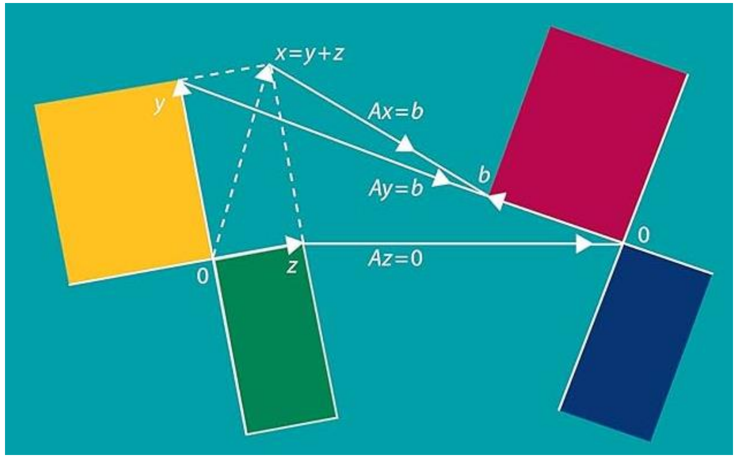
Linear Transformations

(Linear Algebra)

Randil Pushpananda, PhD
rpn@ucsc.cmb.ac.lk



Introduction

When a matrix A multiplies a vector $\boldsymbol{v}$, it "transforms" $\boldsymbol{v}$ into another vector $A\boldsymbol{v}$. ***In goes*** $\boldsymbol{v}$, ***out comes*** $T(\boldsymbol{v}) = A\boldsymbol{v}$. A transformation T follows the same idea as a function. In goes a number x, out comes $f(x)$. For one vector $\boldsymbol{v}$ or one number x, we multiply by the matrix or we evaluate the function. The deeper goal is to see all vectors $\boldsymbol{v}$ at once. We are transforming the whole space $\mathbf{V}$ when we multiply every $\boldsymbol{v}$ by A.

Start again with a matrix A. It transforms $\boldsymbol{v}$ to $A\boldsymbol{v}$. It transforms $\boldsymbol{w}$ to $A\boldsymbol{w}$. Then we *know* what happens to $\boldsymbol{u} = \boldsymbol{v} + \boldsymbol{w}$. There is no doubt about $A\boldsymbol{u}$, it has to equal $A\boldsymbol{v} + A\boldsymbol{w}$. Matrix multiplication $T(\boldsymbol{v}) = A\boldsymbol{v}$ gives an important ***linear transformation*** :

> A ***transformation*** T assigns an output $T(\boldsymbol{v})$ to each input vector $\boldsymbol{v}$ in $\mathbf{V}$.
>
> The transformation is ***linear*** if it meets these requirements for all $\boldsymbol{v}$ and $\boldsymbol{w}$:
>
> (a) $T(\boldsymbol{v} + \boldsymbol{w}) = T(\boldsymbol{v}) + T(\boldsymbol{w})$ (b) $T(c\boldsymbol{v}) = cT(\boldsymbol{v})$ for all c.

If the input is $\boldsymbol{v} = \mathbf{0}$, the output must be $T(\boldsymbol{v}) = \mathbf{0}$. We combine rules (a) and (b) into one :

$$\boxed{\textbf{Linear transformation} \qquad T(c\boldsymbol{v} + d\boldsymbol{w}) \quad \textbf{\textit{must equal}} \quad cT(\boldsymbol{v}) + dT(\boldsymbol{w}).}$$

Again I can test matrix multiplication for linearity: $A(c\boldsymbol{v} + d\boldsymbol{w}) = cA\boldsymbol{v} + dA\boldsymbol{w}$ is *true*.

A linear transformation is highly restricted. Suppose T adds $\boldsymbol{u}_0$ to every vector. Then $T(\boldsymbol{v}) = \boldsymbol{v} + \boldsymbol{u}_0$ and $T(\boldsymbol{w}) = \boldsymbol{w} + \boldsymbol{u}_0$. This isn't good, or at least *it isn't linear*. Applying T to $\boldsymbol{v} + \boldsymbol{w}$ produces $\boldsymbol{v} + \boldsymbol{w} + \boldsymbol{u}_0$. That is not the same as $T(\boldsymbol{v}) + T(\boldsymbol{w})$:

$$\textbf{Shift is not linear} \qquad \boldsymbol{v} + \boldsymbol{w} + \boldsymbol{u}_0 \quad \text{is not} \quad T(\boldsymbol{v}) + T(\boldsymbol{w}) = (\boldsymbol{v} + \boldsymbol{u}_0) + (\boldsymbol{w} + \boldsymbol{u}_0).$$

Examples

Example 1 Choose a fixed vector $\boldsymbol{a} = (1, 3, 4)$, and let $T(\boldsymbol{v})$ be the dot product $\boldsymbol{a} \cdot \boldsymbol{v}$:

$$\boxed{\text{The input is} \quad \boldsymbol{v} = (v_1, v_2, v_3). \qquad \text{The output is} \quad T(\boldsymbol{v}) = \boldsymbol{a} \cdot \boldsymbol{v} = v_1 + 3v_2 + 4v_3.}$$

Dot products are linear. The inputs $\boldsymbol{v}$ come from three-dimensional space, so $\mathbf{V} = \mathbf{R}^3$. The outputs are just numbers, so the output space is $\mathbf{W} = \mathbf{R}^1$. We are multiplying by the row matrix $A = \begin{bmatrix} 1 & 3 & 4 \end{bmatrix}$. Then $T(\boldsymbol{v}) = A\boldsymbol{v}$.

You will get good at recognizing which transformations are linear. If the output involves squares or products or lengths, v_1^2 or $v_1 v_2$ or $\|\boldsymbol{v}\|$, then T is not linear.

Examples

Example 2 The length $T(\boldsymbol{v}) = \|\boldsymbol{v}\|$ is not linear. Requirement (a) for linearity would be $\|\boldsymbol{v}+\boldsymbol{w}\| = \|\boldsymbol{v}\| + \|\boldsymbol{w}\|$. Requirement (b) would be $\|c\boldsymbol{v}\| = c\|\boldsymbol{v}\|$. Both are false!

Not (a): The sides of a triangle satisfy an *inequality* $\|\boldsymbol{v}+\boldsymbol{w}\| \leq \|\boldsymbol{v}\| + \|\boldsymbol{w}\|$.
Not (b): The length $\|-\boldsymbol{v}\|$ is $\|\boldsymbol{v}\|$ and not $-\|\boldsymbol{v}\|$. For negative c, linearity fails.

Example 3 (Rotation) T is the transformation that *rotates every vector by* 30°. The "*domain*" of T is the xy plane (all input vectors $\boldsymbol{v}$). The "*range*" of T is also the xy plane (all rotated vectors $T(\boldsymbol{v})$). We described T without a matrix: rotate the plane by 30°.

Is rotation linear? *Yes it is.* We can rotate two vectors and add the results. The sum of rotations $T(\boldsymbol{v}) + T(\boldsymbol{w})$ is the same as the rotation $T(\boldsymbol{v}+\boldsymbol{w})$ of the sum. **The whole plane is turning together, in this linear transformation.**

The rule of linearity extends to combinations of three vectors or n vectors:

$$\boxed{\begin{array}{l}\textbf{Linearity} \qquad \boldsymbol{u} = c_1\boldsymbol{v}_1 + c_2\boldsymbol{v}_2 + \cdots + c_n\boldsymbol{v}_n \quad \textbf{must transform to} \\ \qquad\qquad T(\boldsymbol{u}) = c_1T(\boldsymbol{v}_1) + c_2T(\boldsymbol{v}_2) + \cdots + c_nT(\boldsymbol{v}_n)\end{array}} \qquad (1)$$

The 2-vector rule starts the 3-vector proof: $T(c\boldsymbol{u} + d\boldsymbol{v} + e\boldsymbol{w}) = T(c\boldsymbol{u}) + T(d\boldsymbol{v} + e\boldsymbol{w})$. Then linearity applies to both of those parts, to give three parts: $cT(\boldsymbol{u}) + dT(\boldsymbol{v}) + eT(\boldsymbol{w})$.

BASIS TELLS ALL

Suppose you know $T(\boldsymbol{v})$ for all vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ in a basis
Then you know $T(\boldsymbol{u})$ for every vector $\boldsymbol{u}$ in the space.

You see the reason: Every $\boldsymbol{u}$ in the space is a combination of the basis vectors $\boldsymbol{v}_j$. Then linearity tells us that $T(\boldsymbol{u})$ **is the same combination of the outputs** $T(\boldsymbol{v}_j)$.

Geometric Transformations

- Mathematical operations that change the position, orientation, shape, or size of geometric objects (like points, lines, and shapes) within a space, such as a 2D plane or 3D space.
- These transformations are essential in computer graphics, computer vision, and physics.
- Linear algebra provides the foundation to represent these transformations using matrices and vectors, making the computation efficient and structured.

Key Types of Geometric Transformations

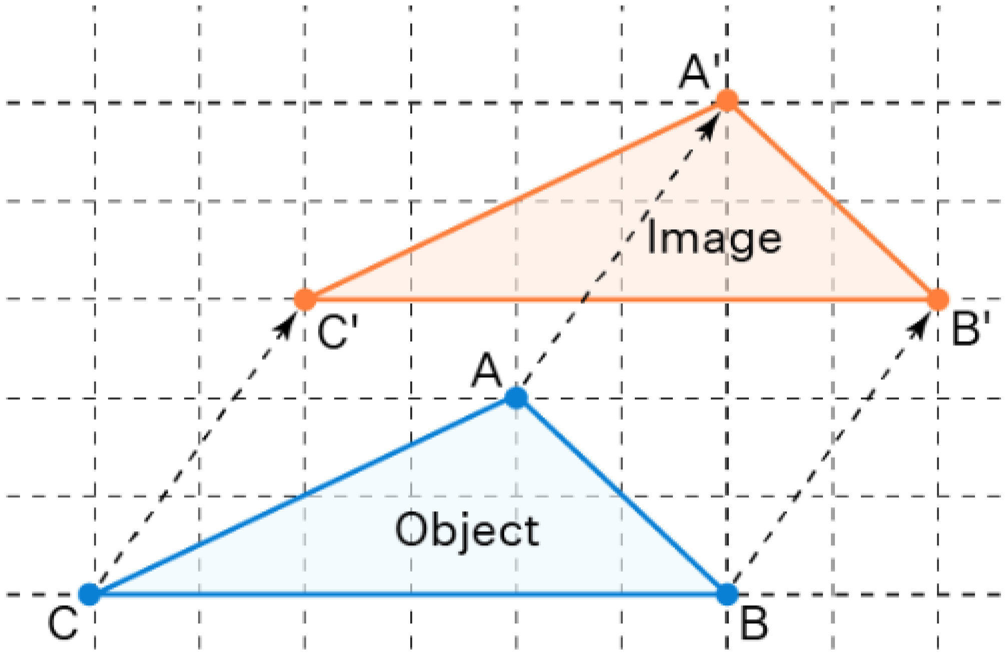
- Translation
- Scaling
- Rotation
- Reflection
- Shearing

Translation

- A translation moves every point of an object by a certain distance in a specific direction.
- Translation is not a linear transformation by itself because it does not preserve the origin, but it can be represented using matrices in homogeneous coordinates.

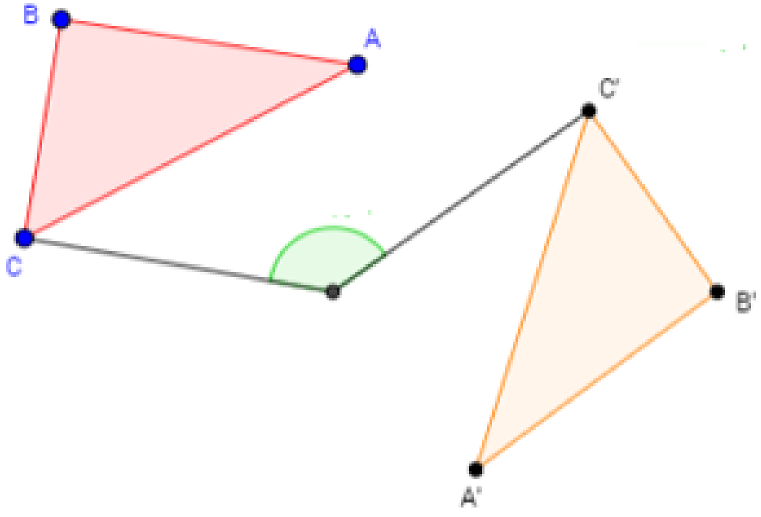
$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x + t_x \\ y + t_y \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

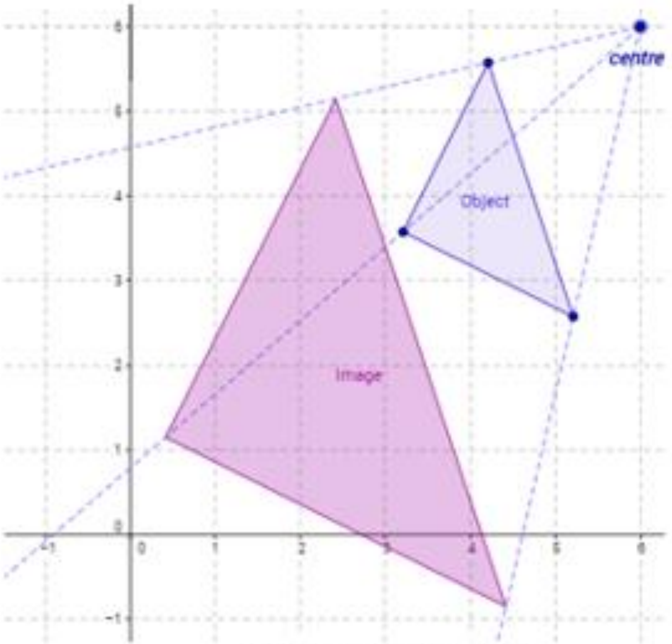


Linear Transformations

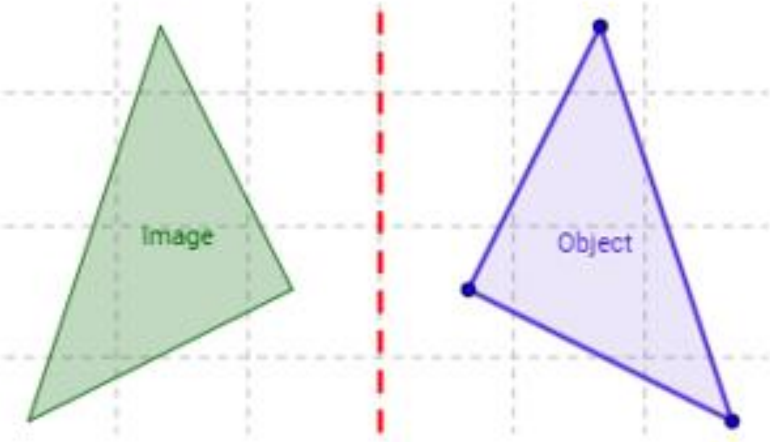
- Linear transformations are a subset of geometric transformations.
- They handle operations like Scaling, Rotation, Shearing and Reflection while keeping the origin fixed and preserving the vector space's linear structure.



Rotation



Enlargement



Reflection

Rotation

- Rotation turns an object around a point (usually the origin) by a certain angle.
- In 2D, a rotation transformation by an angle θ around the origin is represented by the matrix:

$$\text{Rotation Matrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

For a point $\mathbf{p} = (x, y)$, the new rotated point is:

$$\mathbf{p}' = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Scaling

- Scaling changes the size of an object, enlarging or shrinking it by a scale factor.
- Scaling can be represented by multiplying each coordinate by a scaling factor. In matrix form, this is a diagonal matrix

$$\text{Scaling Matrix} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix}$$

$$\mathbf{p}' = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} s_x x \\ s_y y \end{pmatrix}$$

Real life Image Scaling

- For digital images:
- Uniform scaling → zoom in/out (resize).
- Non-uniform scaling → stretch image horizontally/vertically.
- Scaling about a point → e.g., zooming about the mouse cursor in an image editor.

Reflection Transformation

- Reflection flips an object over a specified line or plane. This operation mirrors points about an axis.
- A reflection over the x-axis in 2D can be represented as:

$$\text{Reflection over x-axis} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

- A reflection over the y-axis can be represented as:

$$\text{Reflection over y-axis} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$$

Shear Transformation

- A shear transformation slants the shape of an object.
- In 2D, it skews the shape in the x or y direction by shifting coordinates along one axis by an amount proportional to their position on the other axis.
- The matrix for a shear along the x-axis is:

$$\text{Shearing Matrix (x-axis)} = \begin{pmatrix} 1 & k \\ 0 & 1 \end{pmatrix}$$

- The matrix for a shear along the y-axis:

$$\text{Shearing Matrix (y-axis)} = \begin{pmatrix} 1 & 0 \\ k & 1 \end{pmatrix}$$

3D - Rotation

Rotation Around the x-axis:

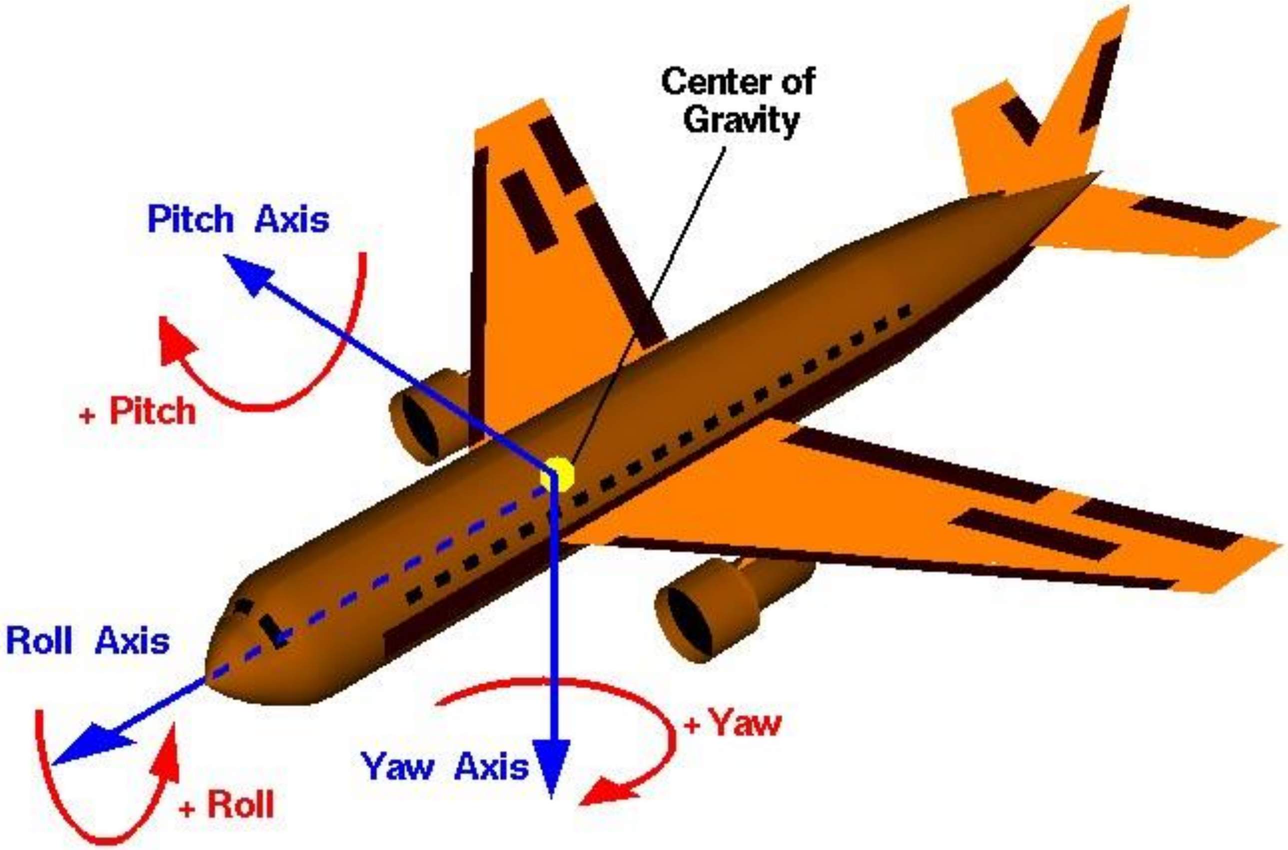
$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix}$$

Rotation Around the y-axis:

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

Rotation Around the z-axis:

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



3D - Reflection

Reflection Across the xy-plane:

This flips the z-coordinates.

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

Reflection of the point $(1, 2, 3)$ across the xy-plane gives $(1, 2, -3)$.

3D - Reflection

Reflection Across the yz-plane:

This flips the x-coordinates.

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

Reflection of the point $(1, 2, 3)$ across the yz-plane gives $(-1, 2, 3)$.

3D - Reflection

Reflection Across the zx-plane:

This flips the y-coordinates.

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

Reflection of the point $(1, 2, 3)$ across the zx-plane gives $(1, -2, 3)$.

3D - Scaling

Scaling Matrix:

$$S = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{pmatrix}$$

Applications

Applications

Applications