

# The Geometry of Special Orthogonal Transformations

## Definition 3.1: Givens Rotation

A Givens rotation is an  $n \times n$  matrix of the form

$$Q(i, k, \theta) = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & c & \cdots & s & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & -s & \cdots & c & \cdots & 0 \\ \vdots & & \vdots & & \vdots & & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix}$$

where  $c = \cos(\theta)$  and  $s = \sin(\theta)$ . The Givens rotation matrix has a total of  $(n + 2)$  nonzero entries.

## Rotations about the Origin

As discussed in the Givens Rotation Definition 3.1, left multiplication of begin polygon  $V$  by matrix

$$Q(1, 2, \theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix},$$

rotates each vertex  $V(:, k)$  clockwise by angle  $\theta$  to produce end vertex  $W(:, k)$  for  $k = 1, 2, 3, 4$ .

### EXAMPLE 3.3.1

Let's rotate the begin polygon  $V$  clockwise by a  $60^\circ$  angle. To do so, set  $\theta = \frac{\pi}{3}$  and define

$$E = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{\sqrt{3}}{2} & \frac{1}{2} \end{bmatrix}$$

$$W = E \cdot V = \begin{bmatrix} \frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{\sqrt{3}}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

In this case, we will construct the vertex matrix  $W$  column-by-column. Using the column partition version of matrix vector multiplication, notice

$$W(:,1) = \begin{bmatrix} \frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{\sqrt{3}}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} = 0 \cdot \begin{bmatrix} \frac{1}{2} \\ -\frac{\sqrt{3}}{2} \end{bmatrix} + 0 \cdot \begin{bmatrix} \frac{\sqrt{3}}{2} \\ \frac{1}{2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

We find that the vector of second coordinates of the end polygon is calculated

$$W(:,2) = \begin{bmatrix} \frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{\sqrt{3}}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0 \cdot \begin{bmatrix} \frac{1}{2} \\ -\frac{\sqrt{3}}{2} \end{bmatrix} + 1 \cdot \begin{bmatrix} \frac{\sqrt{3}}{2} \\ \frac{1}{2} \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{2} \\ \frac{1}{2} \end{bmatrix}$$

Next, the third column of  $W$  is

$$W(:,3) = \begin{bmatrix} \frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{\sqrt{3}}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 1 \cdot \begin{bmatrix} \frac{1}{2} \\ -\frac{\sqrt{3}}{2} \end{bmatrix} + 1 \cdot \begin{bmatrix} \frac{\sqrt{3}}{2} \\ \frac{1}{2} \end{bmatrix} = \begin{bmatrix} \frac{1+\sqrt{3}}{2} \\ \frac{1-\sqrt{3}}{2} \end{bmatrix}$$

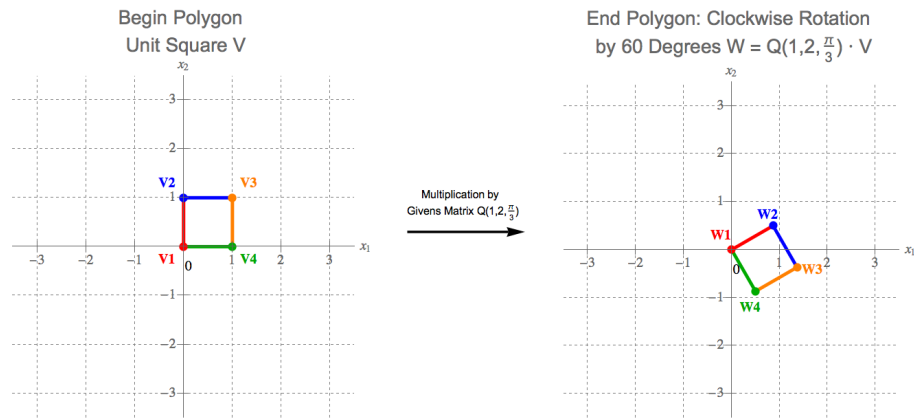
Finally, the last column of our end vertex matrix is given by

$$W(:,4) = \begin{bmatrix} \frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{\sqrt{3}}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = 1 \cdot \begin{bmatrix} \frac{1}{2} \\ -\frac{\sqrt{3}}{2} \end{bmatrix} + 0 \cdot \begin{bmatrix} \frac{\sqrt{3}}{2} \\ \frac{1}{2} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ -\frac{\sqrt{3}}{2} \end{bmatrix}$$

These two vectors together compose the matrix  $W$ , and thus we have

$$W = \begin{bmatrix} 0 & \frac{\sqrt{3}}{2} & \frac{1+\sqrt{3}}{2} & \frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1-\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}$$

The end polygon  $W$  is, indeed, a  $60^\circ$  clockwise rotation of the begin polygon  $V$  as demonstrated in the diagram below.



To rotate our polygon by an angle of  $\theta$  counter clockwise around the origin, we use the Givens rotation

$$G(1, 2, -\theta) = \begin{bmatrix} \cos(-\theta) & \sin(-\theta) \\ -\sin(-\theta) & \cos(-\theta) \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

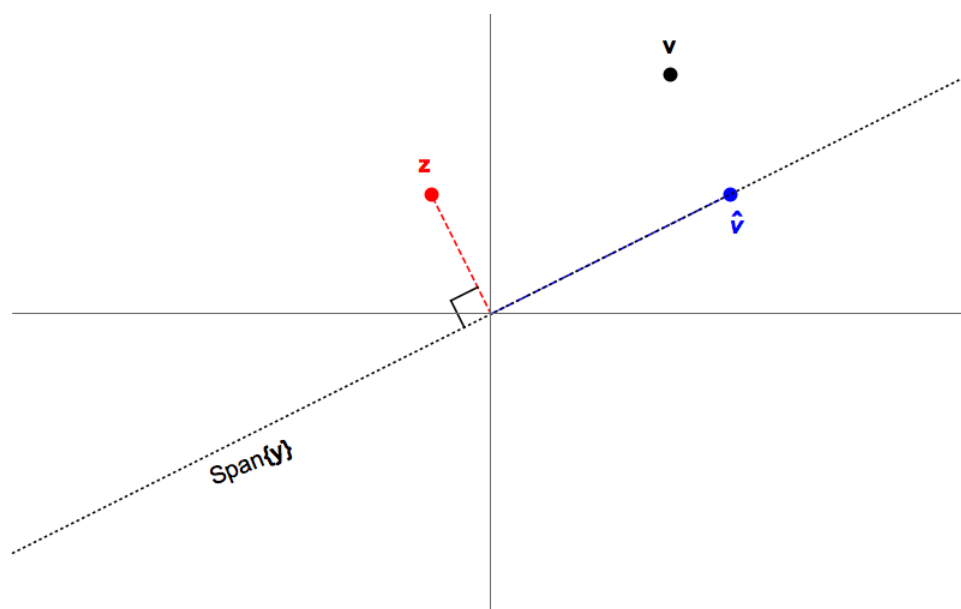
The final equality results from the property that cosine is an even function with  $\cos(-\theta) = \cos(\theta)$  while sine is an odd function with  $\sin(-\theta) = -\sin(\theta)$ . With this in mind, we see that to rotate vectors counter clockwise, we can multiply by the transpose of our Given rotation.

### Projections onto Special Axis

Next, we will consider orthogonal projections of portions of the begin polygon onto a selected line through the origin. Suppose we want to project our shapes onto the span of a given vector  $\mathbf{y} \in \mathbb{R}^2$ . To construct such a projection, let's find a way to decompose any vector  $\mathbf{v} \in \mathbb{R}^2$  into the sum of two vectors

$$\mathbf{v} = \hat{\mathbf{v}} + \mathbf{z}$$

where  $\hat{\mathbf{v}} \in \text{Span}\{\mathbf{y}\}$  and  $\mathbf{z}$  is orthogonal to all vectors in  $\text{Span}\{\mathbf{y}\}$ , as is demonstrated in the figure below.



In other words, we want to construct a matrix  $P_{\mathbf{y}}$  that orthogonally projects any vector onto the  $\text{Span}\{\mathbf{y}\}$  such that

$$\hat{\mathbf{v}} = P_{\mathbf{v}}.$$

Since  $\hat{\mathbf{v}} \in \text{Span}(\mathbf{y})$ , we know that  $\hat{\mathbf{v}} = \alpha \mathbf{y}$  for some scalar  $\alpha \in \mathbb{R}$ . If we can find a closed form equation for the  $\alpha$ , we can construct our projection  $P_{\mathbf{y}}$ . To this end, let's rewrite our equation using the properties of vector operations with

$$\mathbf{z} = \mathbf{v} - \hat{\mathbf{v}}.$$

By the cosine formula for the dot product, we know that scalar  $\alpha$  must be chosen to satisfy the condition that

$$0 = \mathbf{y} \cdot \mathbf{z}$$

Substituting our updated expression for  $\mathbf{z}$  into this orthogonality condition yields

$$0 = \mathbf{y} \cdot (\mathbf{v} - \alpha \mathbf{y}) = \mathbf{y} \cdot \mathbf{v} - \alpha \mathbf{y} \cdot \mathbf{y}$$

The second equality here results from the algebraic properties of the dot product between vectors. We can solve for the scalar  $\alpha$  to find that

$$\alpha = \frac{\mathbf{y} \cdot \mathbf{v}}{\mathbf{y} \cdot \mathbf{y}} = \frac{\mathbf{v}^T \mathbf{y}}{\mathbf{y}^T \mathbf{y}}$$

This produces two projections. First, we consider the projection of  $\mathbf{v}$  onto  $\text{Span}\{\mathbf{y}\}$ , given by

$$\hat{\mathbf{v}} = \alpha \mathbf{y} = \left( \frac{\mathbf{v}^T \mathbf{y}}{\mathbf{y}^T \mathbf{y}} \right) \mathbf{y}$$

We can derive a matrix form of this projection using our algebraic properties of matrix-vector multiplication

$$\begin{aligned} \hat{\mathbf{v}} &= \left( \frac{\mathbf{v}^T \mathbf{y}}{\mathbf{y}^T \mathbf{y}} \right) \mathbf{y} \\ &= \frac{1}{\mathbf{y}^T \mathbf{y}} (\mathbf{v}^T \mathbf{y}) \mathbf{y} \\ &= \frac{1}{\mathbf{y}^T \mathbf{y}} (\mathbf{y}^T \mathbf{v}) \mathbf{y} \\ &= \frac{1}{\mathbf{y}^T \mathbf{y}} \mathbf{y} (\mathbf{y}^T \mathbf{v}) \\ &= \frac{1}{\mathbf{y}^T \mathbf{y}} (\mathbf{y} \mathbf{y}^T) \mathbf{v} \\ &= \left( \frac{\mathbf{y} \mathbf{y}^T}{\mathbf{y}^T \mathbf{y}} \right) \mathbf{v} \end{aligned}$$

This derivation suggests a matrix realization for the projection of any vector onto the span of  $\mathbf{y}$ , given by

$$P_{\mathbf{y}} = \frac{\mathbf{y} \mathbf{y}^T}{\mathbf{y}^T \mathbf{y}}$$

Further, the projection of  $\mathbf{v}$  onto the line perpendicular to  $\text{Span}\{\mathbf{y}\}$  is given by

$$\mathbf{z} = \mathbf{v} - \hat{\mathbf{v}}$$

Using the description of these vectors from above, we see

$$\begin{aligned}
 \mathbf{z} &= \mathbf{v} - \hat{\mathbf{v}} \\
 &= \mathbf{v} - \left( \frac{\mathbf{y} \mathbf{y}^T}{\mathbf{y}^T \mathbf{y}} \right) \mathbf{v} \\
 &= I_2 \mathbf{v} - \left( \frac{\mathbf{y} \mathbf{y}^T}{\mathbf{y}^T \mathbf{y}} \right) \mathbf{v} \\
 &= \left( I_2 - \frac{\mathbf{y} \mathbf{y}^T}{\mathbf{y}^T \mathbf{y}} \right) \mathbf{v}
 \end{aligned}$$

This leads to a matrix realization of the projection of any vector onto the line orthogonal to the  $\text{Span}\{\mathbf{y}\}$  given by

$$P_{\mathbf{y}^\perp} = I_2 - \frac{\mathbf{y} \mathbf{y}^T}{\mathbf{y}^T \mathbf{y}}$$

This gives a systematic approach to creating a projection onto any axis we desire.

### EXAMPLE 3.3.2

Let's project the begin polygon onto the first coordinate axis. To do so, set

$$E = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

Then, we can calculate the end polygon

$$W = E \cdot V = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Again, we carry out a row-by-row calculation to find

$$W = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

### EXAMPLE 3.3.3

To project onto the second coordinate axis, set

$$E = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

Then, we can calculate the end polygon

$$W = E \cdot V = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Again, we carry out a row-by-row calculation to find

$$W = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

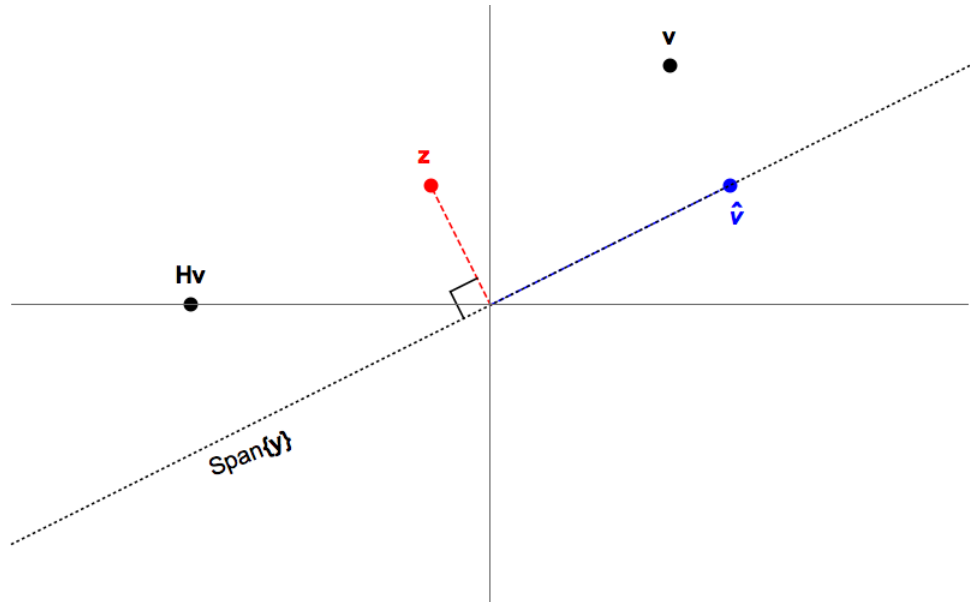
### Householder Reflections

See Trefethen p. 70, Datta p. 183 - 187, Golub and Van Loan p. 209,

This brings us to our final important consideration for reflections: How can we define a reflection about an arbitrary line of reflection defined by the span of vector  $\mathbf{y}$

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

This will be known as a Householder reflector and will be used extensively throughout later sections of this book.



#### Definition 3.2: Householder Reflection

Let  $\mathbf{v} \in \mathbb{R}^n$  be a nonzero vector. A *Householder reflection* is an  $n \times n$  matrix  $P$  of the form

$$P = I_n + \frac{2}{\mathbf{v} \cdot \mathbf{v}} \mathbf{v} \mathbf{v}^T$$

The vector  $\mathbf{v}$  is called a *Householder vector*.

The Householder transformation  $P$  is also called a *Householder transformation* or a *Householder matrix*. These matrices are very powerful tools used to execute numerical calculations on matrices. We will study these in detail during our discussion of matrix-vector multiplication. Householder matrices play an important role when we discuss the numerical properties of algorithms used to solve linear-systems and least-squares problems.

#### EXAMPLE 3.3.4

Let  $n = 4$ . Let's define the Householder matrix  $P$  using the vector  $\mathbf{v} = [4 \ 1 \ 3 \ 2]^T$ .

Then, we see

$$P = I_4 - 2 \frac{\mathbf{v} \mathbf{v}^T}{\mathbf{v} \cdot \mathbf{v}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} - \frac{2}{30} \begin{bmatrix} 16 & 4 & 12 & 8 \\ 4 & 1 & 3 & 2 \\ 12 & 3 & 9 & 6 \\ 8 & 2 & 6 & 4 \end{bmatrix}$$

### Affine Transformations

So far, we have considered transformations that shear, scale, reflect and rotate the begin polygon  $V$ . In all of these cases, the vertex at the origin does not move. However, in some instances, it is very convenient to be able to translate the entire begin polygon by a predefined vector  $\mathbf{x}_0$ . To do so, we craft a different map that sends

$$\mathbf{w} = \mathbf{v} + \mathbf{x}$$

This can be achieved by embedding our two dimensional model into a three dimensional space, as follows

$$\begin{bmatrix} w_1 \\ w_2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_1 \\ 0 & 1 & x_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ 1 \end{bmatrix}$$

Then, we create our end polygon by considering only the first two coordinates of this product.

#### EXAMPLE 3.3.5

Let's translate the begin polygon  $V$  by shifting all vertices by the vector

$$\mathbf{x} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

We encode this map as a matrix-vector product given by

$$\begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_1 \\ 0 & 1 & x_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

We can calculate the product using the column partition version of matrix vector multiplication to find the final.

Notice that the accurate use of affine transformations need not preserve the zero point. In particular, the zero point in the domains space does not necessarily get mapped to the zero point in the codomain.