Практический линал. Лабораторная работа № 2

Троицкая Тамара 368924

21 октября 2023 г.

Todo:

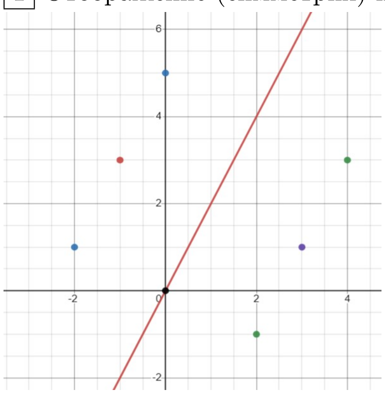
1. Содержание с навигацией

2. Оставить второе отдельно, а третье написать вместе с первым

В прошлый раз я написала код на С, но не заморочилась с оформлением. На этот раз я написала код на C++ для умножения матриц и, как видите, оформила отчёт в латехе. Основные функции из этой работы я вставлю в конце отчёта, чтобы не смущать людей. Также, признаюсь, написала на python код, чтобы оборачивать в LaTeXматрицы и векторы. При желании все исходники можно найти здесь: https://github.com/cgsg-tt6ITMO/s3_practlinal_lab2

Задание 1. Придумайте. Придумайте матрицы 2 × 2, которые задают:

$\boxed{1}$ Отображение (симметрия) плоскости относительно y = ax (a = 2)



Примеры преобразования точек:

$\begin{pmatrix} 0 & 5 \end{pmatrix} \to \begin{pmatrix} 4 & 3 \end{pmatrix}$

$\begin{pmatrix} 3 & 1 \end{pmatrix} \to \begin{pmatrix} -1 & 3 \end{pmatrix}$

$A \cdot B = C$

где матрица A - искомая матрица преобразования, столбцы матрицы B - это исходные векторы, а столбцы матрицы C - это полученые в результате преобразования векторы.

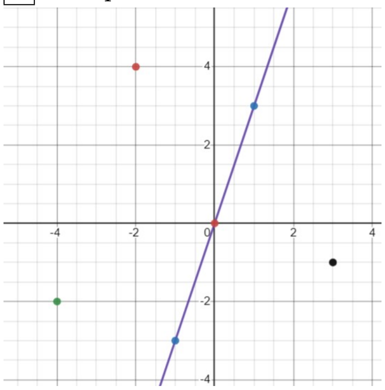
$A = C \cdot B^{-1}$

$$\begin{pmatrix} 4 & -1 \\ 3 & 3 \end{pmatrix} \cdot \begin{pmatrix} 0 & 3 \\ 5 & 1 \end{pmatrix}^{-1} = \boxed{\begin{pmatrix} -0.6 & 0.8 \\ 0.8 & 0.6 \end{pmatrix}}$$

Проверка:

$$\begin{pmatrix} -0.6 & 0.8 \\ 0.8 & 0.6 \end{pmatrix} \cdot \begin{pmatrix} -2 \\ 1 \end{pmatrix} = \begin{pmatrix} (-0.6) \cdot (-2) + 0.8 \cdot 1 \\ 0.8 \cdot (-2) + 0.6 \cdot 1 \end{pmatrix} = \begin{pmatrix} 2 \\ -1 \end{pmatrix}$$

$\boxed{2}$ Отображение всей плоскости в прямую y = bx (b = 3)



Примеры преобразования точек:

$\begin{pmatrix}-2 & 4\end{pmatrix} \to \begin{pmatrix}1 & 3\end{pmatrix}$

$\begin{pmatrix}-4 & -2\end{pmatrix} \to \begin{pmatrix}-1 & -3\end{pmatrix}$

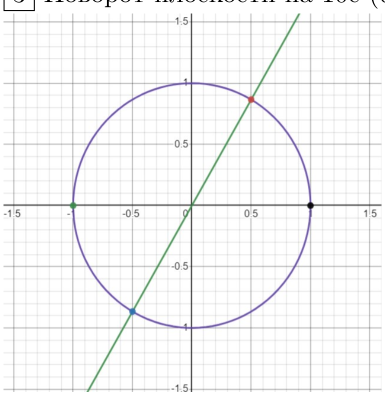
$C \cdot B^{-1} = A$

$$\begin{pmatrix}1 & -1\\ 3 & -3\end{pmatrix} \cdot \begin{pmatrix}-2 & -4\\ 4 & -2\end{pmatrix}^{-1} = \boxed{\begin{pmatrix}0.1 & 0.3\\ 0.3 & 0.9\end{pmatrix}}$$

Проверка:

$$\begin{pmatrix}0.1 & 0.3\\ 0.3 & 0.9\end{pmatrix} \cdot \begin{pmatrix}3\\ -1\end{pmatrix} = \begin{pmatrix}0.1 \cdot 3 + 0.3 \cdot (-1)\\ 0.3 \cdot 3 + 0.9 \cdot (-1)\end{pmatrix} = \begin{pmatrix}0\\ 0\end{pmatrix}$$

$\boxed{3}$ Поворот плоскости на 10c (60) градусов против часовой стрелки.



Примеры преобразования точек:

$\begin{pmatrix}1 & 0\end{pmatrix} \to \begin{pmatrix}1/2 & \sqrt{3}/2\end{pmatrix}$

$\begin{pmatrix}-1 & 0\end{pmatrix} \to \begin{pmatrix}-1/2 & -\sqrt{3}/2\end{pmatrix}$

$C \cdot B^{-1} = A$

$$\begin{pmatrix}1/2 & -1/2\\ \sqrt{3}/2 & -\sqrt{3}/2\end{pmatrix} \cdot \begin{pmatrix}1 & -1\\ 0 & 0\end{pmatrix}^{-1} = \boxed{\begin{pmatrix}1/2 & -\sqrt{3}/2\\ \sqrt{3}/2 & 1/2\end{pmatrix}}$$

Вспомним, что матрица поворота на α:

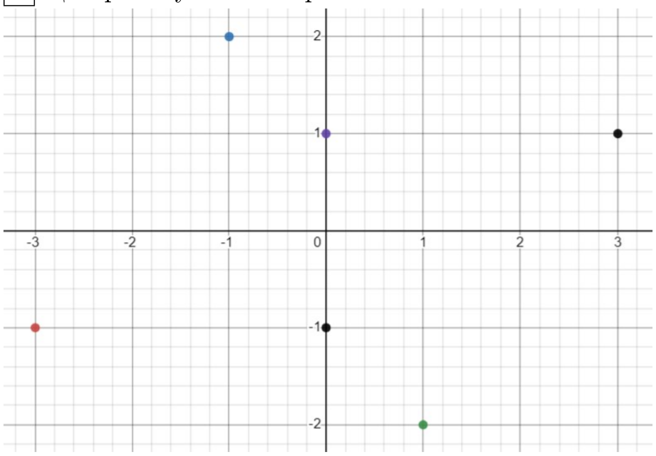
$$\begin{pmatrix}\cos(\alpha) & -\sin(\alpha)\\ \sin(\alpha) & \cos(\alpha)\end{pmatrix}$$

Полученный нами результат согласуется с теорией.

Проверка:

$$\begin{pmatrix}\frac{1}{2} & \frac{-\sqrt{3}}{2}\\ \frac{\sqrt{3}}{2} & \frac{1}{2}\end{pmatrix} \cdot \begin{pmatrix}\frac{\sqrt{3}}{2}\\ \frac{1}{2}\end{pmatrix} = \begin{pmatrix}0\\ 1\end{pmatrix}$$

$\boxed{4}$ Центральную симметрию плоскости относительно начала координат.



Примеры преобразования точек:

$\begin{pmatrix}3 & 1\end{pmatrix} \to \begin{pmatrix}-3 & -1\end{pmatrix}$

$\begin{pmatrix}-1 & 2\end{pmatrix} \to \begin{pmatrix}1 & -2\end{pmatrix}$

$C \cdot B^{-1} = A$

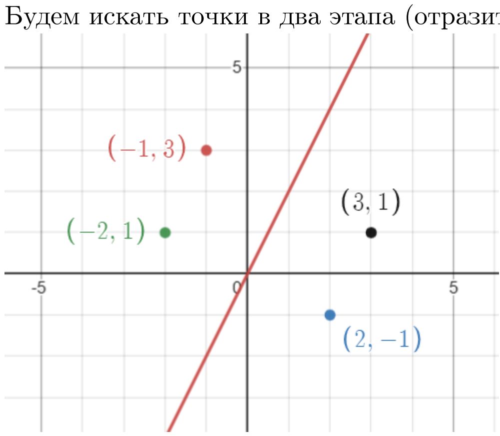
$$\begin{pmatrix}-3 & 1\\ -1 & -2\end{pmatrix} \cdot \begin{pmatrix}3 & -1\\ 1 & 2\end{pmatrix}^{-1} = \boxed{\begin{pmatrix}-1 & 0\\ 0 & -1\end{pmatrix}}$$

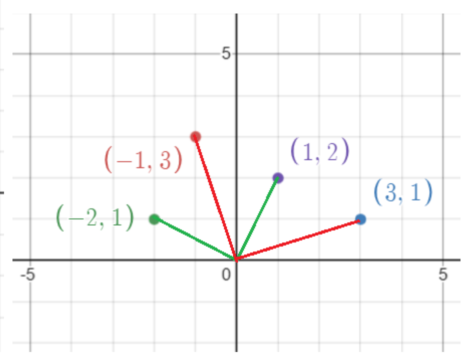
Проверка:

$$\begin{pmatrix}-1 & 0\\ 0 & -1\end{pmatrix} \cdot \begin{pmatrix}0\\ -1\end{pmatrix} = \begin{pmatrix}0\\ 1\end{pmatrix}$$

$\boxed{5}$ Отображение, которое можно описать так: сначала отражение относительно прямой y = ax (y = 2x), потом поворот на 10d (90) градусов по часовой стрелке.

Будем искать точки в два этапа (отразить, повернуть):





Запишем преобразование векторов в два этапа:

$\begin{pmatrix}2 & -1\end{pmatrix} \to \begin{pmatrix}-2 & 1\end{pmatrix} \to \begin{pmatrix}1 & 2\end{pmatrix}$

$\begin{pmatrix}3 & 1\end{pmatrix} \to \begin{pmatrix}-1 & 3\end{pmatrix} \to \begin{pmatrix}3 & 1\end{pmatrix}$

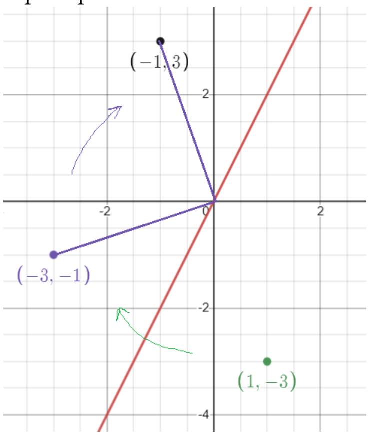
$$C \cdot B^{-1} = A = \begin{pmatrix}1 & 3\\ 2 & 1\end{pmatrix} \cdot \begin{pmatrix}2 & 3\\ -1 & 1\end{pmatrix}^{-1} = \boxed{\begin{pmatrix}0.8 & 0.6\\ 0.6 & -0.8\end{pmatrix}}$$

Заметим, что

$$\begin{pmatrix}0 & 1\\ -1 & 0\end{pmatrix} \cdot \begin{pmatrix}-0.6 & 0.8\\ 0.8 & 0.6\end{pmatrix} = \begin{pmatrix}0.8 & 0.6\\ 0.6 & -0.8\end{pmatrix}$$

То есть эту матрицу можно получить последовательным умножением матрицы поворота (см. п. 3) и матрицы отражения (см. п. 1). Порядок именно такой, потому что при умножении этой конструкции на вектор сначала на него умножится матрица отражения, находящаяся правее, а потом уже матрица поворота. Можно сказать, что действие произойдёт справа налево.

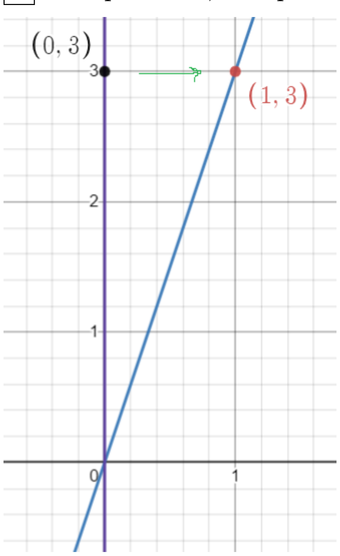
Проверка:

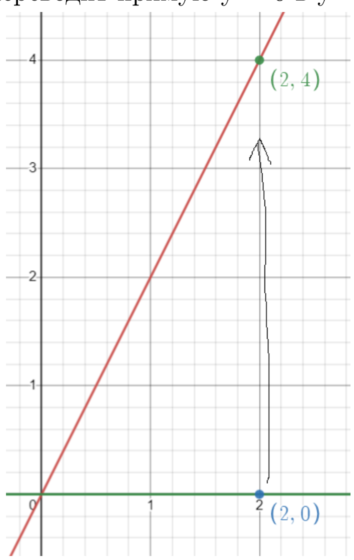


$$\begin{pmatrix}1 & -3\end{pmatrix} \to \begin{pmatrix}-3 & -1\end{pmatrix} \to \begin{pmatrix}-1 & 3\end{pmatrix}$$

$$\begin{pmatrix}0.8 & 0.6\\ 0.6 & -0.8\end{pmatrix} \cdot \begin{pmatrix}1\\ -3\end{pmatrix} = \begin{pmatrix}-1\\ 3\end{pmatrix}$$

$\boxed{6}$ Отображение, которое переводит прямую y = 0 в y = ax (y = 2x) и прямую x = 0 в y = bx (y = 3x).





Примеры преобразования точек:

$$\begin{pmatrix}0 & 3\end{pmatrix} \to \begin{pmatrix}1 & 3\end{pmatrix}$$

$$\begin{pmatrix}2 & 0\end{pmatrix} \to \begin{pmatrix}2 & 4\end{pmatrix}$$

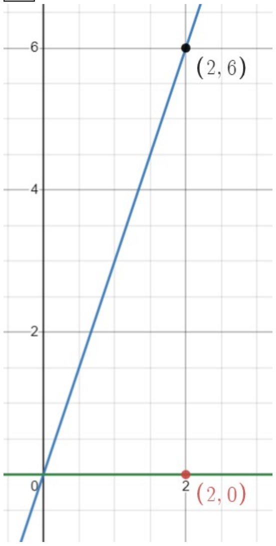
$C \cdot B^{-1} = A$

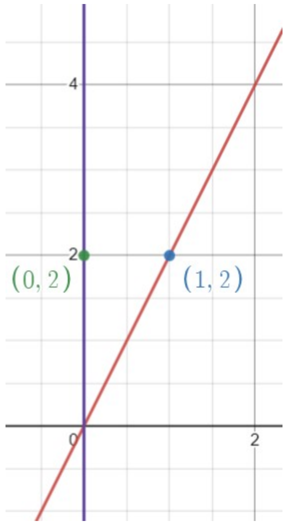
$$\begin{pmatrix}1 & 2\\ 3 & 4\end{pmatrix} \cdot \begin{pmatrix}0 & 2\\ 3 & 0\end{pmatrix}^{-1} = \boxed{\begin{pmatrix}1 & 1/3\\ 2 & 1\end{pmatrix}}$$

Проверка:

$$\begin{pmatrix}1 & 1/3\\ 2 & 1\end{pmatrix} \cdot \begin{pmatrix}0\\ 6\end{pmatrix} = \begin{pmatrix}2\\ 6\end{pmatrix}$$

$\boxed{7}$ Отображение, которое переводит прямую y = ax (y = 2x) в y = 0 и прямую y = bx (y = 3x) в x = 0.





Примеры преобразования точек:

$\begin{pmatrix} 2 & 6 \end{pmatrix} \rightarrow \begin{pmatrix} 2 & 0 \end{pmatrix}$

$\begin{pmatrix} 1 & 2 \end{pmatrix} \rightarrow \begin{pmatrix} 0 & 2 \end{pmatrix}$

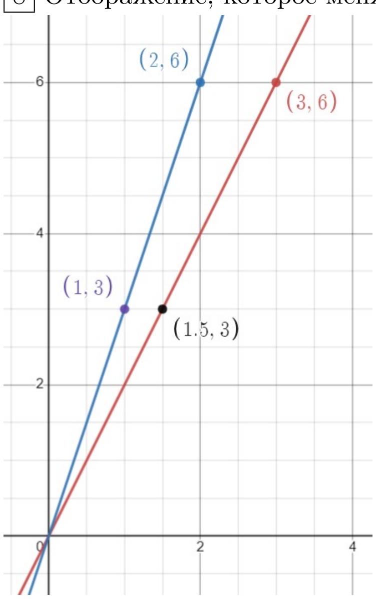
$C \cdot B^{-1} = A$

$$\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \cdot \begin{pmatrix} 2 & 1 \\ 6 & 2 \end{pmatrix}^{-1} = \boxed{\begin{pmatrix} -2 & 1 \\ 6 & -2 \end{pmatrix}}$$

Проверка:

$$\begin{pmatrix} -2 & 1 \\ 6 & -2 \end{pmatrix} \cdot \begin{pmatrix} -1 \\ -2 \end{pmatrix} = \begin{pmatrix} 0 \\ -2 \end{pmatrix}$$

$\boxed{8}$ Отображение, которое меняет местами прямые y = ax и y = bx (y = 2x и y = 3x).



Примеры преобразования точек:

$\begin{pmatrix} 1 & 3 \end{pmatrix} \rightarrow \begin{pmatrix} 1.5 & 3 \end{pmatrix}$

$\begin{pmatrix} 3 & 6 \end{pmatrix} \rightarrow \begin{pmatrix} 2 & 6 \end{pmatrix}$

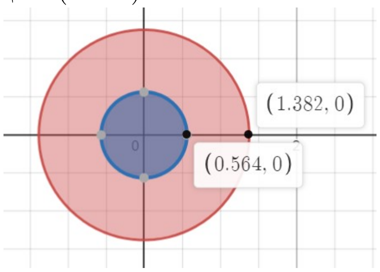
$C \cdot B^{-1} = A$

$$\begin{pmatrix} 1 & 2 \\ 3 & 6 \end{pmatrix} \cdot \begin{pmatrix} 1 & 3 \\ 3 & 6 \end{pmatrix}^{-1} = \boxed{\begin{pmatrix} -1 & 5/6 \\ 0 & 1 \end{pmatrix}}$$

Проверка:

$$\begin{pmatrix} -1 & 5/6 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} -1 \\ -3 \end{pmatrix} = \begin{pmatrix} -1.5 \\ -3 \end{pmatrix}$$

$\boxed{9}$ Отображение, которое переводит круг единичной площади с центром в начале координат в круг площади c (c = 6).



$\pi r^2 = 1$

$r = 1/\sqrt{\pi}$

$\pi R^2 = c$

$R = \sqrt{c/\pi} = \sqrt{6/\pi}$

Примеры преобразования точек:

$\begin{pmatrix} 0 & r \end{pmatrix} \to \begin{pmatrix} 0 & R \end{pmatrix}$

$\begin{pmatrix} r & 0 \end{pmatrix} \to \begin{pmatrix} R & 0 \end{pmatrix}$

B $= \begin{pmatrix} 0 & 1/\sqrt{\pi} \\ 1/\sqrt{\pi} & 0 \end{pmatrix}$

C $= \begin{pmatrix} 0 & \sqrt{6/\pi} \\ \sqrt{6/\pi} & 0 \end{pmatrix}$

A $= \boxed{\begin{pmatrix} \sqrt{6} & 0 \\ 0 & \sqrt{6} \end{pmatrix}}$

$\boxed{10}$ Отображение, которое переводит круг единичной площади с центром в начале координат в некруг площади d (d = 9).

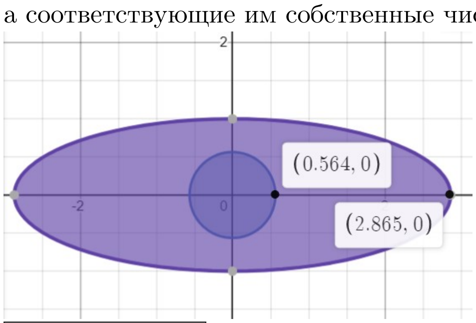
$\pi r^2 = 1$

$r = 1/\sqrt{\pi}$

Допустим, некруг это эллипс. Площадь эллипса вычисляется как $S = ab\pi$. Например, $a = 9/\pi, b = 1$. Тогда по оси оy фигура растягивается в $1/\sqrt{\pi} : 1$ раз, по оси ох в $\frac{1}{\sqrt{\pi}} : \frac{9}{\pi}$ раз.

То есть собственные векторы: v_1 =(0, 1) и $v_2 =$ (1, 0),

а соответствующие им собственные числа: $\lambda_1 = 1/\sqrt{\pi}, \lambda_2 = 9\sqrt{\pi}$



$\boxed{\begin{pmatrix} 1/\sqrt{\pi} & 0 \\ 0 & 9\sqrt{\pi} \end{pmatrix}}$

$\boxed{11}$ Отображение, у которого собственные вектора перпендикулярны, и ни один из них не лежит на прямой y = 0 или y = x.

Пусть $v_1 = \begin{pmatrix} 2 & 1 \end{pmatrix}$ $\lambda_1 = -1$,

$v_2 = \begin{pmatrix} -1 & 2 \end{pmatrix}$ $\lambda_2 = 3/2$

A $\begin{pmatrix} 2 \\ 1 \end{pmatrix}$ = -1 $\begin{pmatrix} 2 \\ 1 \end{pmatrix}$

A $\begin{pmatrix} -1 \\ 2 \end{pmatrix}$ = 3/2 $\begin{pmatrix} -1 \\ 2 \end{pmatrix}$

A $\begin{pmatrix} 2 & -1 \\ 1 & 2 \end{pmatrix} = \begin{pmatrix} -2 & -1.5 \\ -1 & 3 \end{pmatrix}$

$A = C \cdot B^{-1} = \begin{pmatrix} -2 & -1.5 \\ -1 & 3 \end{pmatrix} \begin{pmatrix} 2 & -1 \\ 1 & 2 \end{pmatrix}^{-1} = \boxed{\begin{pmatrix} -0.5 & -1 \\ -1 & 1 \end{pmatrix}}$

$\boxed{12}$ Отображение, у которого нет двух неколлинеарных собственных векторов.

$\boxed{\begin{pmatrix} 2 & 1 \\ 0 & 2 \end{pmatrix}}$

Это Жорданова клетка для $\lambda = 2$ с геометрической кратностью 1. Все собственные векторы пропорциональны $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$, то есть коллинеарны. (Если говорить безграмотно, у неё только один собственный вектор)

$\boxed{13}$ Отображение, у которого нет ни одного вещественного собственного вектора (но при этом само отображение задаётся вещественной матрицей).

$\boxed{\begin{pmatrix} -2 & -1 \\ 9 & -2 \end{pmatrix}}$

$det \begin{pmatrix} -2-\lambda & -1 \\ 9 & -2-\lambda \end{pmatrix} = (\lambda+2)^2 + 9 = 0$

$\lambda_1 = -2 - 3i$ $v_1 = \begin{pmatrix} -i/3 \\ 1 \end{pmatrix}$

$\lambda_2 = -2 + 3i$ $v_2 = \begin{pmatrix} i/3 \\ 1 \end{pmatrix}$

$\boxed{14}$ Отображение, для которого любой ненулевой вектор является собственным.

$\boxed{\begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix}}$

$\lambda = 4$

$\begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 4x \\ 4y \end{pmatrix}$

$\forall x, y$

$\boxed{15}$ Пару отображений, последовательное применение которых даёт различные результаты в зависимости от порядка: AB = BA.

Отразим относительно прямой y = x и повернём на 90° против часовой стрелки.

Матрица поворота на 90° против часовой стрелки:

$A = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$

Матрица отражения относительно прямой y = x:

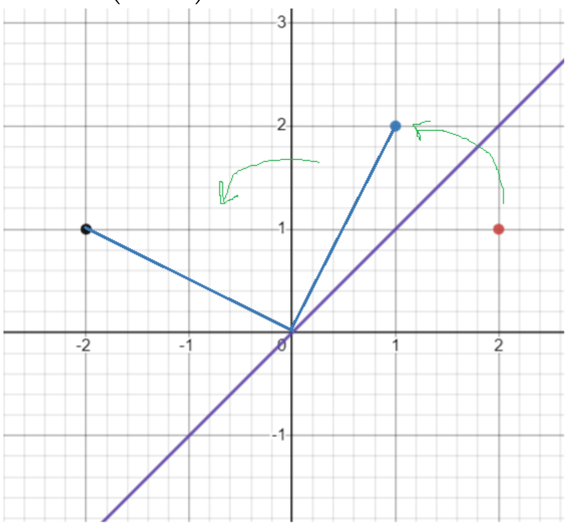
$B = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

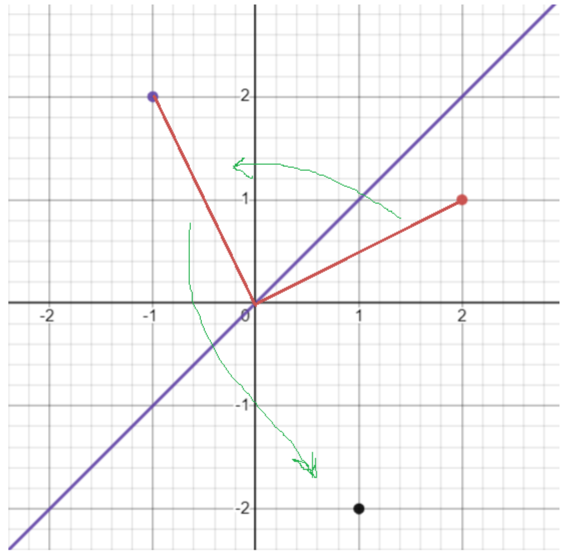
Отразить и потом повернуть:

$A \cdot B = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$

Повернуть и потом отразить:

$B \cdot A = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$





Слева отразили и повернули, справа повернули и отразили. Результаты не совпали.

$\boxed{16}$ Пару отображений, последовательное применение которых даёт одинаковый результат независимо от порядка: AB = BA. Постарайтесь, чтобы матрицы A и B были максимально непохожими друг на друга.

Сделать два поворота. Сначала на 60°, потом на 30°. И в обратной последовательности.

Матрица поворота на 60° против часовой стрелки:

$A = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix} = \begin{pmatrix} 1/2 & -\sqrt{3}/2 \\ \sqrt{3}/2 & 1/2 \end{pmatrix}$

Матрица поворота на 30° против часовой стрелки:

$B = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix} = \begin{pmatrix} \sqrt{3}/2 & -1/2 \\ 1/2 & \sqrt{3}/2 \end{pmatrix}$

$A \cdot B = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$

$B \cdot A = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$

Нетрудно заметить, что в обоих случаях получилась матрица поворота на 90° против часовой оси. Это согласуется с теорией (и здравым смыслом) – порядок выполнения двух поворотов неважен, повороты "складываются".

Задание 2. Проанализируйте.

• Найдите образ и ядро придуманных вами отображений из пунктов 1, 2, 13, 14.

1) $\begin{pmatrix} -0.6 & 0.8 \\ 0.8 & 0.6 \end{pmatrix} \sim \begin{pmatrix} 0 & 1.25 \\ 0.8 & 0.6 \end{pmatrix}$

$\begin{pmatrix} -0.6 & 0.8 \\ 0.8 & 0.6 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} -0.6 \\ 0.8 \end{pmatrix}, \begin{pmatrix} -0.6 & 0.8 \\ 0.8 & 0.6 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0.8 \\ 0.6 \end{pmatrix}$

Получаем образ:

$im(A) = Span[\begin{pmatrix} -0.6 \\ 0.8 \end{pmatrix}, \begin{pmatrix} 0.8 \\ 0.6 \end{pmatrix}] = Span[\begin{pmatrix} -3 \\ 4 \end{pmatrix}, \begin{pmatrix} 4 \\ 3 \end{pmatrix}]$

$rank(A) = 2$

$nullity(A) = 0$

$kerf(A) = \{0\}$

2) $\begin{pmatrix} 0.1 & 0.3 \\ 0.3 & 0.9 \end{pmatrix} \sim \begin{pmatrix} 0.1 & 0.3 \\ 0 & 0 \end{pmatrix}$

$rank(A) = 1$

$nullity(A) = 1$

$im(A) = Span[\begin{pmatrix} 1 \\ 3 \end{pmatrix}]$

Ядро: $0.1x + 0.3y = 0$

$kerf(A) = Span[\begin{pmatrix} -3 \\ 1 \end{pmatrix}]$

13) $\begin{pmatrix} -2 & -1 \\ 9 & -2 \end{pmatrix} \sim \begin{pmatrix} -2 & -1 \\ 0 & -6.5 \end{pmatrix}$

rank(A) = 2

$im(A) = Span[\begin{pmatrix} -2 \\ 9 \end{pmatrix}, \begin{pmatrix} -1 \\ -2 \end{pmatrix}]$

nullity(A) = 0

kerf(A) = {0}

14) $\begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix}$

rank(A) = 2

nullity(A) = 0

Образ – любой вектор длины 2.

Im(A) = Span[$\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}$]

kerf(A) = {0}

• Найдите собственные числа и собственные вектора придуманных вами отображений из пунктов 1, 2, 3, 4, 8, 11, 12, 13, 14, 15, 16.

1) $\begin{pmatrix} -0.6 & 0.8 \\ 0.8 & 0.6 \end{pmatrix}$

$det \begin{pmatrix} -0.6 - \lambda & 0.8 \\ 0.8 & 0.6 - \lambda \end{pmatrix} = (\lambda + 0.6) \cdot (\lambda - 0.6) - 0.64 = \lambda^2 - 0.36 - 0.64 = \lambda^2 - 1 = 0$

$\lambda_1 = 1$, $\lambda_2 = -1$

$v_1 = \begin{pmatrix} 0.5 \\ 1 \end{pmatrix}, v_2 = \begin{pmatrix} -2 \\ 1 \end{pmatrix}$

2) $\begin{pmatrix} 0.1 & 0.3 \\ 0.3 & 0.9 \end{pmatrix}$

$det\begin{pmatrix} 0.1-\lambda & 0.3 \\ 0.3 & 0.9-\lambda \end{pmatrix} = (0.1-\lambda)\cdot(0.9-\lambda) - 0.9 = \lambda^2 - \lambda + 0.9 - 0.9 = 0$

$\lambda_1 = 0,\ \lambda_2 = 1$

$v_1 = \begin{pmatrix} -3 \\ 1 \end{pmatrix},\ v_2 = \begin{pmatrix} 1/3 \\ 1 \end{pmatrix}$

3) $\begin{pmatrix} 1/2 & -\sqrt{3}/2 \\ \sqrt{3}/2 & 1/2 \end{pmatrix}$

$\lambda_1 = \frac{-\sqrt{3}i+1}{2},\ \lambda_2 = \frac{\sqrt{3}i+1}{2}$

$v_1 = \begin{pmatrix} -i \\ 1 \end{pmatrix},\ v_2 = \begin{pmatrix} i \\ 1 \end{pmatrix}$

4) $\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$

$\lambda_1 = -1$

$v_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix},\ v_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

8) $\begin{pmatrix} -1 & 5/6 \\ 0 & 1 \end{pmatrix}$

$\lambda_1 = 1, \lambda_2 = -1$

$v_1 = \begin{pmatrix} 5/12 \\ 1 \end{pmatrix},\ v_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

11) $\begin{pmatrix} -0.5 & -1 \\ -1 & 1 \end{pmatrix}$

$\lambda_1 = -1, \lambda_2 = 1.5$

$v_1 = \begin{pmatrix} 2 \\ 1 \end{pmatrix},\ v_2 = \begin{pmatrix} -0.5 \\ 1 \end{pmatrix}$

12) $\begin{pmatrix} 2 & 1 \\ 0 & 2 \end{pmatrix}$

$\lambda_1 = 2$

$v_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$

13) $\begin{pmatrix} -2 & -1 \\ 9 & -2 \end{pmatrix}$

$\lambda_1 = -2-3i, \lambda_2 = -2+3i$

$v_1 = \begin{pmatrix} -i/3 \\ 1 \end{pmatrix},\ v_2 = \begin{pmatrix} i/3 \\ 1 \end{pmatrix}$

14) $\begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix}$

$\lambda_1 = 4$

$v_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix},\ v_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

15)

$A = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$

$\lambda_1 = i, \lambda_2 = -i$

$v_1 = \begin{pmatrix} i \\ 1 \end{pmatrix},\ v_2 = \begin{pmatrix} -i \\ 1 \end{pmatrix}$

$B = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

$\lambda_1 = 1, \lambda_2 = -1$

$v_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $v_2 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$

$A \cdot B = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$

$\lambda_1 = 1, \lambda_2 = -1$

$v_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, $v_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

$B \cdot A = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$

$\lambda_1 = 1, \lambda_2 = -1$

$v_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $v_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

16)

$A = \begin{pmatrix} 1/2 & -\sqrt{3}/2 \\ \sqrt{3}/2 & 1/2 \end{pmatrix}$

$\lambda_1 = \frac{-\sqrt{3}i+1}{2}$, $\lambda_2 = \frac{\sqrt{3}i+1}{2}$

$v_1 = \begin{pmatrix} -i \\ 1 \end{pmatrix}$, $v_2 = \begin{pmatrix} i \\ 1 \end{pmatrix}$

$B = \begin{pmatrix} \sqrt{3}/2 & -1/2 \\ 1/2 & \sqrt{3}/2 \end{pmatrix}$

$\lambda_1 = \frac{\sqrt{3}-i}{2}$, $\lambda_2 = \frac{\sqrt{3}+i}{2}$

$v_1 = \begin{pmatrix} -i \\ 1 \end{pmatrix}$, $v_2 = \begin{pmatrix} i \\ 1 \end{pmatrix}$

$A \cdot B = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$

$\lambda_1 = i$, $\lambda_2 = -i$

$v_1 = \begin{pmatrix} i \\ 1 \end{pmatrix}$, $v_2 = \begin{pmatrix} -i \\ 1 \end{pmatrix}$

$B \cdot A = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$

• Найдите определитель матриц из пунктов 1, 2, 3, 4, 5, 9, 10.

1) $det \begin{pmatrix} -0.6 & 0.8 \\ 0.8 & 0.6 \end{pmatrix} = -0.36 - 0.64 = -1$

2) $det \begin{pmatrix} 0.1 & 0.3 \\ 0.3 & 0.9 \end{pmatrix} = 0$

3) $det \begin{pmatrix} 1/2 & -\sqrt{3}/2 \\ \sqrt{3}/2 & 1/2 \end{pmatrix} = 1/4 + 3/4 = 1$

4) $det \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} = 1$

5) $det \begin{pmatrix} 0.8 & 0.6 \\ 0.6 & -0.8 \end{pmatrix} = -0.64 - 0.36 = -1$

9) $det \begin{pmatrix} \sqrt{6} & 0 \\ 0 & \sqrt{6} \end{pmatrix} = 6$

10) $det \begin{pmatrix} 1/\sqrt{\pi} & 0 \\ 0 & 9\sqrt{\pi} \end{pmatrix} = 9/\pi$

• В каких пунктах матрица обязательно получается симметричной?

Точно да: 1, 2, 4, 5, 9, 14.

(точно будет 2 вещественных собственных вектора и вещественные собственные числа)

Точно не: 3, 6, 7, 8, 12, 13, 15, 16. 10, 11.

Задание 3. Визуализируйте. Используя MATLAB или Python, выполните визуализацию полученных линейных преобразований. Для этого:

• Задайте произвольную фигуру как многоугольник с вершинами в выбранных вами точках. Постройте её графическое изображение. Это – оригинал.

• Найдите образ каждой вершины многоугольника при линейном отображении рассматриваемой матрицей. Постройте графическое изображение многоугольника на полученных (отображённых) вершинах. Это – результат преобразования, образ.

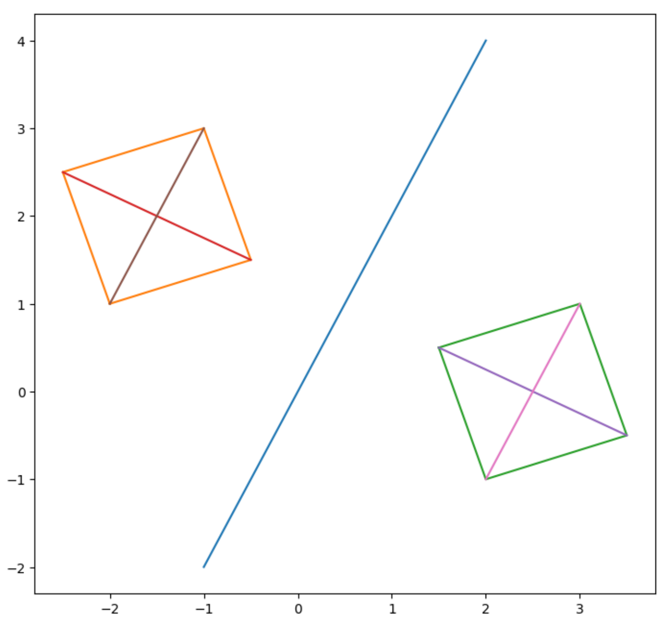
• Выполните указанную визуализацию для всех отображений из первого задания.

• При работе с пунктами 15 и 16 сделайте визуализацию всех рассматриваемых отображений, а именно: A, B, AB и BA.

• Для пунктов 1, 11, 12, 14, 15, 16 добавьте на картинку прямые, совпадающие с направлениями собственных векторов.

В отображениях, где поворот или отражение производится относительно центра, я добавила оси.

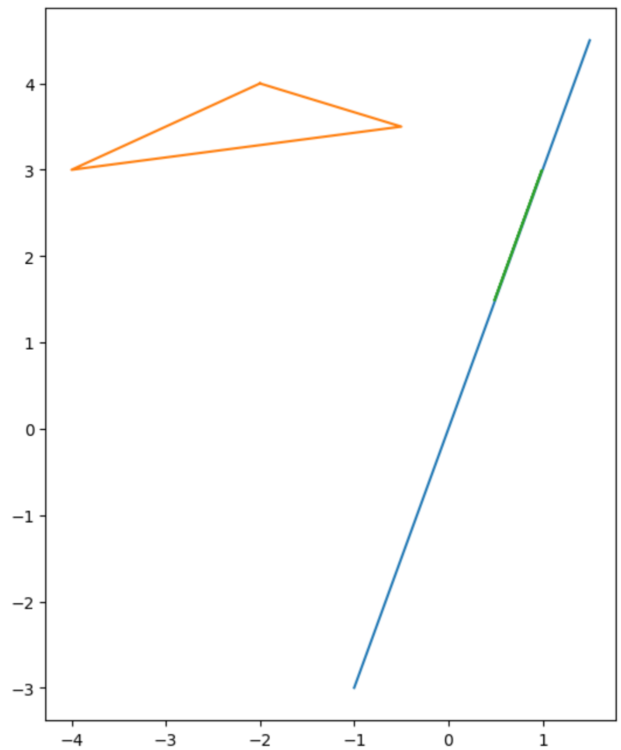
1) !



Зелёный – оригинал, рыжий – его образ. Но вообще они переходят друг в друга. Синия прямая – ось симметрии.

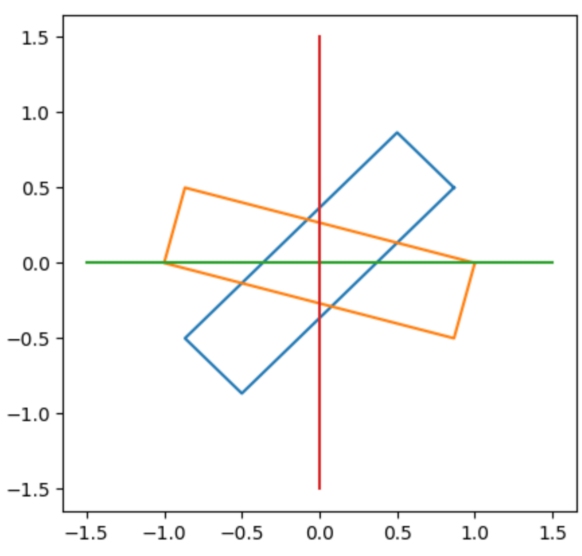
Фиолетовый и красный, розовый и коричневый соответственно коллинеарны собственным векторам матрицы этого отображения, поэтому они не изменили направления.

2)



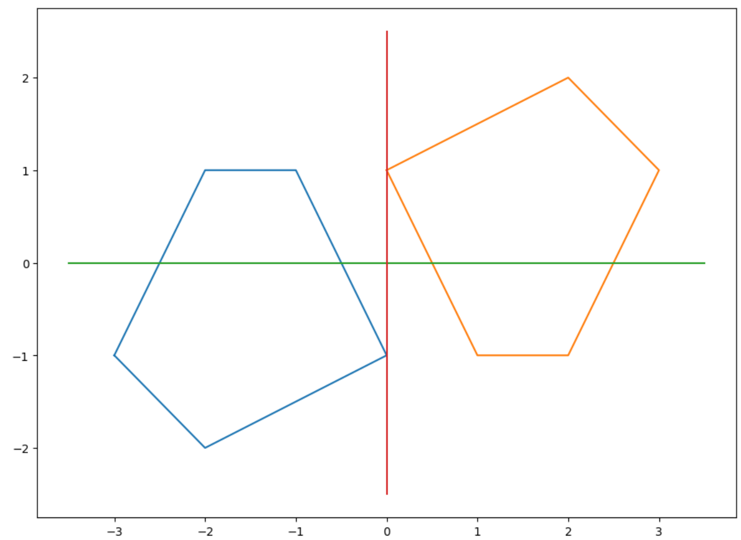
Рыжий – оригинал, он перешёл в зелёный отрезок.

3)



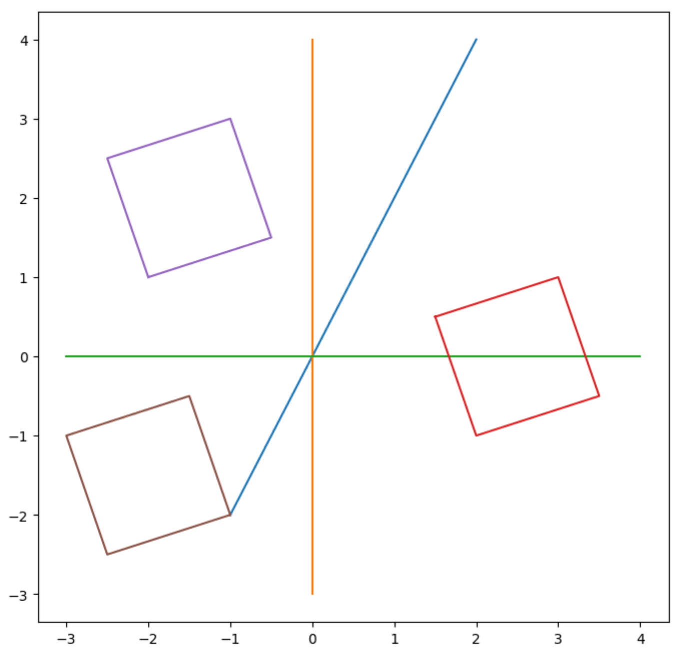
Рыжий – оригинал, синий – его образ

4)



Рыжий – оригинал, синий – его образ (но они переходят друг в друга вообще)

5)



Красный квадратик отразили, получили фиолетовый. Фиолиетовый повернули на 90 градусов – получили коричневый.

6)

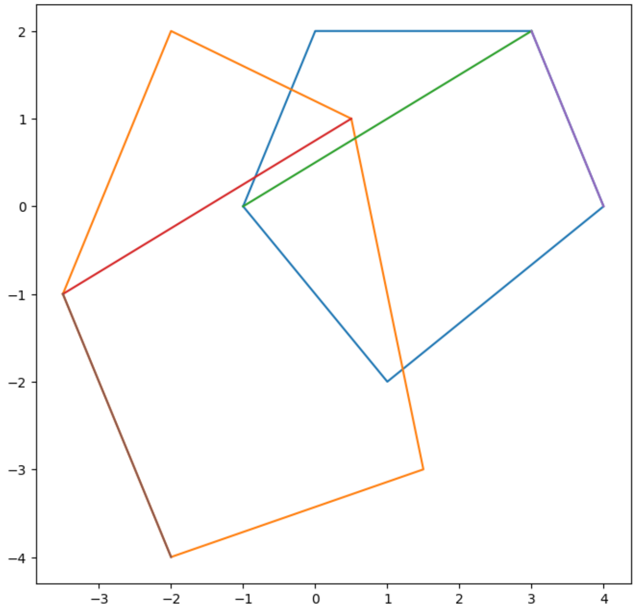
7)

8)

9)

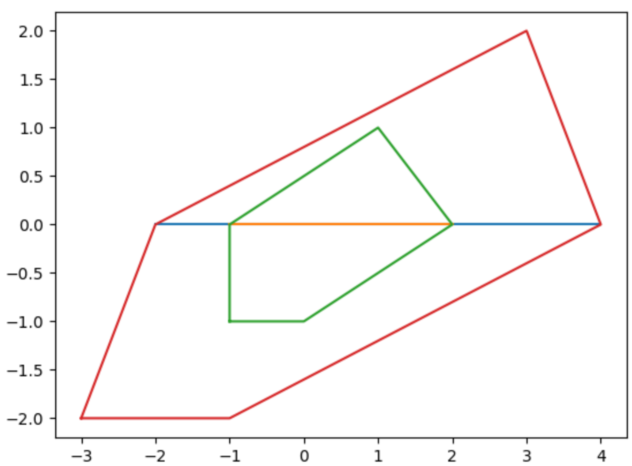
10)

11) !



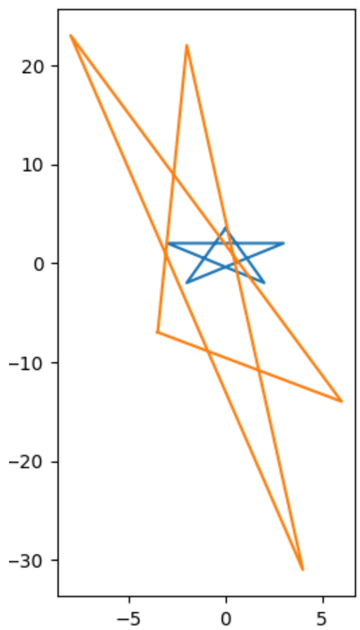
Синий – оригинал, рыжий – его образ. Причём зелёный и красный отрезки остались параллельными, аналогично – фиолетовое и коричневое рёбра. Это произошло потому, что они коллинеарны собственным векторам матрицы данного отображения и не меняют направления, меняют только длину в соответствии с собсвенными числами.

12) !



Зелёный – оригинал, красный – его образ. У данного отображения только один собственный вектор. и вот, не изменили направление только рыжий отрезок, перейдя в синий.

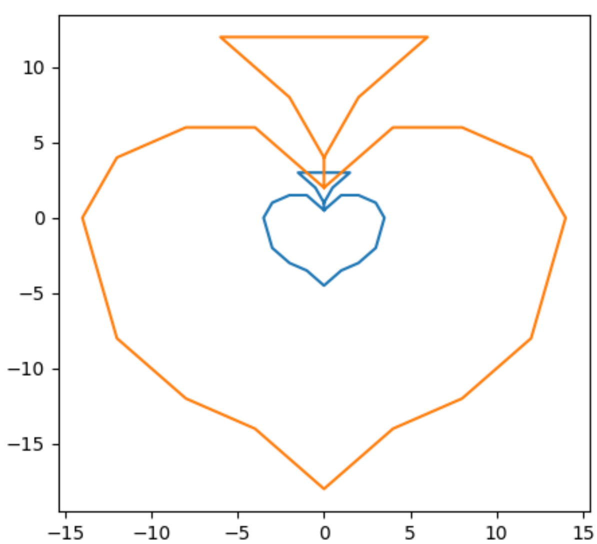
13)



У этого отображения нет вещественных собственных векторов, то есть все векторы будут поворачиваться. Оригинал синий, его образ рыжий.

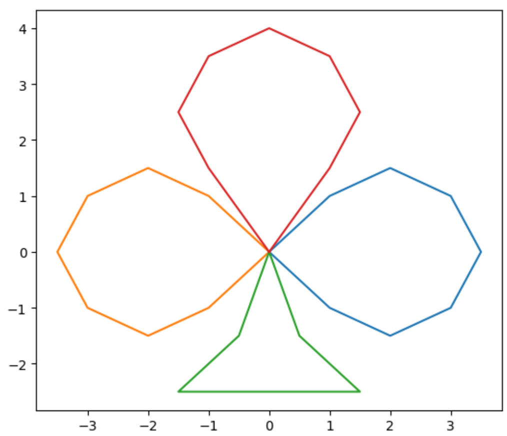
14) !

Для этого отображения любой вектор является собственным.

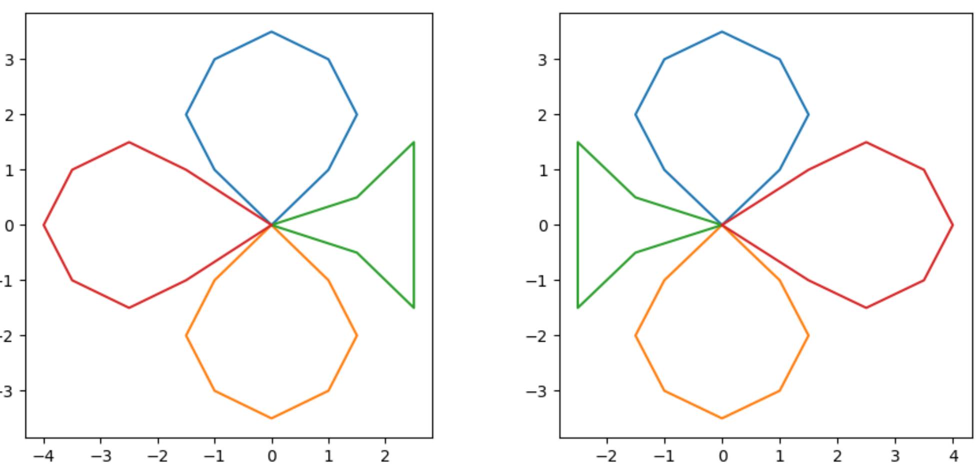


15) ! Собственные векторы этого отображения комплексные, поэтому изобразить их не представляется возможным.

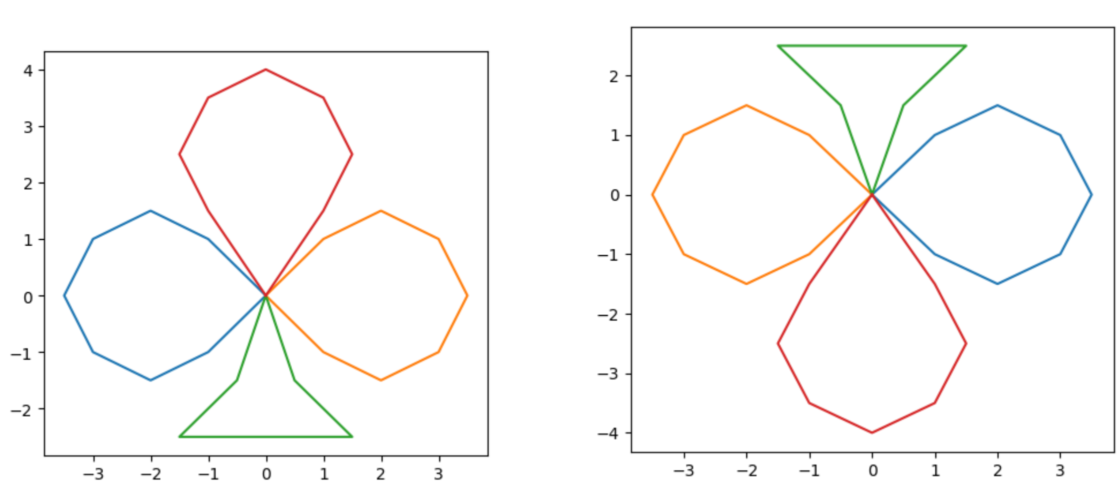
Исходное изображение:



Действие матриц A, B



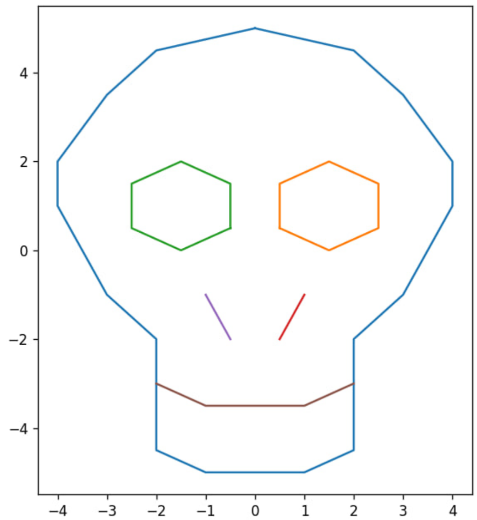
Действие матриц AB и BA (отражение по вертикали и по горизонтали)

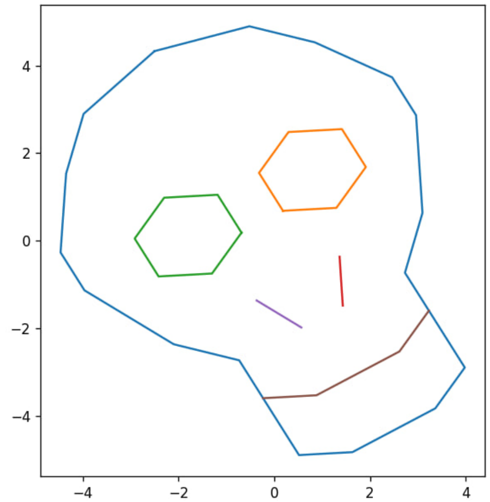


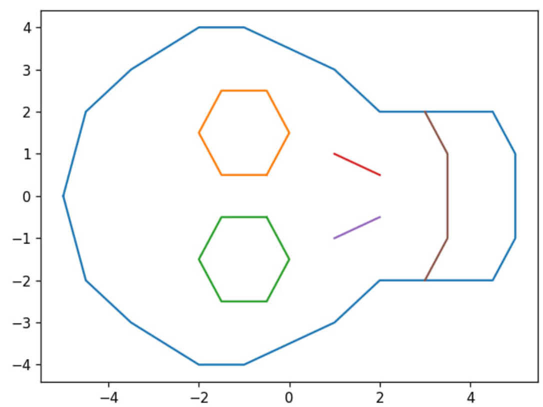
16) !

Собственные векторы этого отображения комплексные, поэтому изобразить их не представляется возможным.

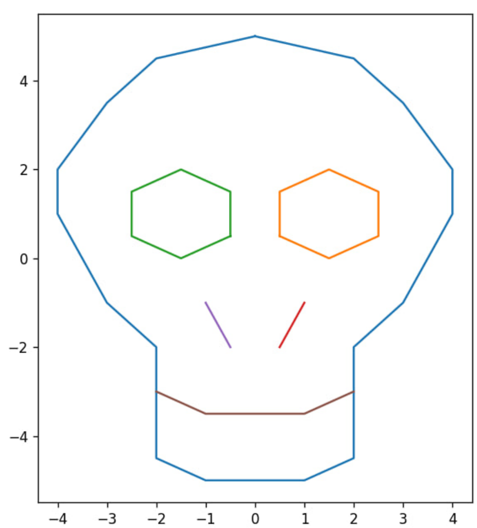
Сначала на 30, потом на 60:

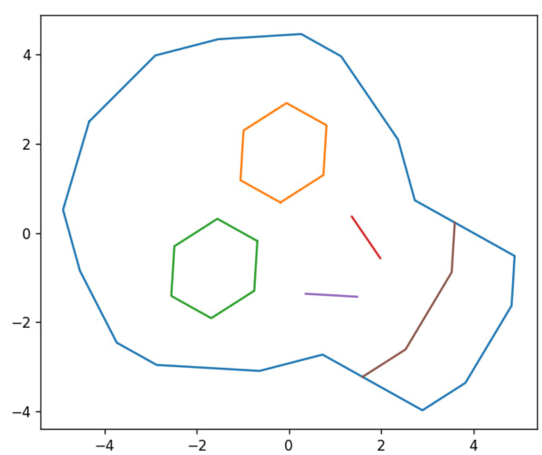


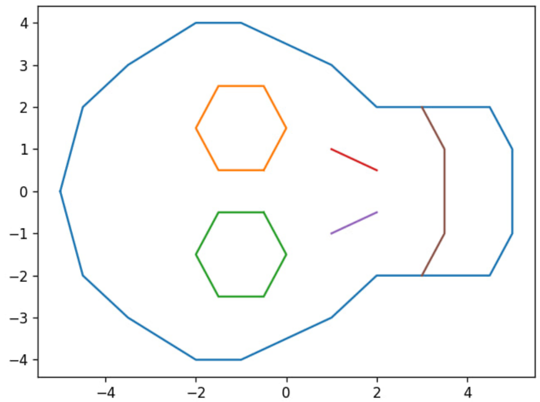




Сначала на 60, потом на 30:







я вертела это всё короче

А здесь мой уважаемый читатель может наконец увидеть небольшую выжимку из самых важных функций моего проекта.

1) Вспомогательные функции для подсчёта определителя (c++)

```cpp
/* Calculates minor of the matrix.
*   Arguments:
*     - pointer to the matrix nxm:
*       matrix* M.
*     - indexes of row and column to be deleted:
*       size_t x, size_t y.
*   Returns:
*     (matrix) - minor matrix (n-1)x(m-1)
*/
matrix math::matr_minor(matrix* M, size_t x, size_t y) {
    size_t n = M->n, m = M->m;
    double** a = M->arr;
    double** res = allocate_memory(n - 1, m - 1);
    size_t i_m = 0, j_m;
    for (size_t i = 0; i < n; i++) {
        if (i == x) continue;
        j_m = 0;
        for (size_t j = 0; j < m; j++) {
            if (j != y && i_m < (n - 1) && j_m < (m - 1)) {
                res[i_m][j_m] = a[i][j];
                j_m++;
            }
        }
        if (i != x) {
            i_m++;
        }
    }
    return matrix{ n - 1, m - 1, res };
}
```

```cpp
/* Calculates the determinant.
*   Arguments:
*     - pointer to the matrix:
*       matrix *M.
*   Returns:
*     (double) - the determinant.
*/
double math::determinant(matrix* M) {
    size_t n = M->n;
    if (n != M->m) {
        std::cout << "This matrix is not squared, determinant cannot be calculated."
            << std::endl;
    }
    double** a = M->arr;
    if (M->n == 1) return a[0][0];
    if (M->n == 2) return a[0][0] * a[1][1] - a[0][1] * a[1][0];

    double res = 0, coef = 1;
    for (size_t i = 0; i < n; i++) {
        matrix cur_minor = matr_minor(M, 0, i);
        res += coef * a[0][i] * determinant(&cur_minor);
        coef *= -1;
    }
    return res;
}
```

2) Функции для работы с матрицами для первого задания (c++)

```cpp
/* There is an equation A * B = C. Get matrix A.
*   Arguments:
*     - matrix B:
*       matrix B,
*     - matrix C:
*       matrix C.
*   Returns:
*     (matrix) - matrix A.
*/
matrix math::get_matrix(matrix B, matrix C) {
  return matr_mul_matr(C, matr_invert(&B));
}
```

```cpp
/* Calculates inverted matrix.
*   Arguments:
*     - pointer to matrix to be inverted:
*       matrix* M.
*   Returns:
*     (matrix) - the inverted matrix.
*/
matrix math::matr_invert(matrix* M) {
  size_t n = M->n, m = M->m;
  if (n != m) {
    std::cout << "This matrix cannot be inverted." << std::endl;
    return matrix{ 0, 0, NULL };
  }
  double** arr = allocate_memory(n, m);
  double coef;
  // adjusted
  for (size_t i = 0; i < n; i++) {
    for (size_t j = 0; j < n; j++) {
      if (i % 2 != j % 2) {
        coef = -1;
      }
      else coef = 1;
      matrix minorr = matr_minor(M, i, j);
      arr[i][j] = determinant(&minorr) * coef;
    }
  }
  // transpose
  matrix res = matr_transpose(matrix{ n, m, arr });
  // multiply to inv to det
  return num_mul_matr(1.0 / determinant(M), res);
}
```

```
/* Multiplies two matrices. Validates sizes.
 *   Arguments:
 *     - first matrix:
 *       matrix A,
 *     - second matrix:
 *       matrix B.
 *   Returns:
 *     (matrix) - result of multiplication.
 *   Notes:
 *     Returns { 0, 0, NULL }, if matrices are incompatible.
 */
matrix math::matr_mul_matr(matrix A, matrix B) {
  size_t n = A.n, m = B.m, l = A.m;
  // validate sizes
  if (l != B.n) {
    printf("Those matrices cannot be multiplied\n");
    printf("A: %zux%zu, B: %zux%zu\n", n, l, B.n, m);
    matrix res = { 0, 0, NULL };
    return res;
  }
  double** a = A.arr, ** b = B.arr;
  double** res = allocate_memory(n, m);
  // calculations
  for (size_t i = 0; i < n; i++) {
    for (size_t j = 0; j < m; j++) {
      res[i][j] = 0;
      for (size_t k = 0; k < l; k++) {
        res[i][j] += a[i][k] * b[k][j];
      }
    }
  }
  matrix matr = { n, m, res };
  return matr;
}
```