# 1 Vectors

The classical definition of a vector is quantity that has both a direction and a magnitude. In this class, that definition still holds, but often we will be thinking of a vector as representing the location of a point in a Euclidean coordinate system. So the point $\mathbf{v} = (4, 3)$ could also be represented as the vector

$$\mathbf{v} = \begin{bmatrix} 4 \\ 3 \end{bmatrix}.$$

Here we're using bold font to denote vectors. If we drew an arrow from the origin (0,0) to (4,3), that would represent our vector $\mathbf{v}$. Its direction is the direction of the arrow, and its magnitude $|\mathbf{v}|$ is the length of the arrow, in this case 5. In general, if

$$\mathbf{v} = \begin{bmatrix} x \\ y \end{bmatrix}, \quad \text{then} \quad |\mathbf{v}| = \sqrt{x^2 + y^2}.$$

## 1.1 Vector addition

To add two (or more) vectors together, we can add their $x$ components and $y$ components separately, which has a very visual representation, shown in Figure 1. In this example, adding the components separately gives us

$$\mathbf{v}_1 + \mathbf{v}_2 = \begin{bmatrix} 8 \\ 3 \end{bmatrix} + \begin{bmatrix} 1 \\ 5 \end{bmatrix} = \begin{bmatrix} 8+1 \\ 3+5 \end{bmatrix} = \begin{bmatrix} 9 \\ 8 \end{bmatrix},$$

which is exactly what we obtain from placing the vectors head to tail in the figure.

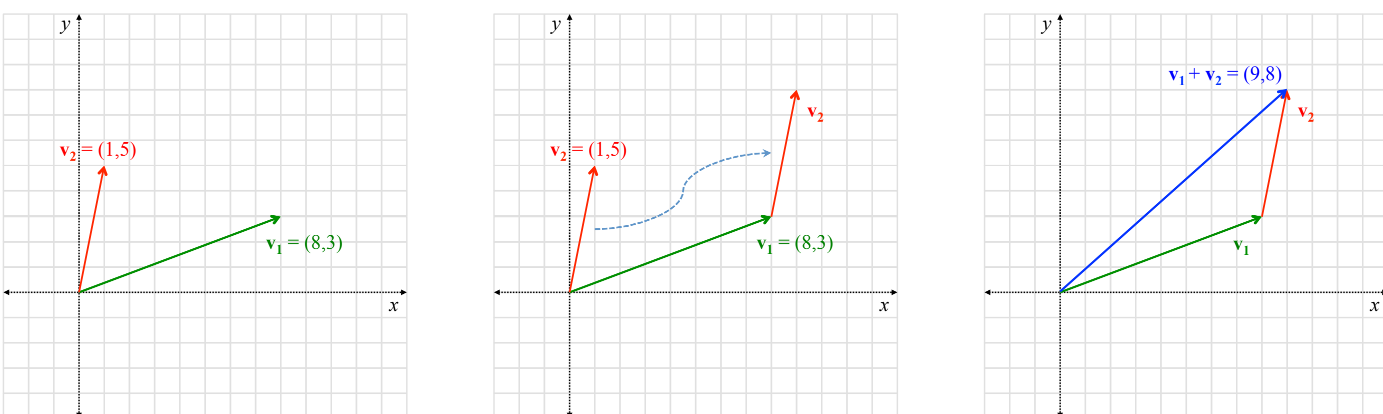



Figure 1: Addition of two vectors $\mathbf{v}_1 = (8, 3)$ in green and $\mathbf{v}_2 = (1, 5)$ in red. If we place the vectors head to tail, the resulting vector is their addition, which is the same as adding the $x$ and $y$ components separately. Either way we get that $\mathbf{v}_1 + \mathbf{v}_2 = (9, 8)$, represented by the blue vector.

We will also be using 3D vectors, but the same notation and algebra still applies. Vector notation displays the components as a column, which is important for matrix-vector multiplication later on. However, to save space we can also use transpose (switch rows and columns) notation:

$$\mathbf{v} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x & y \end{bmatrix}^{\mathrm{T}}$$

# 2 Matrices

In this class, a matrix is an array of numbers, arranged in rows and columns. For us matrices will almost always represent a transformation or a collection of points. It is important to note that a vector is also a matrix. Matrices are not the only way to encode transformations - they happen to be convenient and well-studied for this purpose, which is why we are using them. Usually we denote matrices by a bold capital letter, for example

$$\mathbf{A} = \begin{bmatrix} 4 & 1 & 10 \\ -3 & 2 & 0 \end{bmatrix}$$

The dimension of **A** is ordered "rows × columns", so in this case it is $2 \times 3$.

## 2.1 Matrix addition

Matrices can be added together component-wise just like vectors, provided they are of the *same* dimension. In fact, we can think of each column of a matrix as being a vector (i.e. point). So the following addition of matrices **A** and **B** can be thought of as adding $[a \quad c]^{\mathrm{T}} + [e \quad g]^{\mathrm{T}}$ (first column of each matrix) and $[b \quad d]^{\mathrm{T}} + [f \quad h]^{\mathrm{T}}$ (second column of each matrix):

$$\mathbf{A} + \mathbf{B} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a+e & b+f \\ c+g & d+h \end{bmatrix}.$$

## 2.2 Matrix multiplication

We can also multiply matrices, but this is not as straightforward as addition. Matrix multiplication is defined in a particular way, and later on when we look at different transformations, there will be a better visual representation of what matrix multiplication is doing. Using the matrices from our addition example, we can build up **AB** one entry at a time:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae+bg & \\ & \end{bmatrix} \qquad \begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae+bg & ab+fh \\ & \end{bmatrix}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae+bg & ab+fh \\ ce+dg & \end{bmatrix} \qquad \begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae+bg & ab+fh \\ ce+dg & cf+dh \end{bmatrix}$$

Therefore overall:

$$\mathbf{AB} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae+bg & ab+fh \\ ce+dg & cf+dh \end{bmatrix}.$$

Here we multiplied two $2 \times 2$ matrices, but, unlike addition, we can multiply matrices that do not have exactly the same dimension. However, the number of columns of the first matrix *must* match the number of rows of the second matrix. So we could multiply a $4 \times 2$ matrix and a $2 \times 3$ matrix to obtain a $4 \times 3$ matrix. One way to remember this is that the "inner" dimensions must match (2 in this example), and the "outer" dimensions will be the dimensions of the resulting matrix. Note that swapping the multiplication order in this case would not be valid (we cannot multiply a $2 \times 3$ matrix and a $4 \times 2$ matrix in that order).

This brings us to an important point: matrix multiplication is not commutative, which means that $\mathbf{AB}$ is not necessary equal to $\mathbf{BA}$. Even in the cases when both operations are valid (i.e. two square matrices), they do not always produce the same result. In fact, if we were to select two random square matrices (for some definition of random), it's more likely that $\mathbf{AB} \neq \mathbf{BA}$ than $\mathbf{AB} = \mathbf{BA}$. To practice, multiply the following two matrices in both orders:

$$\mathbf{A} = \begin{bmatrix} 1 & -1 \\ 3 & 2 \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} 0 & 2 \\ -4 & 1 \end{bmatrix}.$$

You should get

$$\mathbf{AB} = \begin{bmatrix} -4 & 1 \\ -8 & 8 \end{bmatrix} \quad \text{and} \quad \mathbf{BA} = \begin{bmatrix} 6 & 4 \\ -1 & 6 \end{bmatrix}.$$

# 3 Transformations

Now we are going to use matrices to represent different transformations of vectors. In fact, every matrix is a transformation in that if we multiply vector $\mathbf{v}$ by matrix $\mathbf{T}$, we'll obtain some new vector $\mathbf{v}'$. In this class, we want the transformations to mean something intuitive, like rotating a ball or reflecting an image in a mirror. So we will use a specific set of transformation matrices to manipulate digital images in interesting ways.

## 3.1 Reflect

The first transformation we'll consider is reflection across an axis, specifically the $y$-axis. Imagine we wanted to reflect the vector $\mathbf{v} = (1, 1)$ across the $y$-axis, as shown in Figure 2. If we imagine doing this, we can probably see that the resulting vector $\mathbf{v}'$ should be (-1,1). How can we get there using a transformation matrix?

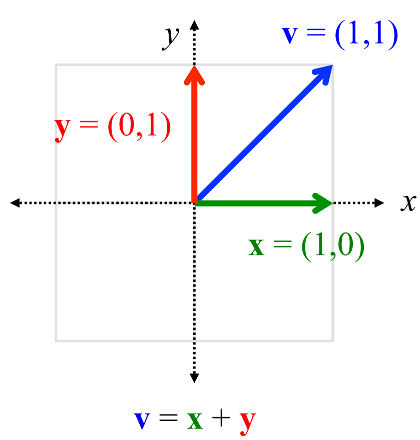



Figure 2: Breaking up the vector $\mathbf{v}$ (shown in blue) into its $x$ (green) and $y$ (red) component vectors.

First we can break up $\mathbf{v}$ into its $x$ and $y$ components, which are also vectors, denoted $\mathbf{x}$ and $\mathbf{y}$. Like our vector addition example, we know that

$$\mathbf{v} = \mathbf{x} + \mathbf{y} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Now we can reflect $\mathbf{x}$ and $\mathbf{y}$ separately, as shown in Figure 3.

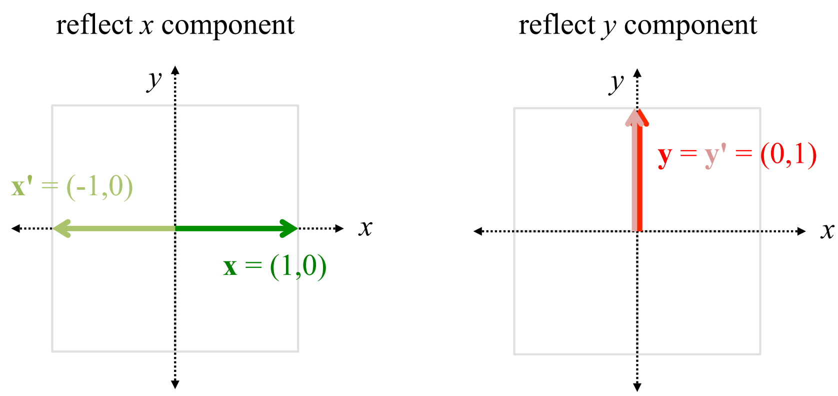



Figure 3: Reflect the $x$ component and $y$ component separately across the $y$-axis. The vector $\mathbf{x}$ (green) becomes $\mathbf{x}'$ (light green), and the vector $\mathbf{y}$ (red) becomes $\mathbf{y}'$ (pink), which is actually the same vector.

Finally, we can add together the $\mathbf{x}'$ and $\mathbf{y}'$ to obtain the reflected result $\mathbf{v}'$, as shown in Figure 4.

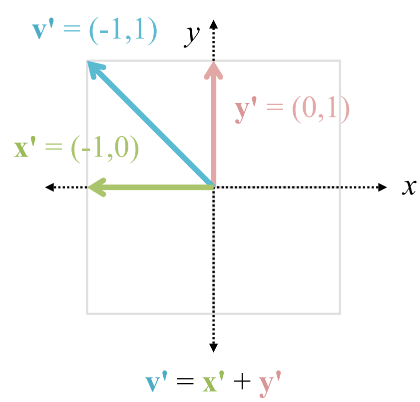



Figure 4: Final vector $\mathbf{v}'$ (shown in light blue) obtained from the reflected $x$ component $\mathbf{x}'$ (light green) and reflected $y$ component $\mathbf{y}'$ (pink).

These figures help us visualize what is going on, but also help us translate the process into a matrix multiplication problem. When we're reflecting, we're always multiplying the $x$ coordinate by -1 and leaving the $y$ coordinate alone. So we can make a small modification to the identity matrix to obtain a reflection matrix:

$$\text{Reflection across the } y\text{-axis matrix:} \quad \mathbf{F}_y = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Now we can multiply $\mathbf{v}$ by this matrix to obtain $\mathbf{v}'$. Component-wise, this looks like

$$\mathbf{F}_y\mathbf{v} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ \phantom{1} \end{bmatrix} = \begin{bmatrix} -1 & + & \\ 0 & + & \end{bmatrix} \qquad x \text{ component vector } \mathbf{x}$$

$$\mathbf{F}_y\mathbf{v} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \phantom{1} \\ 1 \end{bmatrix} = \begin{bmatrix} & + & 0 \\ & + & 1 \end{bmatrix} \qquad y \text{ component vector } \mathbf{y}$$

So the addition part of matrix multiplication is analogous to adding the $x$ and $y$ component vectors as we saw in the figures above. Overall, we obtain

$$\mathbf{F}_y\mathbf{v} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \mathbf{v}'$$

as we expected. And in general, for $\mathbf{w} = (x, y)$:

$$\mathbf{F}_y\mathbf{w} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -x \\ y \end{bmatrix}.$$

Often we will not only want to transform one point, but a set of points or an entire shape. For example, if we wanted to reflect a unit square across the $y$-axis, we could transform all the points at once, using our reflection matrix $\mathbf{F}_y$. Let $\mathbf{S}$ be a matrix denoting the points of our unit square (so each column is a point). To reflect $\mathbf{S}$ to obtain $\mathbf{S}'$, we multiply:

$$\mathbf{F}_y\mathbf{S} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} = \mathbf{S}'.$$

If we plot each of these columns as a point, we should obtain the reflected square, as shown in Figure 5.

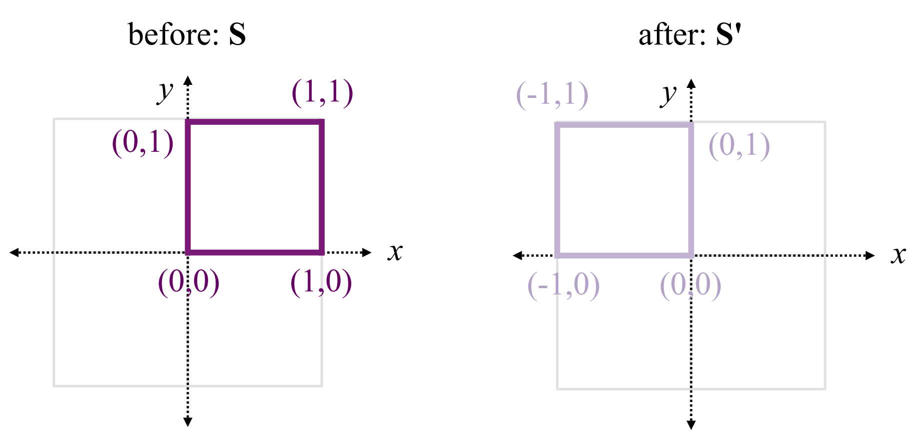



Figure 5: Square $\mathbf{S}$ (purple) reflected across the $y$-axis to obtain square $\mathbf{S}'$ (light purple). Each column of $\mathbf{S}$ is a vertex of the square, which is transformed (i.e. reflected) using the matrix $\mathbf{F}_y$ as shown above.

**Q**: What is the transformation matrix $\mathbf{F}_x$ for reflection across the $x$-axis?

### 3.2 Scale

We can write down a similar transformation matrix for rescaling a vector, called $\mathbf{A}_{(a_x,a_y)}$. Let $a$ be the scaling factor for the $x$ component of the vector, and $b$ be the scaling factor for the $y$ component of the vector. So to transform our generic vector $\mathbf{w} = (x, y)$ into $\mathbf{w}'$, we can multiply:

$$\mathbf{A}_{(a_x,a_y)}\mathbf{w} = \begin{bmatrix} a_x & 0 \\ 0 & a_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_x \cdot x \\ a_y \cdot y \end{bmatrix} = \mathbf{w}'.$$

Figure 6 shows a scaling transformation applied to our unit square, with $a = 2$ and $b = 3$. In matrix notation, this transformation looks like

$$\mathbf{A}_{(2,3)}\mathbf{S} = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 2 & 2 & 0 \\ 0 & 0 & 3 & 3 \end{bmatrix} = \mathbf{S}'.$$

So we can again see that the columns of $\mathbf{S}'$ are the 4 vertices of the transformed square (now a rectangle).

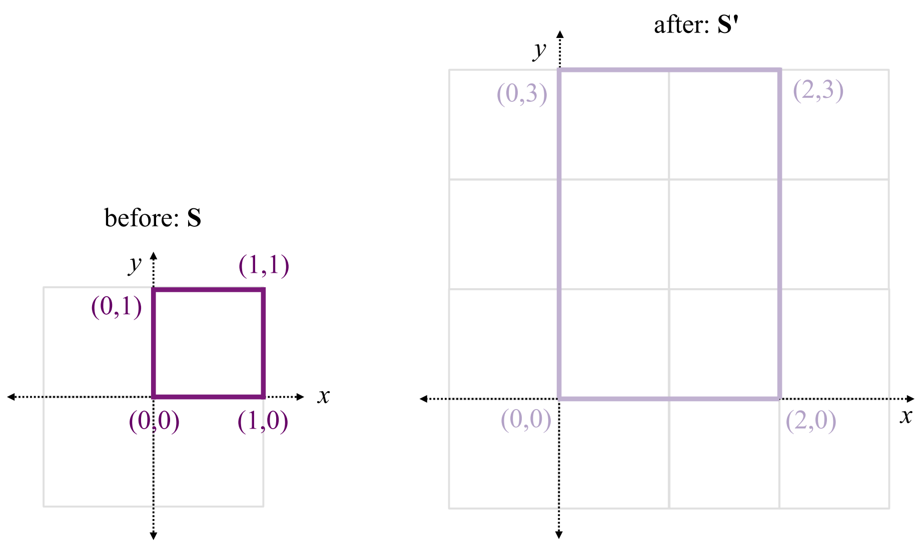



Figure 6: Square $\mathbf{S}$ (purple), with the $x$ component scaled by 2 and the $y$ component scaled by 3 to obtain square $\mathbf{S}'$ (light purple).

### 3.3 Rotate

The next transformation we'll investigate is rotating a vector counterclockwise by an angle $\theta$. Figure 7 shows an example vector $\mathbf{v} = (x, y)$, which we would like to rotate by an angle $\theta$ to obtain $\mathbf{v}'$. As we have done before, we'll decompose $\mathbf{v}$ into its $x$ and $y$ components, shown in green and red.

Now we can rotate each component separately, starting with rotating $\mathbf{x} = (x, 0)$ to obtain $\mathbf{x}'$. In Figure 8 on the left, we can see that the yellow lines and the vector $\mathbf{x}'$ (hypotenuse) form a right triangle. Using right triangle trigonometry, we can write down the coordinates of $\mathbf{x}'$:

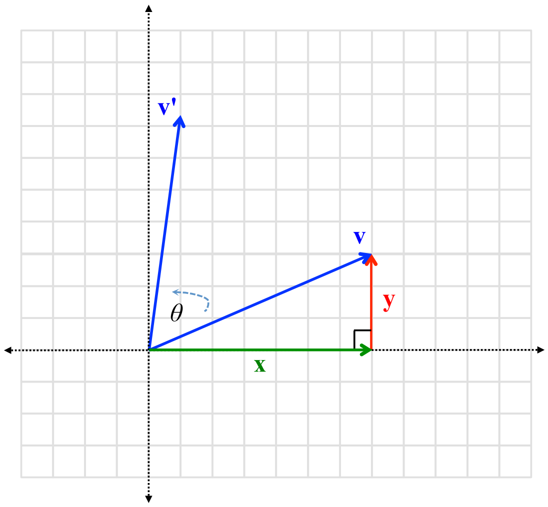
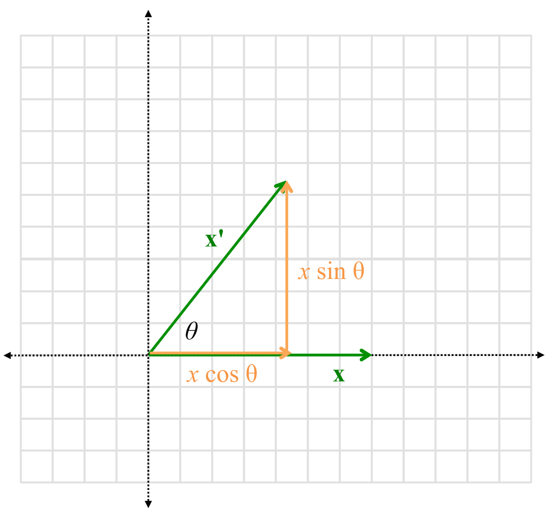



Figure 7: We would like to rotate vector $\mathbf{v} = (x, y)$ (blue) by an angle $\theta$, to obtain $\mathbf{v}'$. First we decompose $\mathbf{v}$ into its $x$ component: $\mathbf{x} = (x, 0)$ (green) and $y$ component: $\mathbf{y} = (0, y)$ (red).

$$\mathbf{x}' = \begin{bmatrix} x\cos\theta \\ x\sin\theta \end{bmatrix}.$$

Similarly, we can rotate the $y$ component (Figure 8 on the right), and find the new coordinates (in purple) of $\mathbf{y}'$ using right triangle trigonometry:

$$\mathbf{y}' = \begin{bmatrix} -y\sin\theta \\ y\cos\theta \end{bmatrix}.$$



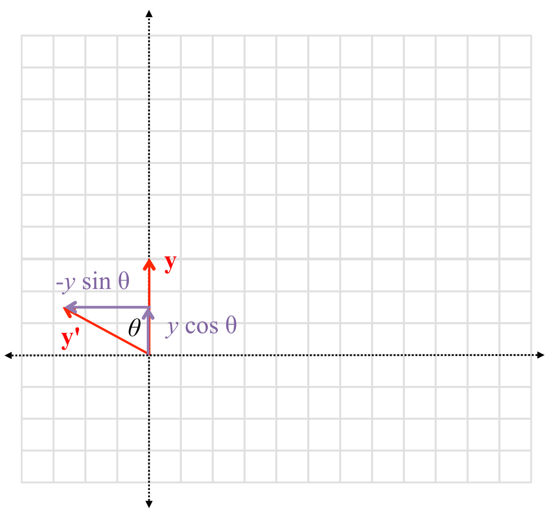




Figure 8: On the left, we've rotated the $x$ component (green) and found the new coordinates (yellow) that make up $\mathbf{x}'$. Similarly, on the right we've rotated the $y$ component (red) and found the coordinates (purple) of $\mathbf{y}'$.

After obtaining these separate components, we can add them together:

$$\mathbf{v}' = \mathbf{x}' + \mathbf{y}' = \begin{bmatrix} x\cos\theta \\ x\sin\theta \end{bmatrix} + \begin{bmatrix} -y\sin\theta \\ y\cos\theta \end{bmatrix} = \begin{bmatrix} x\cos\theta - y\sin\theta \\ x\sin\theta + y\cos\theta \end{bmatrix}.$$

This is shown visually in Figure 9. As a matrix multiplication problem, we can now define a rotation matrix, denoted $\mathbf{R}_\theta$, and make sure it has the correct effect on $\mathbf{v}$:

$$\mathbf{R}_\theta\mathbf{v} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x\cos\theta - y\sin\theta \\ x\sin\theta + y\cos\theta \end{bmatrix}.$$

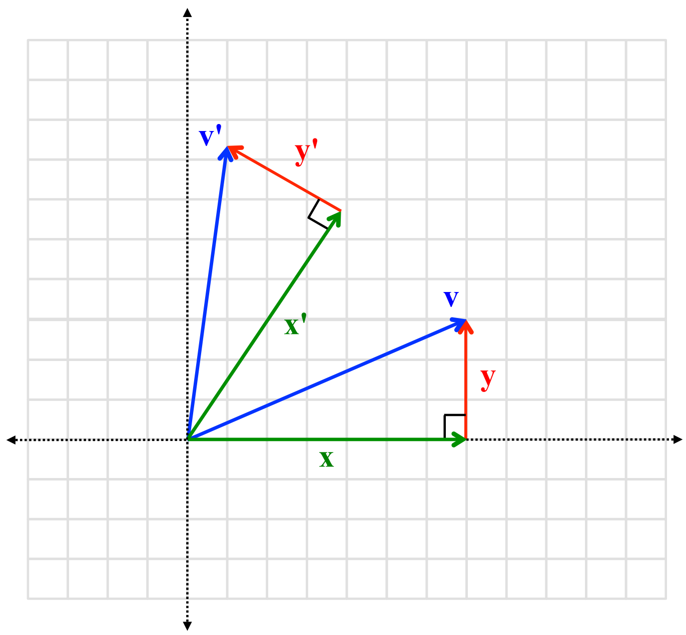



Figure 9: After adding the vectors $\mathbf{x}'$ (green) and $\mathbf{y}'$ (red), we obtain the final rotated vector $\mathbf{v}'$ (blue).

## 3.4 Translate

Translation is a very common transformation which moves objects from one place to another, but does not change their size, shape, or orientation. Translating an object is fundamentally an addition problem. If we want to move an object 2 units to the right and 1 unit up (as shown in Figure 10), then we can add 2 to each $x$ component, and 1 to each $y$ component. To create the square $\mathbf{S}'$ as shown in the figure, we could write:

$$\mathbf{S}' = \begin{bmatrix} 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 3 & 2 \\ 1 & 1 & 2 & 2 \end{bmatrix}.$$

However, this would require creating different addition matrices for different numbers of points. To create a more general way of translating shapes, we need to cast translation as a matrix multiplication problem. To do this, we will augment a 2×2 identity matrix with the desired translation values (usually called $t_x$ and $t_y$), to create a 3×3 translation matrix called $\mathbf{T}_{(t_x,t_y)}$. Then we will also augment each of our points with a 1, which will get multiplied by $t_x$ and $t_y$. This might seem strange and arbitrary, but it achieves our goal of writing an addition problem as a multiplication problem, as shown below:

$$\mathbf{T}_{(t_x,t_y)}\mathbf{v} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix}.$$

Using this form, we can think of all our objects the same way as before, but now if we imagine a 3D coordinate system, they all live in the plane $z = 1$. So to make all our transformations work together, we need to make all 2D transformation matrices 3×3 and all our points need to have their third dimension set to 1.

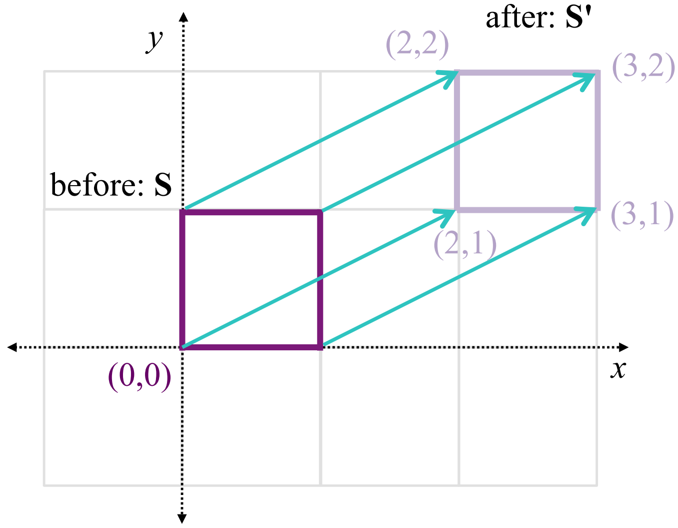



Figure 10: Our square $\mathbf{S}$ after (light purple) a translation with matrix $\mathbf{T}_{(2,1)}$.

Looking at our example square $\mathbf{S}$ again, we can rewrite its translation as a matrix multiplication problem:

$$\mathbf{T}_{(2,1)}\mathbf{S} = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 3 & 2 \\ 1 & 1 & 2 & 2 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \mathbf{S}'.$$

## 3.5 Shear

The last transform we'll investigate is *shear*, which is a way of making an object look like it's leaning over, as shown in Figure 11. Similar to reflection, there are two types of shear - shearing parallel to the $x$-axis (as shown below) and shearing parallel to the $y$-axis. We will denote a shear matrix in the $x$ direction as $\mathbf{S}_{(x,\lambda)}$, where $\lambda$ is the shearing factor (not to be confused with our example square $\mathbf{S}$).

A shear in the $x$ direction will not change the $y$ values at all, but will change the $x$ values *proportional to their $y$ values.* To achieve this effect with a matrix, in this example we can write:

$$\mathbf{S}_{(x,2)}\mathbf{S} = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 3 & 3 \\ 0 & 0 & 1 & 1 \end{bmatrix} = \mathbf{S}'.$$

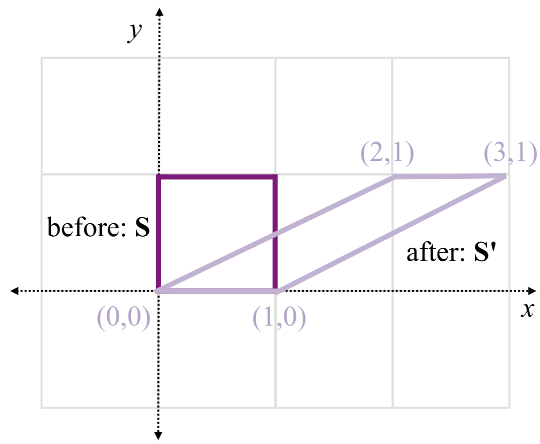



Figure 11: Our square **S** after (light purple) a shear with matrix $\mathbf{S}_{(x,2)}$.

**Q**: What is the transformation matrix $\mathbf{S}_{(y,\lambda)}$ for a shear parallel to the $y$-axis?

## 3.6 List of 2D Transformation Matrices

For each of our transformation matrices besides translation, we need to "augment with the identity" to make them 3×3 matrices. To do this, we'll make both the last row and last column the vector [0, 0, 1], which is the same as the last row and last column of a 3×3 identity matrix.

- Scale: $\mathbf{A}_{(a_x,a_y)} = \begin{bmatrix} a_x & 0 & 0 \\ 0 & a_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$

- Rotate counter-clockwise: $\mathbf{R}_\theta = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$

- Translate: $\mathbf{T}_{(t_x,t_y)} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$

- Reflect across $y$-axis: $\mathbf{F}_y = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

- Shear parallel to $x$-axis: $\mathbf{S}_{(x,\lambda)} = \begin{bmatrix} 1 & \lambda & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$