLine["L2", 5 f + 7 == 0, {f, a}]
kmLine[km, L2, {5, 0, 7}]
```

```
kmLine["L2", 5 f == 0, {f, a}]
kmLine[km, L2, {5, 0, 0}]
```

```
kmLine["L2", 5 f == 0, {a, f}]
kmLine[km, L2, {0, 5, 0}]
```

```
kmLine["L3", 5 == 0, {f, a}]
kmLine[L3, False, {f, a}]
```

В последнем из тестовых примеров *Mathematica* не смогла представить заданную «прямую». Это произошло потому, что первый аргумент после вычисления представлен символом **False**, и он не удовлетворяет ни одному из видов образцов, указанных в (2), (3).

Подстрахуемся от возможного ошибочного программного представления прямой в случае, когда коэффициенты при переменных равны нулю. Укажем, что в представлении общего уравнения прямой по крайней мере один из коэффициентов при **x** или **y** должен быть отличен от нуля, иначе вычисления должны быть прекращены

```
kmLine[km, "", {A_, B_, _}] := (4)
Abort[] /; PossibleZeroQ[Abs@A + Abs@B]
```

```
kmLine[km, "", {0, 0., 3}]
$Aborted
```

Выражения (2), (3), (4) составляют базовый конструктор прямой. Они вычисляют представление прямой, когда прямая задана именем и общим уравнением.

Задание 5. Обеспечьте возможность задания объекта «прямая на плоскости», указывая общее уравнение прямой.

Выполнение задания 5

Случай, когда прямая задана общим уравнением, мы сведем к базовому конструктору, состоящему из глобальных определений (2), (3), (4).

```
kmLine[equ : ( __ == 0 ), vars : { _Symbol, _Symbol } ] := (5)
kmLine["", equ, vars]
```

Проверим работу конструктора

```
kmLine[5 f + 3 == 0, {f, a}]
kmLine[km, , {5, 0, 3}]
```

2.3 Прямая через точку в направлении

Задание 6. Обеспечьте возможность задания объекта «прямая на плоскости» указанием точки, принадлежащей прямой, и направляющего вектора прямой, в случаях, когда прямая именована либо нет.

Выполнение задания 6

Требуется написать глобальное определение, вычисляющее представление прямой вида (1).

Решим задачу двумя различными способами.

Способ 1.

Способ 1 заключается в построении общего уравнения прямой и вызове базового конструктора (2) - (4) с указанием этого уравнения.

Пусть заданы точка p_0 , принадлежащая прямой l , и направляющий вектор прямой $V(Vx, Vy)$. Рассмотрим также $p(x, y)$ – произвольную точку прямой l . Требуется построить выражение, вычисляющее общее уравнение прямой.

Если точка $p(x, y)$ принадлежит прямой, то векторы V и $(p - p_0)$ коллинеарны и, определитель матрицы, содержащий координаты этих векторов, должен удовлетворять условию

$$\begin{vmatrix} Vx & Vy \\ -p_0x + x & -p_0y + y \end{vmatrix} = 0 \quad (6)$$

Представим выражение (6) в *Mathematica*. Построим объекты «точка на плоскости» и «вектор на плоскости»

```
{P, P0} = kmPoint /@ {{x, y}, {p0x, p0y}}
{kmPoint[km, , {x, y}], kmPoint[km, , {p0x, p0y}]}
```

```
V = kmVector[{Vx, Vy}]
kmVector[km, , {Vx, Vy}]
```

Сформируем матрицу, определитель которой нужно вычислить

```
{V@"coord", P@"coord" - P0@"coord"} // MatrixForm

$$\begin{pmatrix} Vx & Vy \\ -p_0x + x & -p_0y + y \end{pmatrix}$$

```

Тогда выражение, представляющее общее уравнение прямой в случае, когда она задана точкой $p_0 \in l$ и направляющим вектором V , запишется в виде

$$\text{Det}[\{V@{\text{"coord"}}, P@{\text{"coord"}} - P0@{\text{"coord"}}\}] == 0 \quad (7)$$

$$-p0y Vx + p0x Vy - Vy x + Vx y == 0$$

Здесь x и y – координаты произвольной точки, принадлежащей прямой. Теперь можно задать прямую, обратившись к базовому конструктору (2) – (3)

$$\text{kmLine}[\text{id_String}, P0_kmPoint, Dir_kmVector] := \quad (8)$$

$$\text{kmLine}[\text{id},$$

$$\text{Det}[\{Dir@{\text{"coord"}}, \{x, y\} - P0@{\text{"coord"}}\}] == 0,$$

$$\{x, y\}]$$

$$\text{kmLine}["L", \text{kmPoint}[\{5, 5\}], \text{kmVector}[\{-1, 1\}]]$$

$$\text{kmLine}[\text{km}, L, \{-1, -1, 10\}]$$

Способ 2.

Способ 2 сводится к вычислению коэффициентов A , B , C общего уравнения прямой $Ax + By + C = 0$ и непосредственному построению представления (1) прямой линии: в этом случае глобальное правило будет играть роль базового конструктора.

Анализируем результат вычисления выражения (7). Сопоставляя (7) с общим уравнением прямой, получим

$$A = -Vy, \quad (9)$$

$$B = Vx,$$

$$C = -p0y Vx + p0x Vy.$$

Вектор с координатами $(-Vy, Vx)$ является нормальным вектором N к прямой. Его координаты можно получить, построив матрицу поворота и действуя ею на направляющий вектор прямой

$$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \cdot V[{\text{"coord"}}]$$

$$\{-Vy, Vx\}$$

Значение параметра C , записанное в (9), может быть вычислено с использованием скалярного произведения радиус-вектора точки $p0 \in l$ и вектора нормали N , а именно $C = -(p0, N)$.

С учетом вышесказанного представим в *Mathematica* выражение, вычисляющее параметр C

$$-\left(\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \cdot \mathbf{v}@\text{"coord"}\right) \cdot \mathbf{P0}@\text{"coord"} \\ -p0y Vx + p0x Vy$$

Теперь формируем искомый список **{A, B, C}**, необходимый для представления вида (1) объекта «прямая на плоскости». Громоздкое выражение для вычисления координат вектора нормали запишем только раз, используя чистую функцию

```
Append[# , -#.P0["coord"]] &[ $\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \cdot \mathbf{v}["coord"]]$ 
{-Vy, Vx, -p0y Vx + p0x Vy}.
```

Теперь конструктор прямой **kmLine**, представляющий прямую через точку и направляющий вектор, можно построить как базовый: сформировать представление прямой (1) непосредственно в правой части правила (10)

```
kmLine[id_String, P0_kmPoint, Dir_kmVector] := (10)
kmLine[km, id,
Append[# , -#.P0["coord"]] &[
 $\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \cdot \mathbf{Dir}["coord"]]$ ]]
```

Проверим работу конструктора (10)

```
kmLine["", P0, V]
kmLine[km, , {-Vy, Vx, -p0y Vx + p0x Vy}]

kmLine["", kmPoint[{2, 3}], kmVector[{10, 20}]]
kmLine[km, , {-20, 10, 10}]
```

Конструктор (10) создает объект «прямая на плоскости» в случае, когда заданы имя прямой, точка, принадлежащая прямой и направляющий вектор прямой. Осталось построить правило для задания прямой через точку и направляющий вектор в случае, когда имя прямой непосредственно не указывается

```
kmLine[P_kmPoint, Dir_kmVector] := kmLine[
    If[And@@ (NamedQ /@ {P, Dir}),
        StringJoin[P@"id", Dir@"id"], ""],
    P, Dir] (11)
```

```
kmLine[P0, V]
kmLine[km, , {-Vy, Vx, -p0y Vx + p0x Vy}]
```

```
P1 = kmPoint["A", {Ax, Ay}]
kmPoint[km, A, {Ax, Ay}]
```

```
a = kmVector["a", {ax, ay}]
kmVector[km, a, {ax, ay}]
```

```
kmLine[P1, a]
kmLine[km, Aa, {-ay, ax, Ax ay - ax Ay}]
```

2.4 Прямая через две точки

Задание 7. Создайте конструкторы объекта «прямая на плоскости», если прямая l , именованная либо нет, задана двумя различными точками, ей принадлежащими.

Выполнение задания 7

Задание 7 выполните **самостоятельно**. Используйте конструкторы, позволяющие задать прямую через точку и направляющий вектор прямой.

Конструктор построит экземпляр объекта «прямая на плоскости», если задать две не совпадающие точки плоскости, принадлежащие этой прямой

```
kmLine["LL", kmPoint[{2, 3}], kmPoint[{10, 20}]]
kmLine[km, LL, {-17, 8, 10}]
```

2.5 Связь вектора нормали и направляющего вектора прямой

Пусть прямая l на плоскости задана точкой $p_0 \in l$ и направляющим вектором $V \neq 0$ прямой. Нормальный вектор N прямой может быть направлен двояко: N_L перпендикулярно влево или N_R перпендикулярно вправо по отношению к направляющему вектору V , если смотреть из начала в конец вектора V .

Какова связь между координатами V и N в каждом случае?

Задание 8. Изобразите на плоскости векторы V , N_L , N_R и определите связь между координатами векторов V и N_L , V и N_R . Запишите полученные соотношения, обоснуйте их.

Выполнение задания 8

В результате выполнения задания 8 координаты нормальных векторов к прямой должны быть выражены через координаты направляющего вектора прямой.

А именно, для произвольного направляющего вектора прямой $V(Vx, Vy)$ векторы, перпендикулярные ему, имеют следующие координаты: лево ориентированная нормаль к прямой $N_L(-Vy, Vx)$ и право ориентированная нормаль к прямой $N_R(Vy, -Vx)$.

Зафиксируем *левую ориентацию нормали N к прямой l* , и везде далее будем придерживаться этого выбора. Это означает, что вектор нормали N направлен перпендикулярно влево от направляющего вектора прямой V , если смотреть из начала вектора V в его конец. Будем обозначать эту нормаль N_L , ей соответствует право ориентированный направляющий вектор прямой V_R .

Задание 9. Представьте в *Mathematica* лево ориентированную нормаль N_L к прямой l , если задан направляющий вектор V прямой l .

Выполнение задания 9

Из задания 8 следует, что для направляющего вектора прямой $V(Vx, Vy)$ лево ориентированная нормаль к этой прямой имеет координаты $N_L(-Vy, Vx)$.

Пусть задан направляющий вектор прямой l

```
V = kmVector[{Vx, Vy}]
kmVector[km, , {Vx, Vy}]
```

Для вычисления координат нормального вектора N_L к прямой, лево ориентированного по отношению к направляющему вектору V прямой l , используем матрицу поворота

```

$$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \cdot v@"coord"$$

{-Vy, Vx}
```

Тогда искомый вектор нормали к прямой можно задать посредством конструктора

```
Nleft = kmVector[ $\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \cdot v@"coord"$ ]
```

```
kmVector[km, , {-Vy, Vx}]
```

Задание 10. Представьте в *Mathematica* направляющий вектор V прямой l , если задана лево ориентированная по отношению к вектору V нормаль N_L к прямой l .

Выполнение задания 10

Если нормальный вектор прямой имеет координаты $N (N_x, N_y)$, и он является лево ориентированным к направляющему вектору V , то координаты вектора V могут быть записанными в виде $V (N_y, -N_x)$.

Рассмотрим направляющий вектор прямой l , лево ориентированный к искомому направляющему вектору V прямой l .

```
Nleft = kmVector[{Nx, Ny}]  
kmVector[km, , {Nx, Ny}]
```

Тогда координаты искомого направляющего вектора V прямой l могут быть получены посредством матрицы поворота

```

$$\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \cdot \text{Nleft@\"coord\"}$$
  
{Ny, -Nx}
```

Таким образом, выражение для представления экземпляра объекта «вектор на плоскости», являющегося направляющим вектором прямой, может быть записано с использованием нормального вектора этой прямой

```
V = kmVector[ $\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \cdot \text{Nleft@\"coord\"}$ ]  
kmVector[km, , {Ny, -Nx}]
```

2.6 Прямая, заданная точкой и нормальным вектором

Задание 11. Создайте возможность задания прямой l на плоскости указанием координат точки, ей принадлежащей, и координат нормального вектора к этой прямой. Прямая может быть именована либо нет.

Выполнение задания 11

Задание 11 выполните **самостоятельно**.

3 Запросы свойств прямой

Задание 12. Напишите оператор, возвращающий имя объекта «прямая на плоскости», если оно существует, и нарицательное имя "Прямая" в противном случае.

Выполнение задания 12

Задание 12 выполните самостоятельно.

Задание 13. Напишите оператор, возвращающий список коэффициентов общего уравнения прямой.

Выполнение задания 13

Задание 13 выполните самостоятельно.

Задание 14. Напишите оператор, возвращающий общее уравнение прямой.

Выполнение задания 14

Задание 14 выполните самостоятельно.

Задание 15. Напишите оператор, возвращающий нормальный вектор прямой.

Выполнение задания 15

Задание 15 выполните самостоятельно.

Задание 16. Напишите оператор, возвращающий направляющий вектор прямой.

Выполнение задания 16

Задание 16 выполните самостоятельно.

4 Отображение прямой на рисунке

Для отображения прямой на рисунке используем графический примитив `InfiniteLine`, в качестве аргументов которого требуется указать координаты двух различных точек прямой.

Задание 17. Постройте функцию, которая для заданной прямой на плоскости, вычисляет координаты двух различных точек, принадлежащих этой прямой.

Выполнение задания 17

Рассмотрим случай, когда оба коэффициента при неизвестных в общем уравнении прямой отличны от нуля. Например, пусть прямая **L** представлена в виде (1)

```
L = kmLine[km, "", {17, -8, -10}]
kmLine[km, , {17, -8, -10}]
```

Вычислим координаты некоторой точки, принадлежащей заданной прямой. Для этого в общем уравнении прямой зафиксируем значение одной из координат, решим полученное уравнение относительно оставшейся переменной и запишем полученную пару чисел – искомые координаты точки.

Например, для прямой **L** укажем конкретное значение $x = x_0$

```
L@"coef".{x0, y, 1} == 0
-10 - 8 y + 17 x0 == 0
```

и решим полученное уравнение относительно оставшейся переменной

```
First@Solve[L@"coef".{x0, y, 1} == 0, y]
{y -> 1/8 (-10 + 17 x0)}
```

Осталось сформировать из полученной пары искомый список координат точки, принадлежащей прямой

```
{x0, y} /. First@Solve[L@"coef".{x0, y, 1} == 0, y]
{x0, 1/8 (-10 + 17 x0)}
```

(17)

Выражение (17) вычисляет координаты точки, принадлежащей прямой, при условии, что коэффициент при переменной y отличен от нуля.

Чтобы получить две различные точки прямой, зафиксируем два различных значения координаты x , в частности, $x_0 = -1$ и $x_0 = 1$

```
Table[
  {x0, y} /.
    First@Solve[L@"coef".{x0, y, 1} == 0, y],
  {x0, {-1, 1}}]
{{-1, -27/8}, {1, 7/8}}
```

В силу специфики выполнения функции **Table**

```
Attributes[Table]
{HoldAll, Protected}
```

последнее выражение можно записать, не указывая имя переменной, относительно которой **Solve** решает уравнение. Это возможно, т.к. уравнение содержит только одну переменную, после того как переменной **x** последовательно задаются значения **x0=-1** и **x0=1**.

```
Table[
  {x0, y} /. First@Solve[L@"coef".{x0, y, 1} == 0],
  {x0, {-1, 1}}]
```

Убедимся в том, что выражение вычисляет координаты двух различных точек прямой и в случае, когда коэффициент **A**, который стоит при переменной **x**, равен нулю. Например,

```
Ly = kmLine[km, "", {0, -8, -10}]
kmLine[km, , {0, -8, -10}]
Table[
  {x0, y} /. First@Solve[Ly@"coef".{x0, y, 1} == 0],
  {x0, {-1, 1}}]
{{-1, -5/4}, {1, -5/4}}
```

Однако, в случае, когда в общем уравнении прямой равен нулю коэффициент при переменной **y**, этот описанный выше способ вычисления

координат двух различных точек прямой не работает: при фиксировании оставшейся единственной переменной x правая часть уравнения прямой становится числом, и само уравнение вырождается в логическое высказывание.

```
Lx = kmLine[km, "", {17, 0, -10}]
kmLine[km, , {17, 0, -10}]
```

```
Lx@"coef".{1, y, 1} == 0
False
```

Тогда в случае, когда коэффициент при переменной y равен нулю, мы можем использовать такой же прием, но фиксировать координату $y = y_0$.

Подводя итог рассуждениям, приведенным выше, формулируем следующий **алгоритм вычисления координат двух различных точек прямой**:

- выбираем наименьший по абсолютной величине коэффициент в паре **A**, **B**, коэффициентов при неизвестных общего уравнения прямой;
- координате, соответствующей выбранному коэффициенту, придаём по очереди любые два различных значения, например, +1 и -1;
- вычисляем значения второй координаты точки, удовлетворяющей заданному уравнению прямой;
- формируем список, содержащий координаты искомых точек, где каждая координата представлена списком.

Для реализации алгоритма используем структуру выбора **If** и структуру повторения **Table**

```
If[Apply[Less, Abs@L["coef"]][{1, 2}],
  Table[
    {x0, y} /. First@Solve[L["coef"].{x0, y, 1} == 0,
      {x0, {-1, 1}}],
  Table[
    {x, y0} /. First@Solve[L["coef"].{x, y0, 1} == 0,
      {y0, {-1, 1}}]]]
{{2/17, -1}, {18/17, 1}}
```


Теперь построим функцию **kmGetTwoLinePoints**, которая для заданной прямой возвращает две различные точки (два экземпляра объекта «точка на плоскости»), принадлежащие этой прямой.

```
ClearAll[kmGetTwoLinePoints];
kmGetTwoLinePoints[line_kmLine] := Map[kmPoint,
  If[Apply[Less, Abs@line["coef"]][{1, 2}]],
  Table[
    {x, y} /. First@Solve[line["coef"].{x, y, 1} == 0],
    {x, {-1, 1}}],
  Table[
    {x, y} /. First@Solve[line["coef"].{x, y, 1} == 0],
    {y, {-1, 1}}]]]
```

Убедимся, корректно ли построена функция

```
kmGetTwoLinePoints/@{L, Lx, Ly}
{{kmPoint[km, , {2/17, -1}], kmPoint[km, , {18/17, 1}]},
 {kmPoint[km, , {10/17, -1}], kmPoint[km, , {10/17, 1}]},
 {kmPoint[km, , {-1, -5/4}], kmPoint[km, , {1, -5/4}]}}
```

Задание 18. Обучите *Mathematica* отображать на рисунке объект «точка на плоскости» и объект «прямая на плоскости». Используйте правила локальных преобразований.

Выполнение задания 18

Создадим препроцессор, который объекты «точка на плоскости» и «прямая на плоскости» преобразовывает в выражения, необходимые *Mathematica* для построения графических образов этих объектов. А именно, подготовим входные аргументы для графических примитивов **Point** и **InfiniteLine**.

Кроме того, посредством функции **Tooltip** обеспечим всплывающую подсказку, которая будет появляться всякий раз, когда на изображенный графический объект наведем указатель мыши. Текст, появляющийся во всплывающей подсказке при наведении на объект, совпадает с внутренним представлением объекта.

```
ClearAll[kmGraphics2D]; (19)
kmGraphics2D[gp_List, opts___] := Graphics[
  ReplaceAll[gp, {
    P_kmPoint  $\Rightarrow$  Tooltip[Point@P@"coord",
      Style[Format[P, StandardForm], Large]],
    L_kmLine  $\Rightarrow$  Tooltip[
      InfiniteLine@
        Through[kmGetTwoLinePoints[L]@"coord",
          Style[Format[L, StandardForm], Large]]
    ]],
  opts]
```

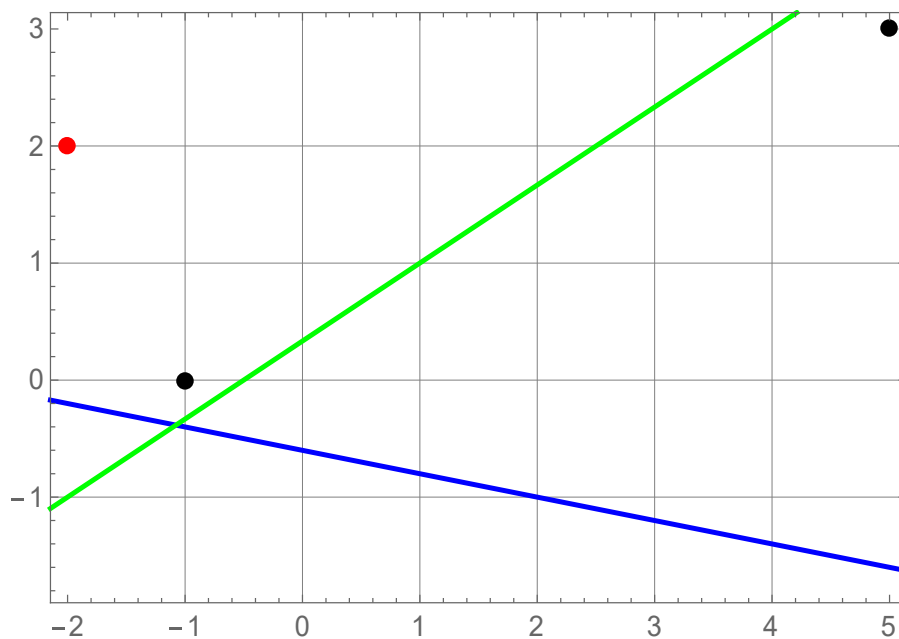
Зададим несколько точек и прямых,

```
{P1, P2, P3} = kmPoint /@ {{-1, 0}, {5, 3}, {-2, 2}}
{L1, L2} = kmLine[km, "", #] & /@ {{1, 5, 3}, {-2, 3, -1}}

{kmPoint[km, , {-1, 0}],
 kmPoint[km, , {5, 3}], kmPoint[km, , {-2, 2}]}
{kmLine[km, , {1, 5, 3}], kmLine[km, , {-2, 3, -1}]}
```

Отообразим представленные объекты на рисунке. Для точек используем директивы размера и цвета, для прямых укажем толщину и цвет линий.

```
kmGraphics2D[{
  {PointSize@Large, P1, P2, Red, P3},
  {Thick, Blue, L1, Green, L2}}, Frame  $\rightarrow$  True,
  GridLines  $\rightarrow$  Automatic, PlotRange  $\rightarrow$  All]
```



Литература

1. Голубева Л.Л., Малевич А.Э., Щеглова Н.Л. Компьютерная математика. Символьный пакет *Mathematica*. Лаб. практикум в 2 ч. Ч 1.- Минск: БГУ, 2012. – 235 с.