1.2 Matrix-Vector Multiplication

As a vector is simply a column of numbers, a matrix is simply a row of columns, or a column of rows. This necessarily requires two numbers, the row and column indices, to specify each matrix element. For example

$$A = \begin{bmatrix} A[0,0] & A[0,1] \\ A[1,0] & A[1,1] \end{bmatrix} = \begin{bmatrix} 5 & 0 \\ 2 & 3 \end{bmatrix} \tag{1}$$

is a 2-by-2 matrix. It takes two indices to address an element, and these are always ordered $[i, j]$ where i denoted the **row** and j denotes the **column**.

The addition of two matrices and the multiplication of a matrix by a scalar proceed exactly as in the vector case. In particular,

$$(A+B)[i,j] = A[i,j] + B[i,j], \qquad \text{e.g.,} \quad \begin{bmatrix} 5 & 0 \\ 2 & 3 \end{bmatrix} + \begin{bmatrix} 2 & 4 \\ 1 & -3 \end{bmatrix} = \begin{bmatrix} 7 & 4 \\ 3 & 0 \end{bmatrix},$$

and

$$(cA)[i,j] = cA[i,j], \qquad \text{e.g.,} \quad 3\begin{bmatrix} 5 & 0 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 15 & 0 \\ 6 & 9 \end{bmatrix}.$$

If A is 2-by-2 and $x \in \mathbb{R}^2$ we **define** $y = Ax$ to be the column of inner products of each row of A with x. In particular, denoting the ith row of A by $A[i,:]$ we find

$$y[i] = A[i,:]x = A[i,0]x[0] + A[i,1]x[1] \tag{2}$$

For example

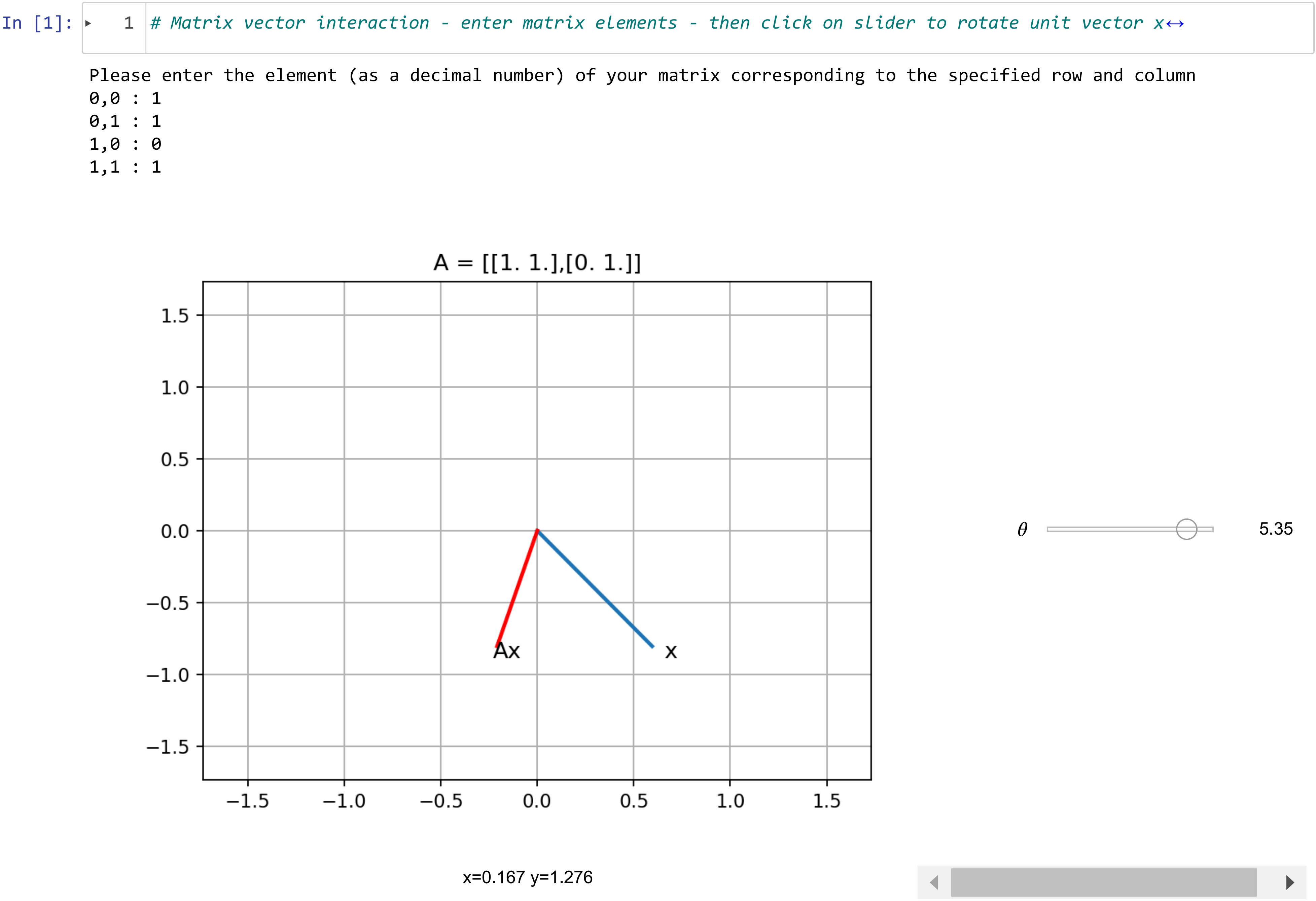
$$\begin{bmatrix} 2 & 3 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 2 \cdot 1 + 3 \cdot (-1) \\ 1 \cdot 1 + 4 \cdot (-1) \end{bmatrix} = \begin{bmatrix} -1 \\ -3 \end{bmatrix}. \tag{3}$$

We see that this A takes planar vectors to planar vectors. The arithmetic is simple but static. We can use python to visualize any such planar transformation. In the widget below, please first enter a 2-by-2 matrix, A, and then move the slider to rotate x round the unit circle and observe the associated movement of Ax.

I recommend that you experiment, in the widget below, with

$$\textbf{dilations}: \ A(\alpha) = \begin{bmatrix} \alpha & 0 \\ 0 & 1/\alpha \end{bmatrix} \qquad \textbf{shears}: \ N(\sigma) = \begin{bmatrix} 1 & \sigma \\ 0 & 1 \end{bmatrix} \qquad \text{and} \qquad \textbf{rotations}: \ K(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{4}$$

for several values of α, σ, θ.



In [1]:

```
# Matrix vector interaction - enter matrix elements - then click on slider to rotate unit vector x↔
```

```
Please enter the element (as a decimal number) of your matrix corresponding to the specified row and column
0,0 : 1
0,1 : 1
1,0 : 0
1,1 : 1
```

Figure 1 *An interactive matrix-vector multiplier. After responding to the prompts for the four matrix elements, use the θ slider to move the blue x vector along the unit circle. The widget plots, in red, the transformed vector Ax.*

This widget hopefully justifies our uses of the words dilation, shear, and rotation. With these words in mind the construction of the associated matrices in (4) comport with their action. In particular, with $\alpha > 1$, the matrix $A(\alpha)$ stretches in the horizontal direction and compresses in the vertical direction. Similarly, with $\sigma > 0$, the matrix $N(\sigma)$ stretches in the horizontal direction while leaving the vertical direction unchanged. Regarding the putative rotation matrix, $K(\theta)$, although $\cos(\theta)$ and $\sin(\theta)$ are indeed suggestive, their particular layout is not exactly obvious.

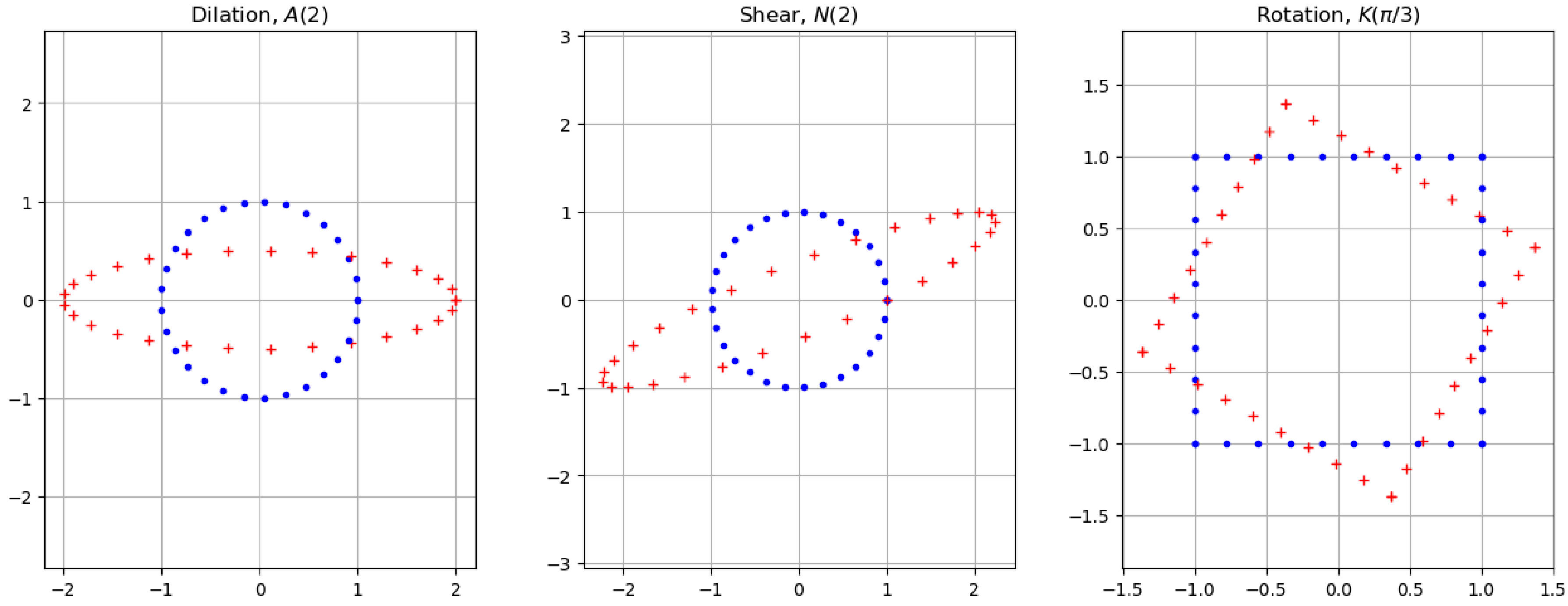
Exercise 1 Given a vector x and angle θ we build a matrix R that rotates x counter-clockwise by θ to the vector y. Write

$$x = \begin{bmatrix} r\cos(\phi) \\ r\sin(\phi) \end{bmatrix} \quad \text{and} \quad y = \begin{bmatrix} r\cos(\theta+\phi) \\ r\sin(\theta+\phi) \end{bmatrix} \tag{5}$$

and then note $y = Rx$ is two equations for the four elements of R. As these equations are to hold for each ϕ. Show that the two easy choices, $\phi = 0$ and $\phi = \pi/2$, lead to $R = K(\theta)$.

The very responsiveness of our widget may actually defeat your ability to see what type of curve is traced by Ax as x traverses the unit circle. We therefore offer a static version below of the transformation of points on the unit disk by representative dilations, shears, and rotations.

```
In [21]: ▸ # Matrix Vector Multiplication↔
```


Dilation, $A(2)$

Shear, $N(2)$

Rotation, $K(\pi/3)$

Figure 2 *An illustration of the matrix vector multiplication conducted in (4). The first two panels show the transformation of the unit circle by a dilation of 2 and shear of 2. As the rotation of a circle is hard to see we instead, in the third panel, rotate a square by $\pi/3$.*

These before and after figures give the gist of the transformation but fail to track individual points. We fix that below.

```
In [10]:   1 # Track the shear of the unit circle↔
```

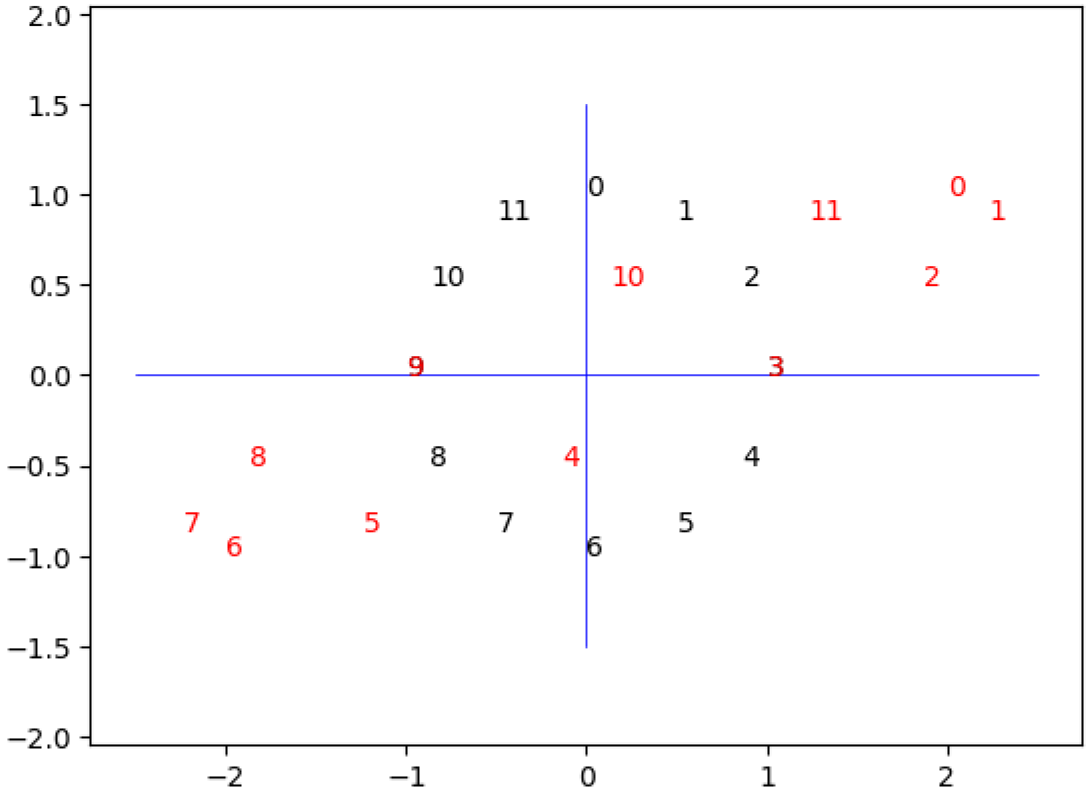


Figure 3 *Illustration of the action* ***shear by 2****, i.e.,* $N(2)x$*, in red, for the twelve* x *vectors, in black. Do you see that* $N(2)$ *takes the black 0 to the red 0, the black 1 to the red 1 and so on? Yes, both the black 3 and black 9 remain unmoved by* $N(2)$*. Why?*

Let us take look at this code. If your code does not come with line numbers please select "View" at the top of this page and "Toggle Line Numbers."

- Python is a small language with a large set of expansion modules. Regarding lines 2 and 3 we import two modules, **numpy** for numerical work, and **matplotlib.pyplot** for graphics, using the two standard abbreviations.
- Graphical output can be directed to several possible places via the **%matplotlib** command. We have directed output to occur **inline,** that is within the notebook itself, via line 4.
- On lines 5 and 6 we plot two lines to serve as our horizontal and vertical axes.
- On lines 9 and 10 we build our shear matrix, N.

- On lines 12 through 18 we step through the hours of the clock, via a **for** loop, that
  - constructs an angle based on the value of hour, line 13
  - constructs a unit vector, x, with this angle, lines 14 and 15
  - shears this unit vector to y=Nx, line 16 (matrix-vector multiplication via @)
  - plots the hour at x and the sheared hour at y, lines 17 and 18
- finally we set the aspect ratio and title the graph, lines 20 and 21.

Exercise 2 Copy our clock code to the cell below and edit it to track minutes rather than hours by replacing each 12 with 60. Note that the default fontsize used in lines 17 and 18 is too large. Add a final argument to each plt.text call to set the fontsize to a readable value. Explore shear values from -2.5 up to 2.5.

In [4]:

```
# Your code here
```

We have so far considered the transformation of single points and simple shapes. In our next example we transform each point in a gray-scale image. Such an image has gray-scale value between 0 (black) and 255 (white) at each pixel. In general the image, img, has m rows and n columns with the $(0,0)$ pixel at top left. We shift the origin to the center of the image and so transform pixel coordinates $[i,j]$ to spatial coordinates via

$$x = [j - int(n/2), int(m/2) - i] \tag{6}$$

we then transform this point with our matrix A via $y = Ax$ and then transform these spatial coordinates into pixel coordinates in our final image, img2, of size $M \times N$, via

$$\begin{aligned} [I,J] &= [int(M/2) - int(y[1]), int(y[0]) + int(N/2)] \\ img2[I,J] &= img[i,j] \end{aligned} \tag{7}$$

We implement these steps in the code cell below, with a bit of preliminary work to first find the dimensions of the transformed image (by transforming the edges of the original) and a bit of clean-up work to paint "missing" pixels in the transformed image.

In [25]:

```
# Transform an image↔
```

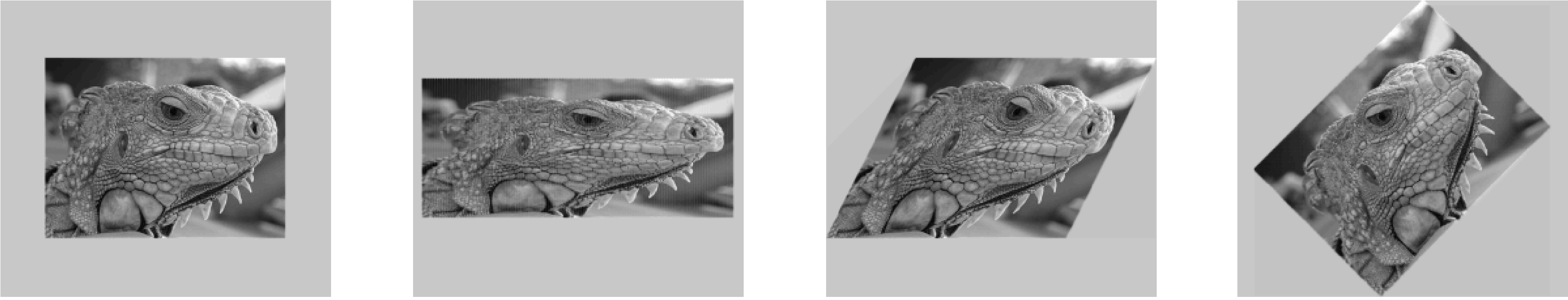


Figure 4 *An image of a lizard, normal at left and then dilated, sheared and rotated as we move left to right.*

Exercise 3 Please adapt the code cell above to transform color images by transforming each of its three color planes.

We surmise, on studying Figures 4 and 2, that dilations, shears, and rotations preserve area. That is, when transforming shapes with one of these matrices, the new shape has the same area as the original. In order to make this precise, we return to this Figure (1.1.VectorOrientation.ipynb/#mf2) from our previous notebook and stack its two columns into the matrix

$$M = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \tag{8}$$

This matrix transforms the square $\{0 \le x \le 1,\ 0 \le y \le 1\}$ to the parallelogram of the aforementioned Figure. That is, M transforms a shape with area 1 to a shape with area $ad - bc$. As such, we name this combination of elements the **determinant** of M and abbreviate it by

$$\boxed{\det(M) \equiv ad - bc} \tag{9}$$

and note that $\det(M)$ encodes the area multiplier. We offer a concrete example of this in Figure 5.

Exercise 4 Please confirm that $\det(A(\alpha)) = \det(N(\sigma)) = \det(K(\theta)) = 1$.

Your solution here.

```
In [7]: ▸  # impact of column swapping on orientation and area↔
```

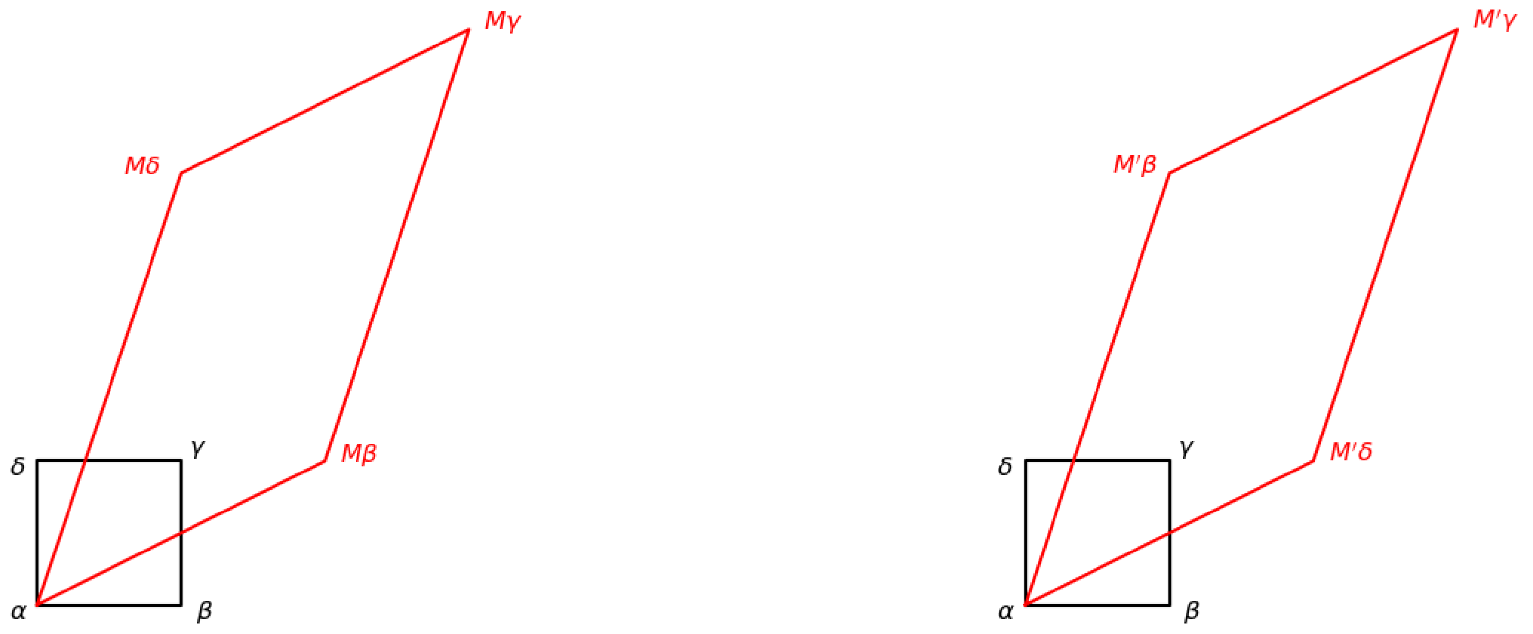


Figure 5 *The deformation of the black square,* $\{(x, y) : 0 \le x \le 1,\ 0 \le y \le 1\}$, *with boundary orientation* $\{\alpha, \beta, \gamma, \delta\}$, *by*

$$M = \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix} \quad \text{at left, and} \quad M' = \begin{pmatrix} 1 & 2 \\ 3 & 1 \end{pmatrix} \quad \text{at right}$$

We observe that they each transform the square to the parallelogram of area $2 \cdot 3 - 1 \cdot 1 = 5$, *but that* M *retains the original orientation while* M' *reverses it.*

Figure 5, by contrasting the action of M and M', the matrix obtained by swapping the columns of M, reveals that although they each map the unit square to the same parallelogram, that $\det(M') = -\det(M)$. As such, the true area multiplier is $|\det(M)|$.

Figure 5 also provides a means for seeing when a determinant is, positive, negative, or zero. In particular, again with regard to Figure 5 we note that the second column of M is counterclockwise from the first column - in agreement with the convention for orientation laid down in our previous notebook (1.1.VectorOrientation.ipynb), while this is reversed when we swap the order of the two columns. As a result; $\det(M) > 0$ when its 2nd column is *counterclockwise* from the first, $\det(M) < 0$ when its 2nd column is *clockwise* from the first. Finally, so long as its two columns are not colinear the parallelogram in Figure 5 will have a nonzero area. Let us see that this is the only way to achieve zero area or, equivalently, a zero determinant. That is, let us show that if $\det(M) = 0$ then its second column is a multiple of its first.

If $\det(M) = 0$ then $ad = bc$ and we recognize a two cases.

(i) $ad \neq 0$ then either is bc and so we may divide $ad = bc$ by bd and achieve

$$\frac{a}{b} = \frac{c}{d} \tag{10}$$

and call this common value t. From (10) we learn that $b = ta$ and $c = td$ and that when stacked this reads

$$\begin{bmatrix} b \\ d \end{bmatrix} = t \begin{bmatrix} a \\ c \end{bmatrix} \tag{11}$$

i.e., the second column of M is a multiple of its first.

Exercise 5 Consider the second case. That is, show that when $ad = bc = 0$ the second column of M remains a multiple of its first.

Your solution here.

In addition to mapping regions, Figure 5 also shows us how the boundary of the region gets mapped. The convention for boundary orientation is that the region stays on our left as we traverse the boundary counterclockwise. Do you see that our choice $\{\alpha, \beta, \gamma, \delta\}$ in Figure 5 for the unit square follows this convention?

In general, if R is a planar region bounded by an oriented curve then M transforms R to a region with area $|\det(M)|\text{area(R)}$ and that the new region retains the orientation of R when $\det(M) > 0$ and reverses the orientation of R when $\det(M) < 0$. As such, **when $\det(M) = 1$ we note that M preserves both area and orientation.**

Next Section: 1.3 Matrix Matrix Multiplication (1.3.MatrixMatrixMultiplication.ipynb)