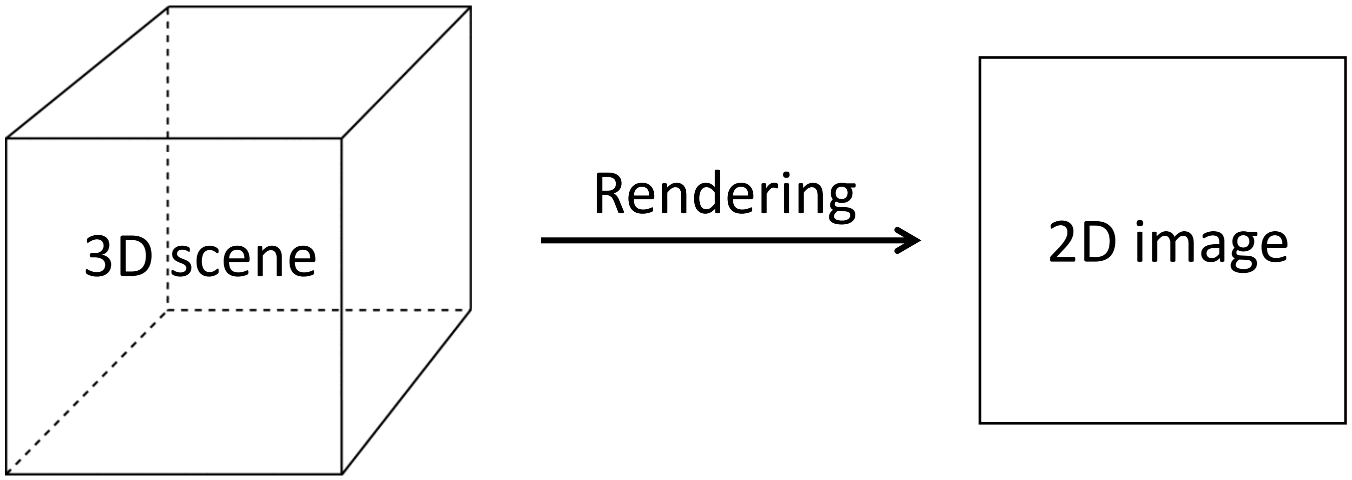
Modelowanie Wirtualnych Swiatow 3D

Dr Wojciech Pałubicki

3D Computer graphics in a nutshell

- Images generated with computers
- In this course we focus on 3D rendering



Films, special effects

Rendering specifications

- Fundamental specifications for rendering are **OpenGL** and **Direct3D**
- **Game engines** use rendering specifications (e.g. Unity)

What will you learn

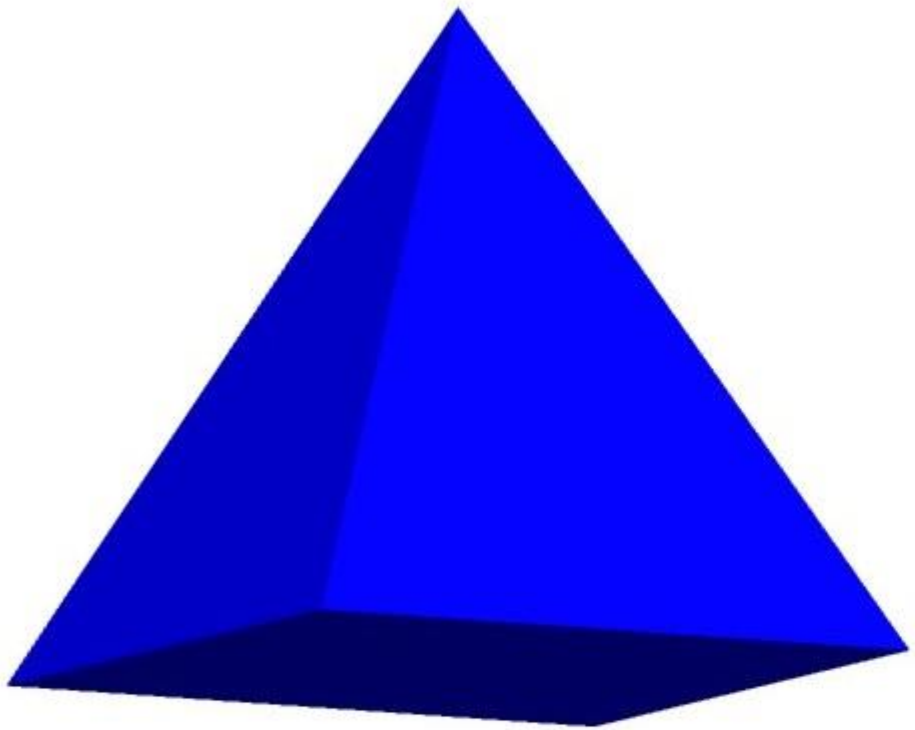
- Rendering pipeline (how to generate 2D images from 3D scenes)
- 3D Modeling
- L-Systems
- **Unity**
- Experience with **C#**
- Fundamental elements of **Shaders**, a program executed on the graphics card

Passing Computer graphics

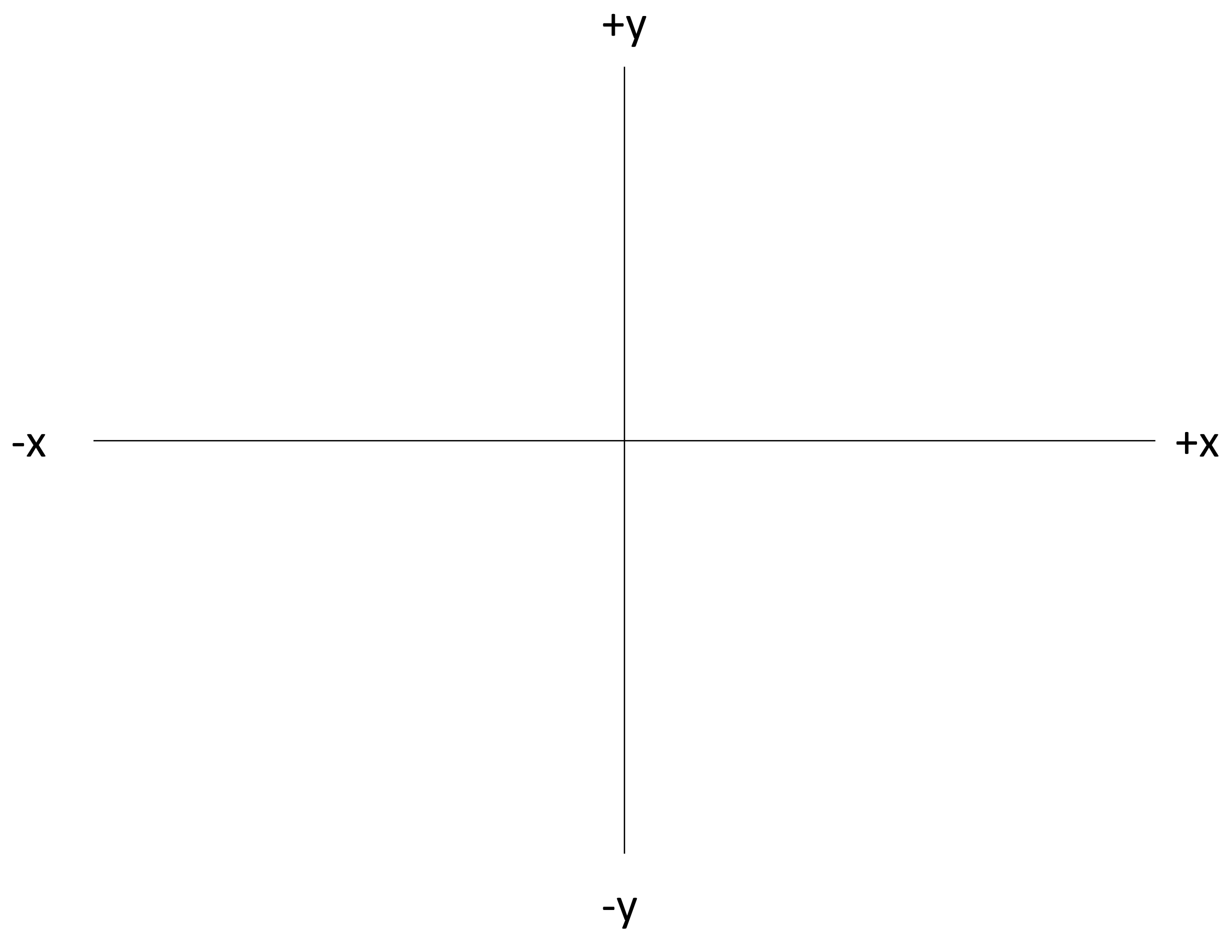
- Home work (30%)
- Semester project (70%)
- More details in the labs
- Information will be available: MS Teams and https://wp.faculty.wmi.amu.edu.pl/MWS.html

Semester Project

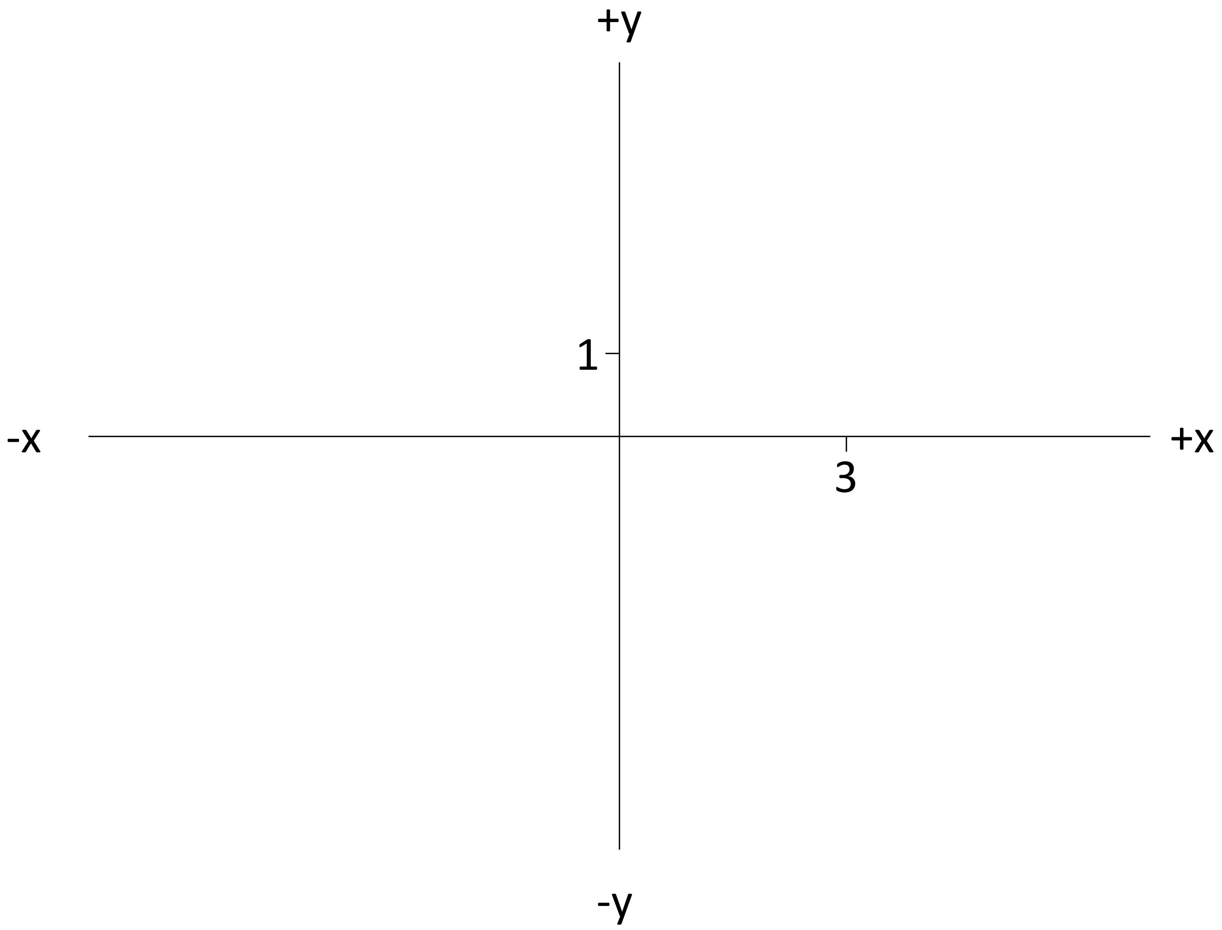
How to express 3D objects mathematically?



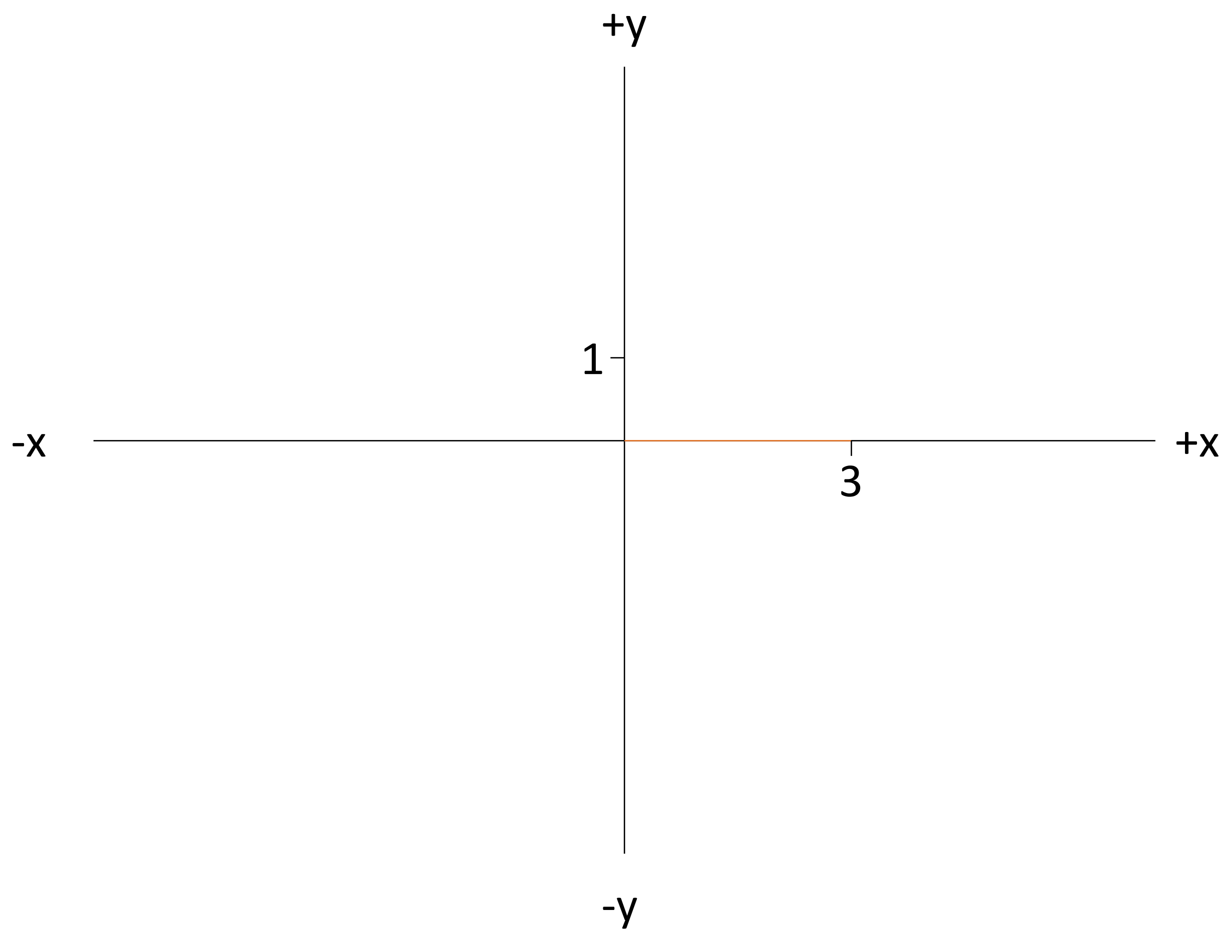
2D coordinate system



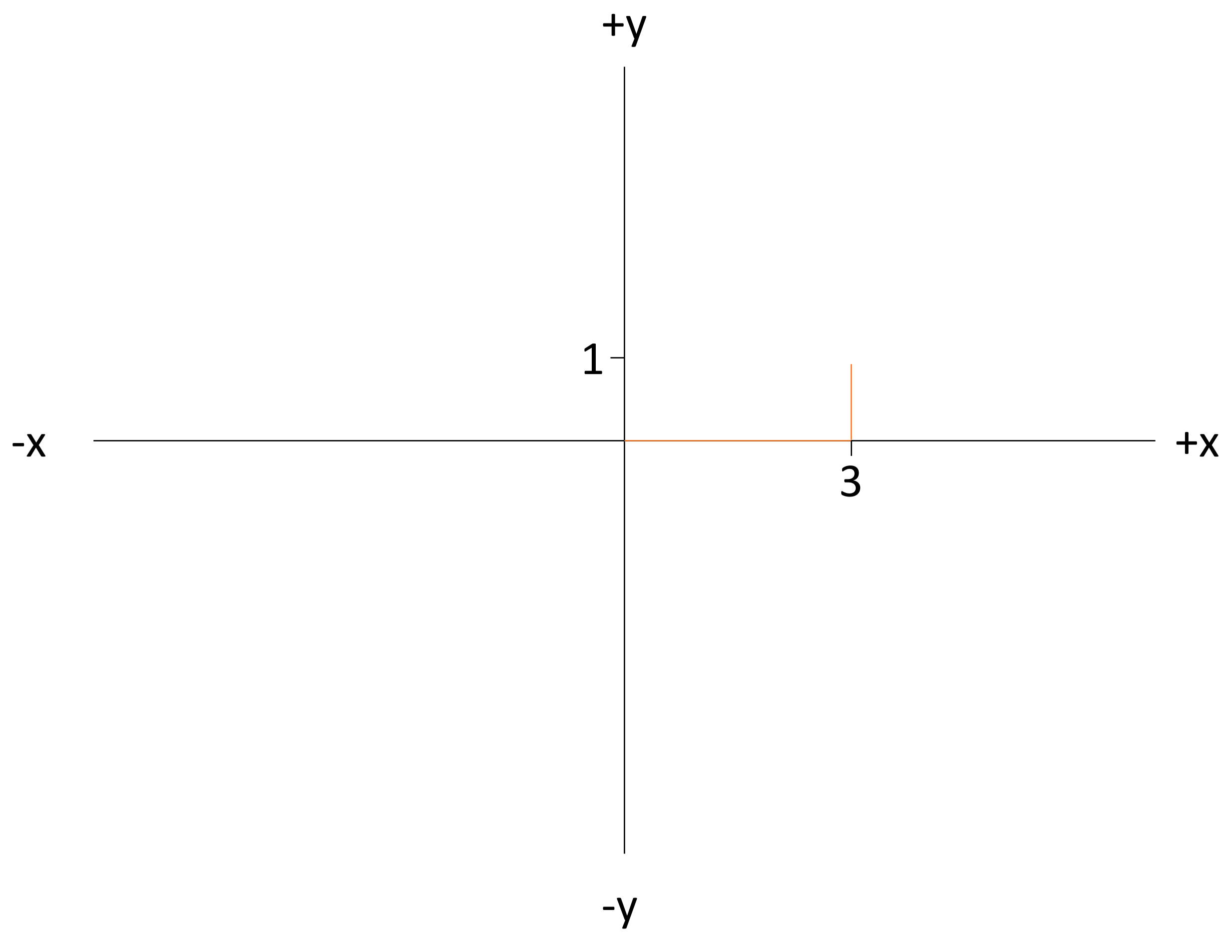
2D coordinate system



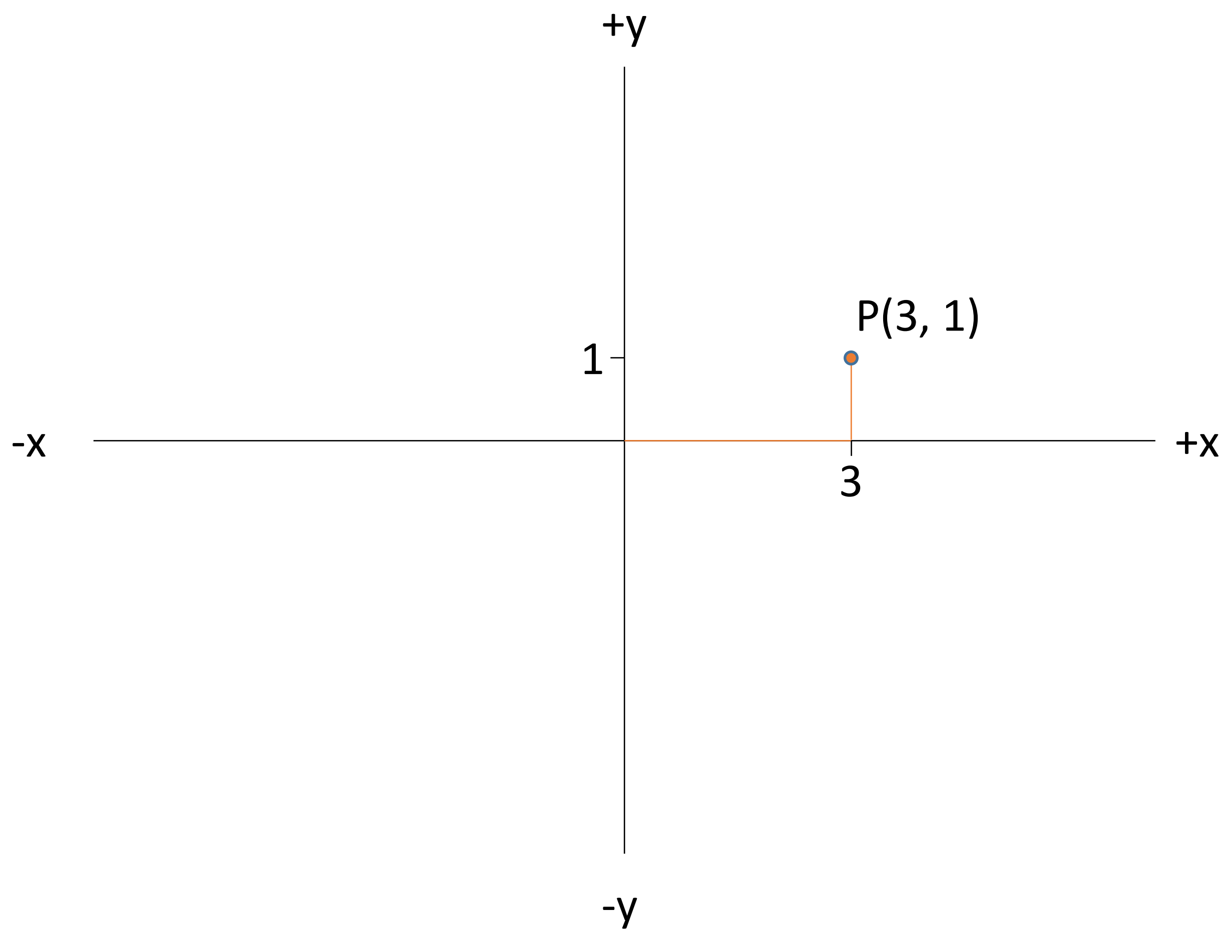
2D coordinate system



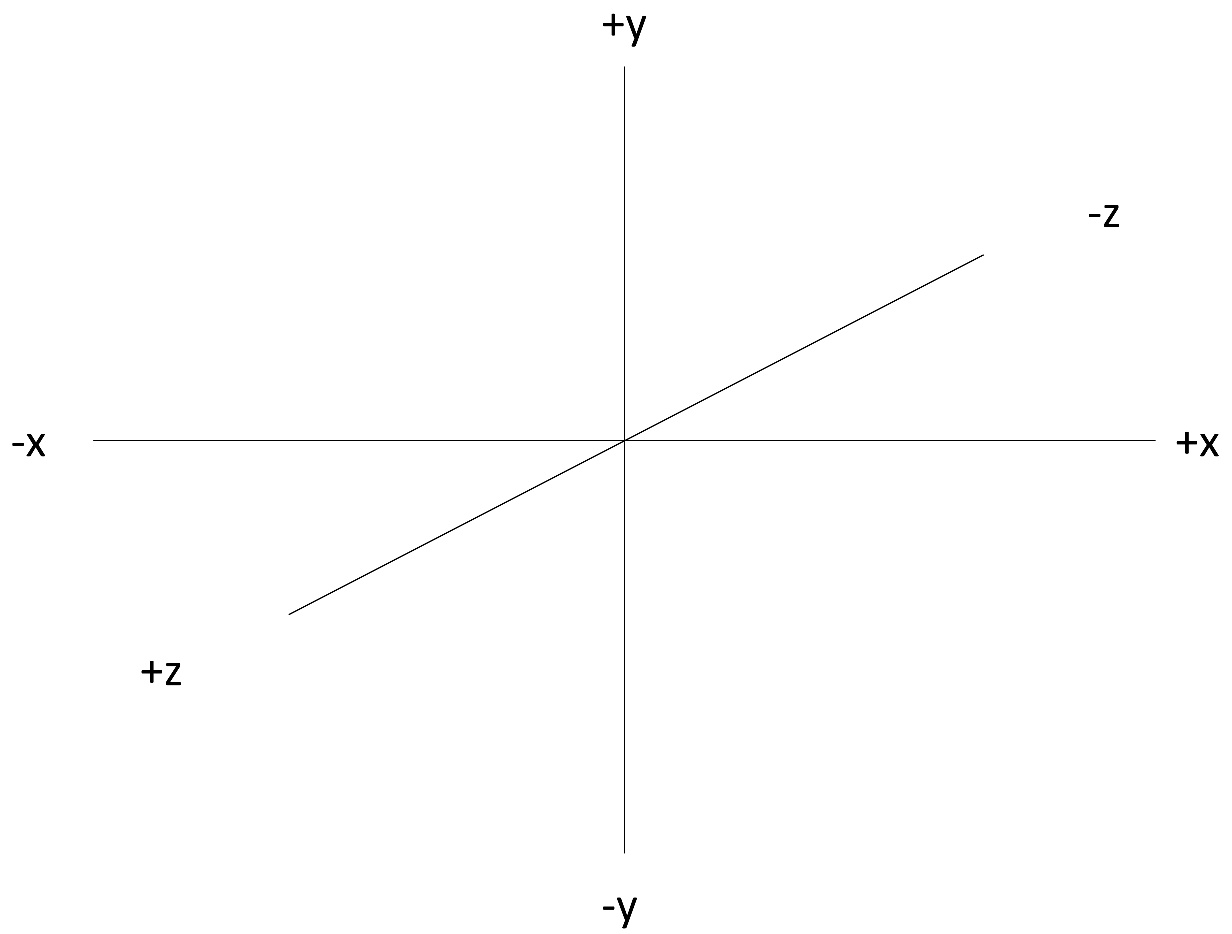
2D coordinate system



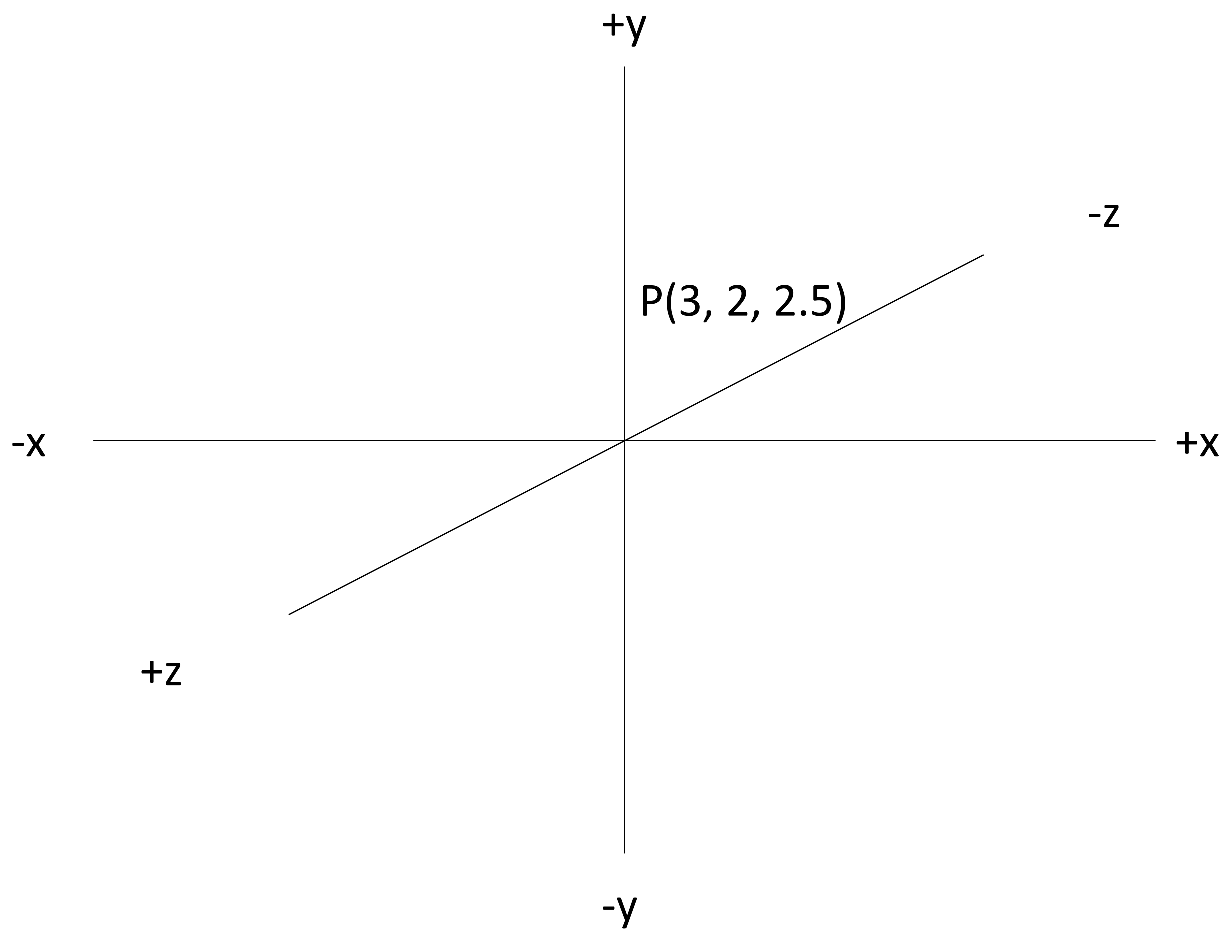
2D coordinate system



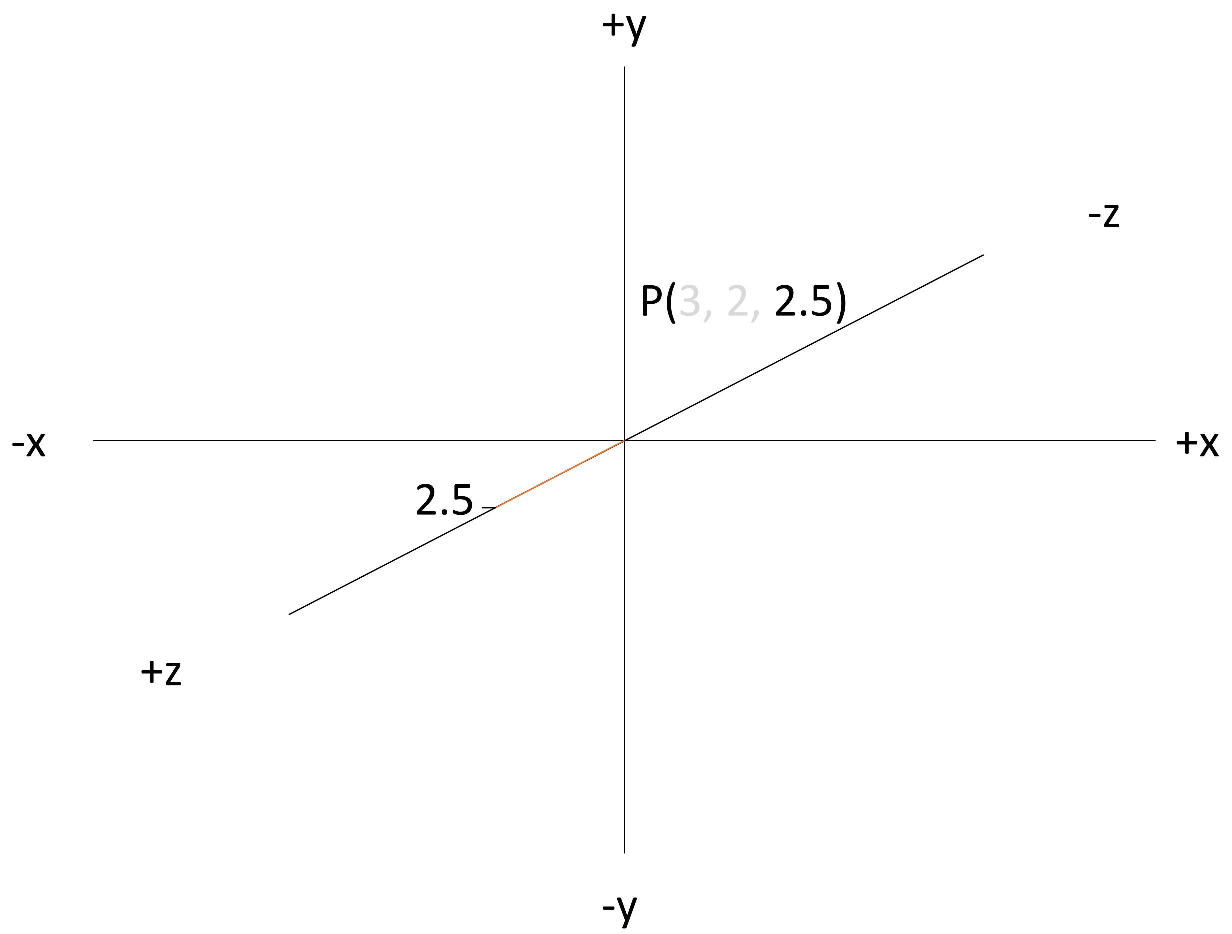
3D coordinate system



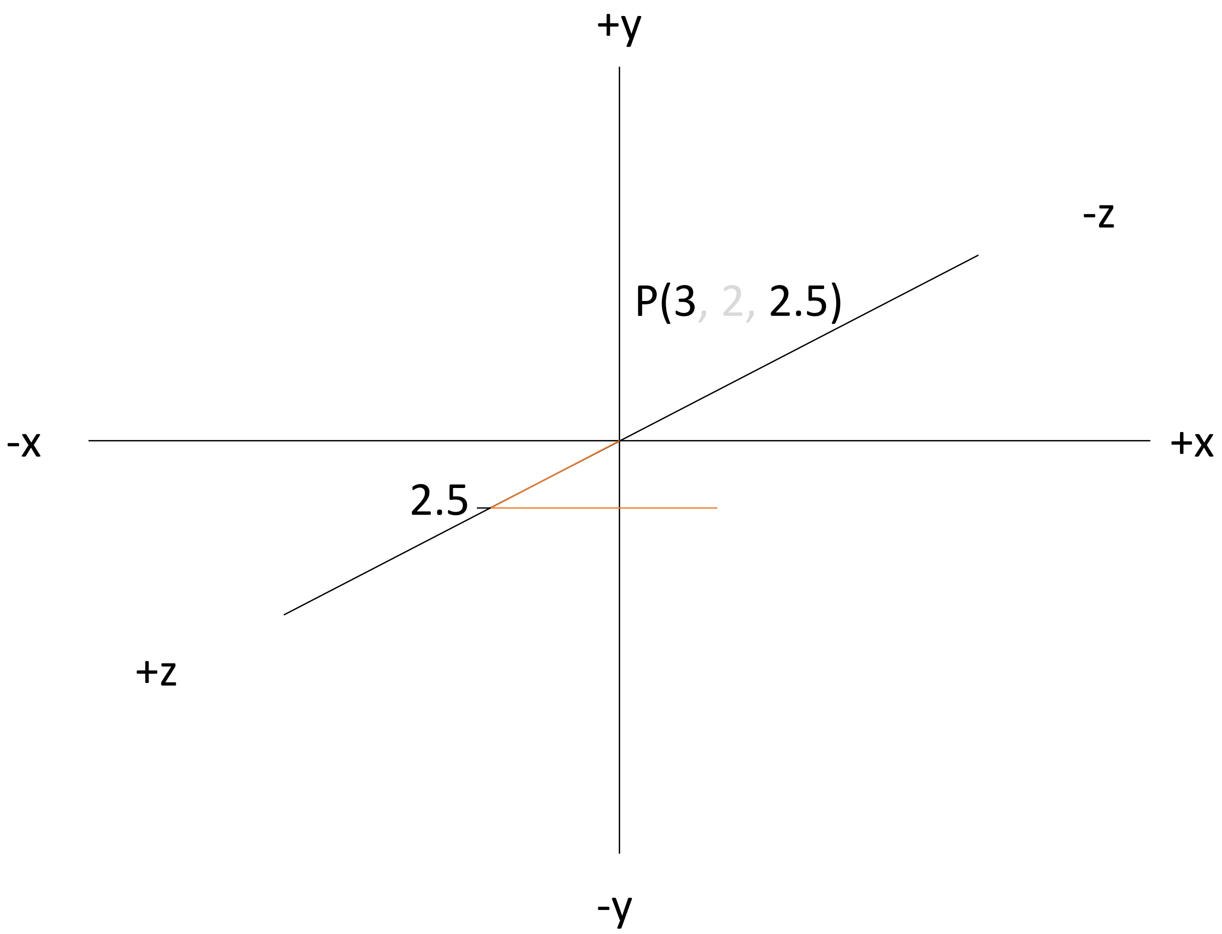
3D coordinate system



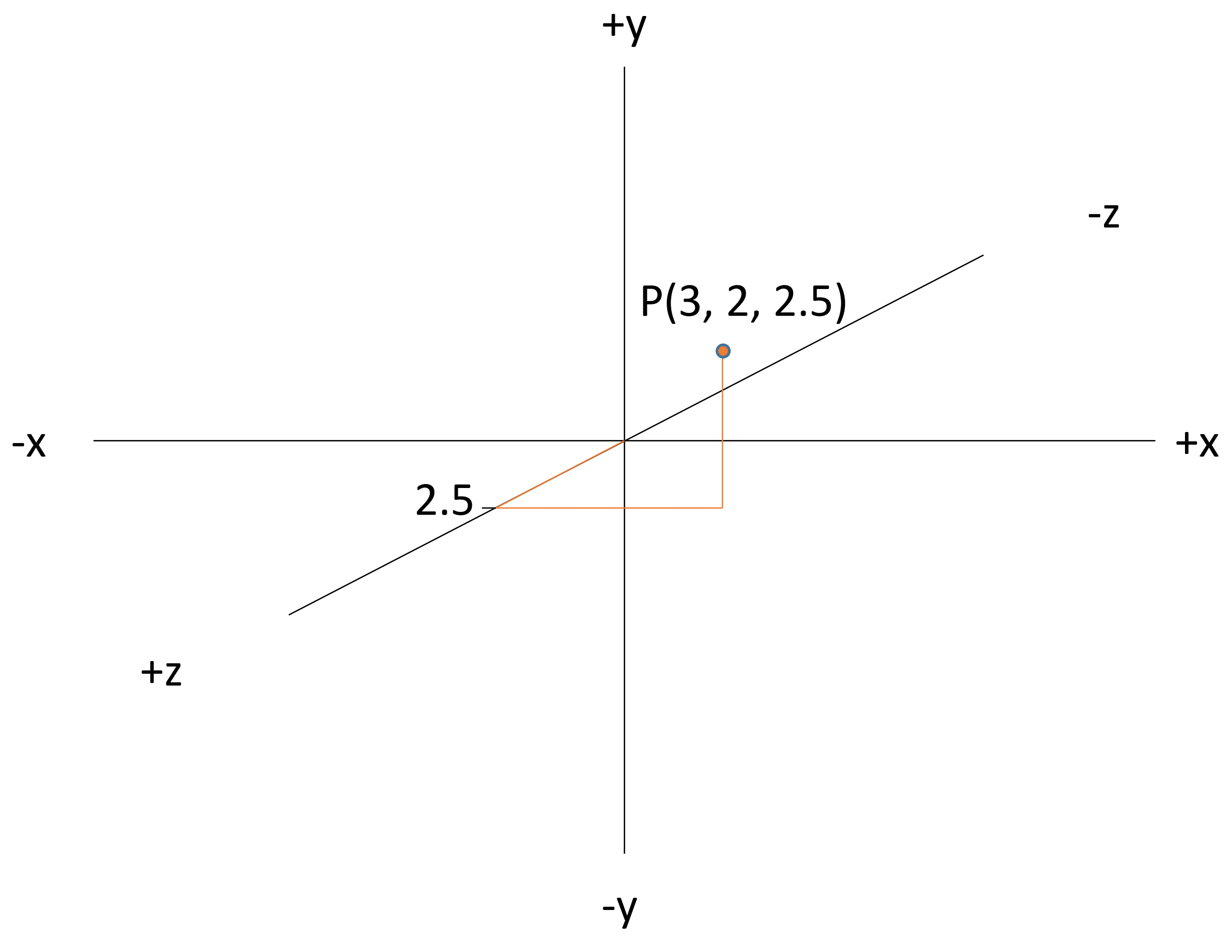
3D coordinate system



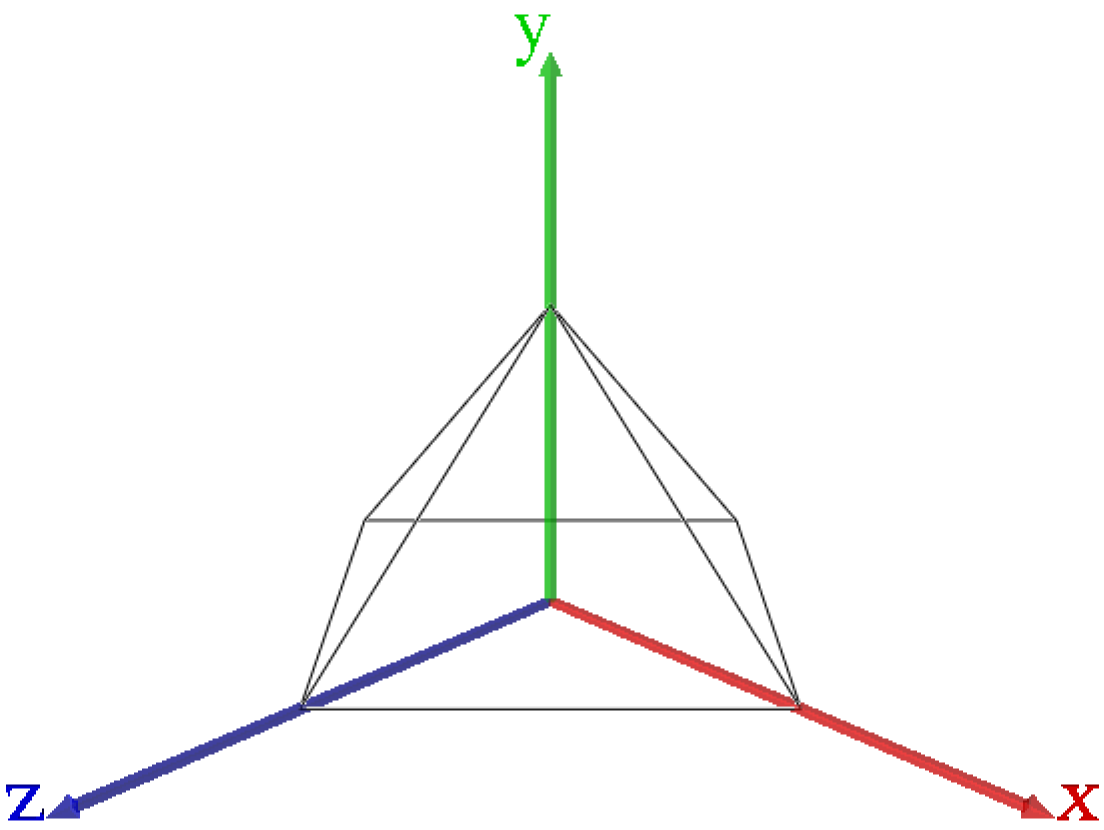
3D coordinate system



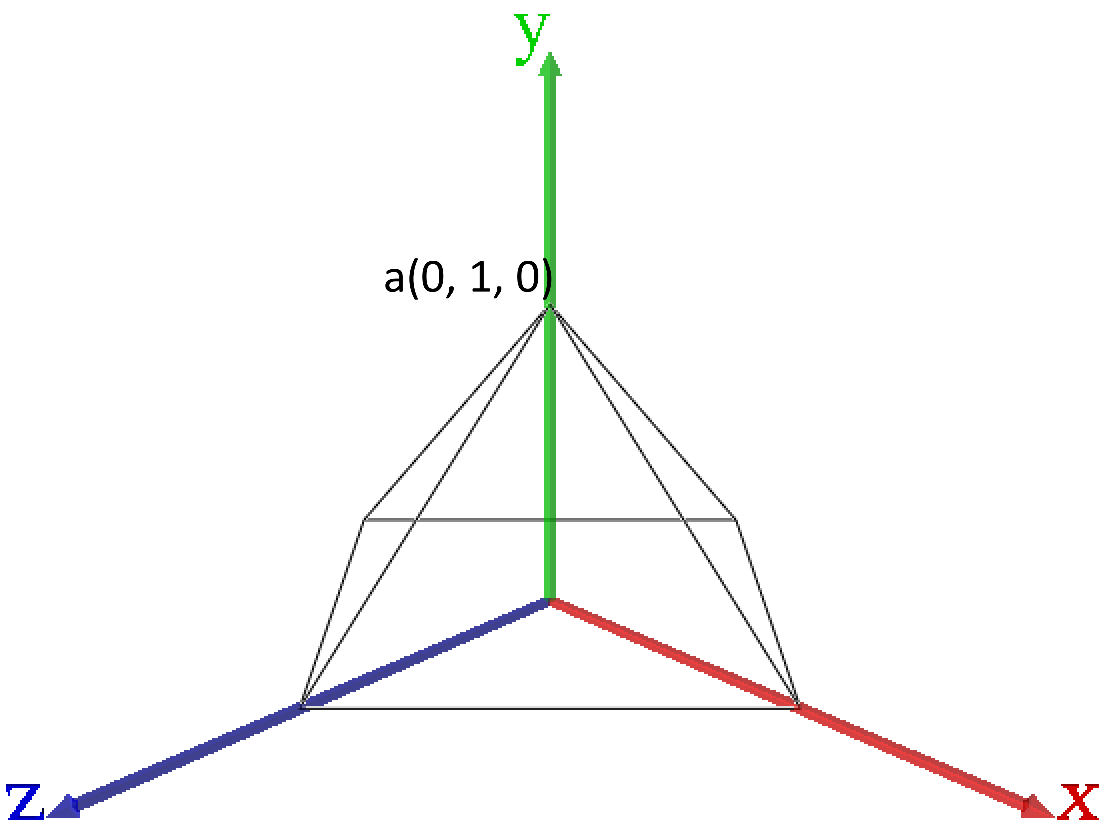
3D coordinate system



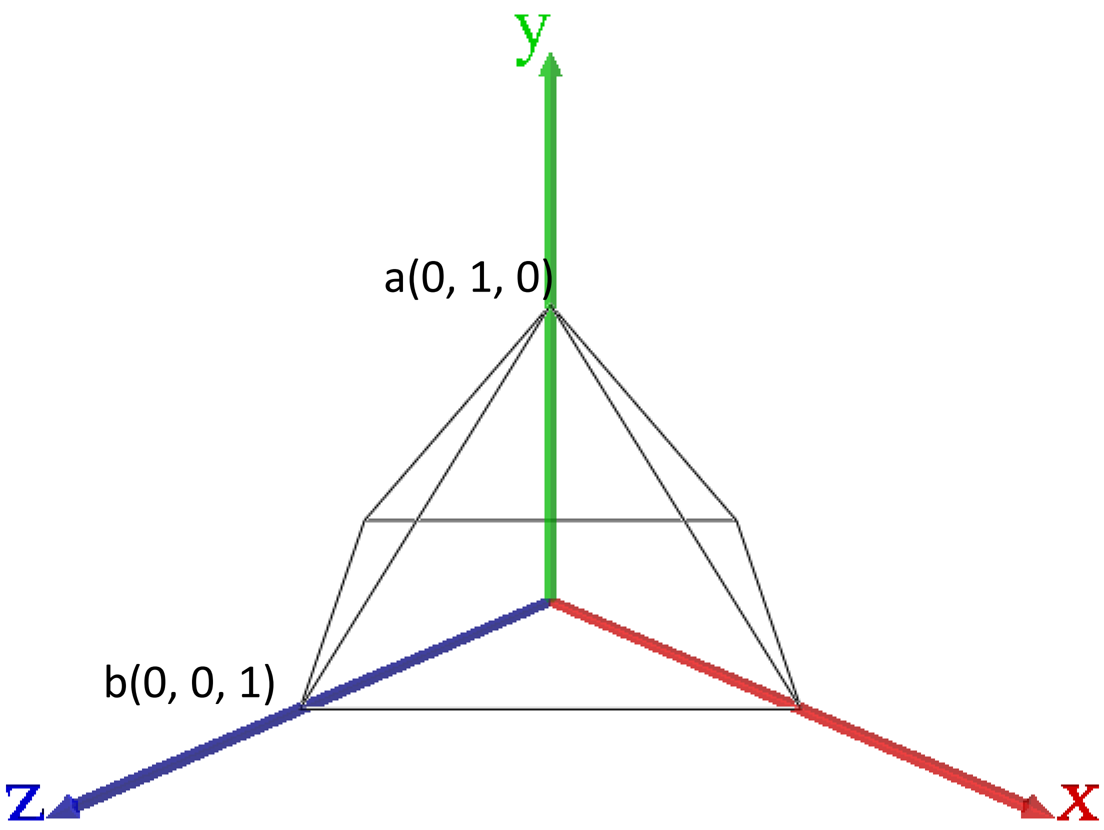
Example: Pyramid



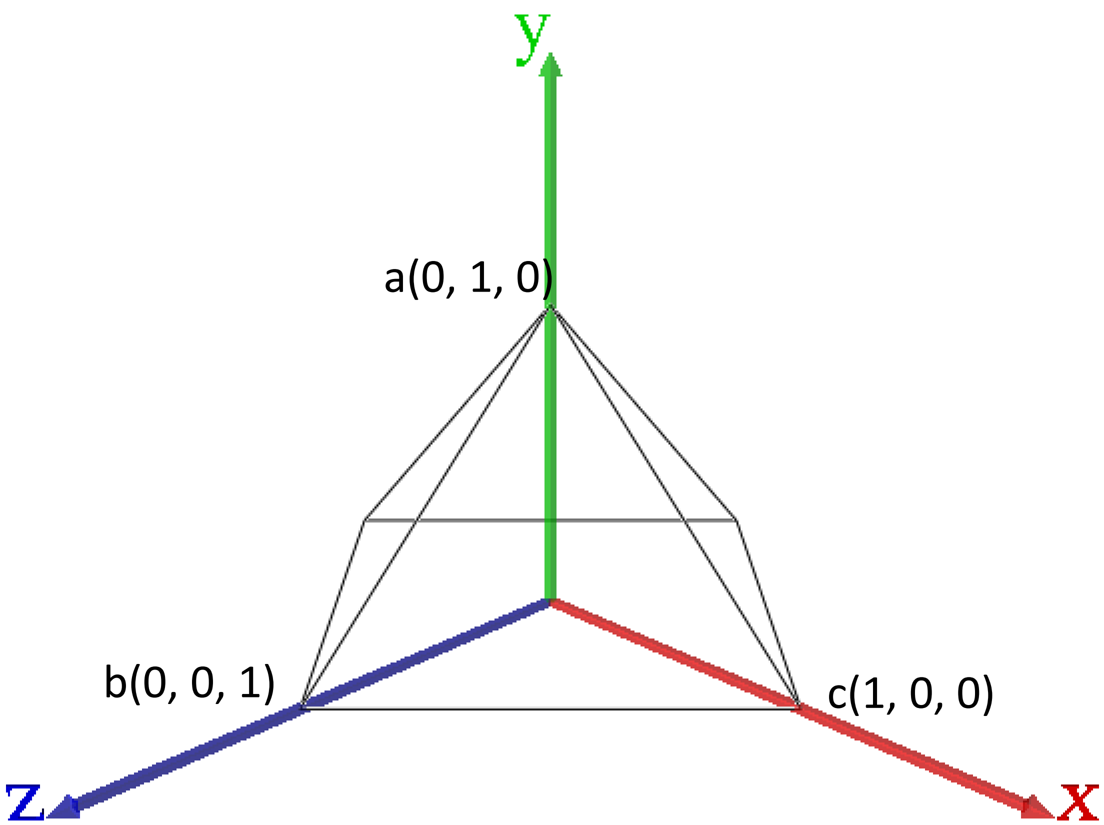
Example: Pyramid



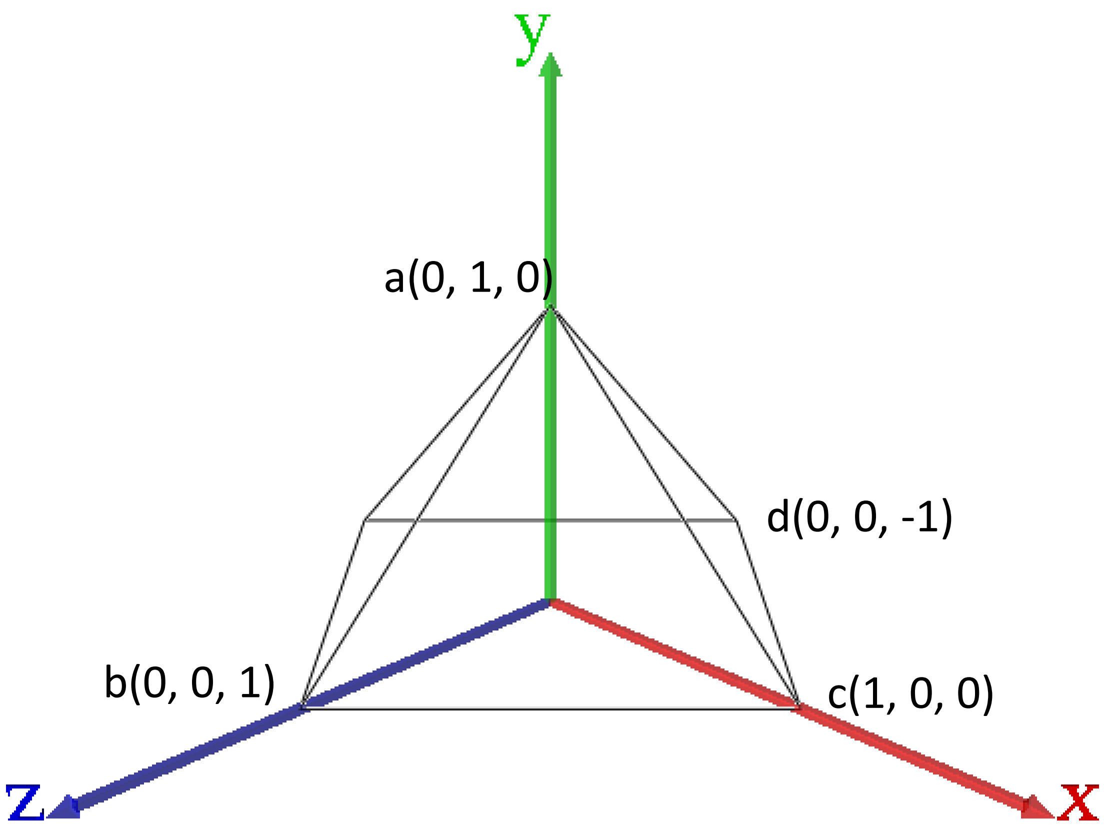
Example: Pyramid



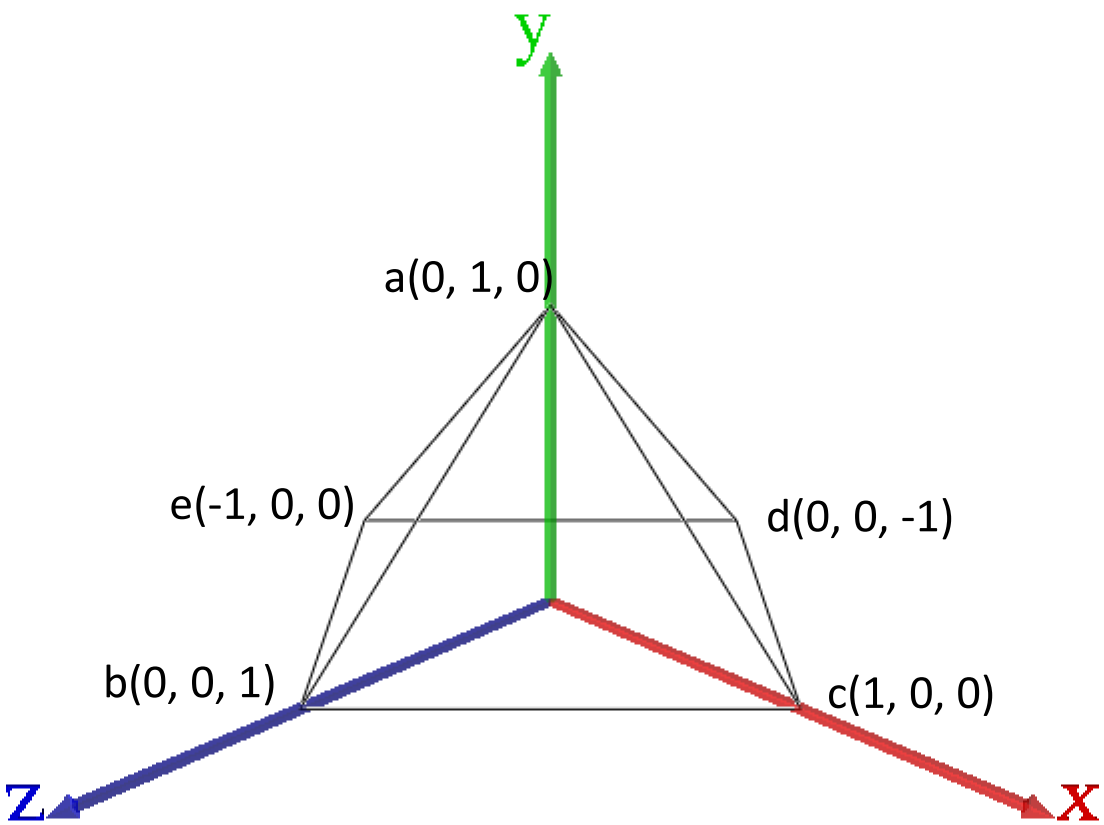
Example: Pyramid



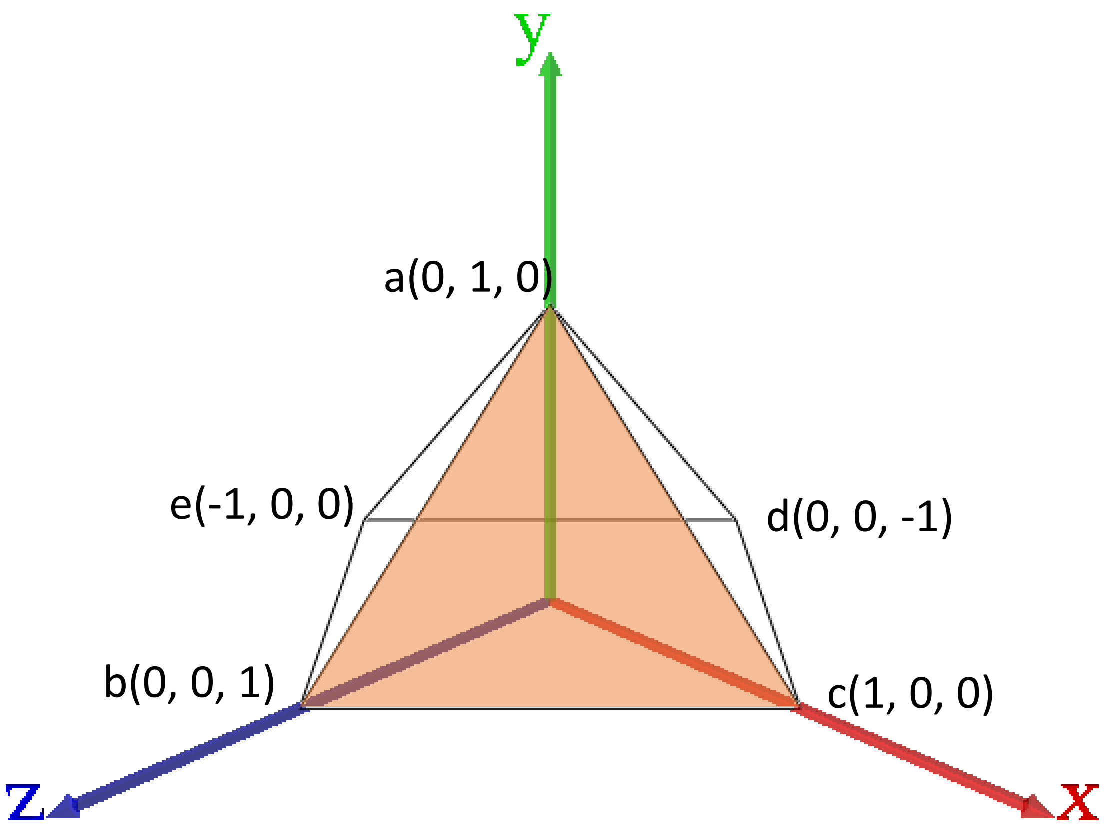
Example: Pyramid



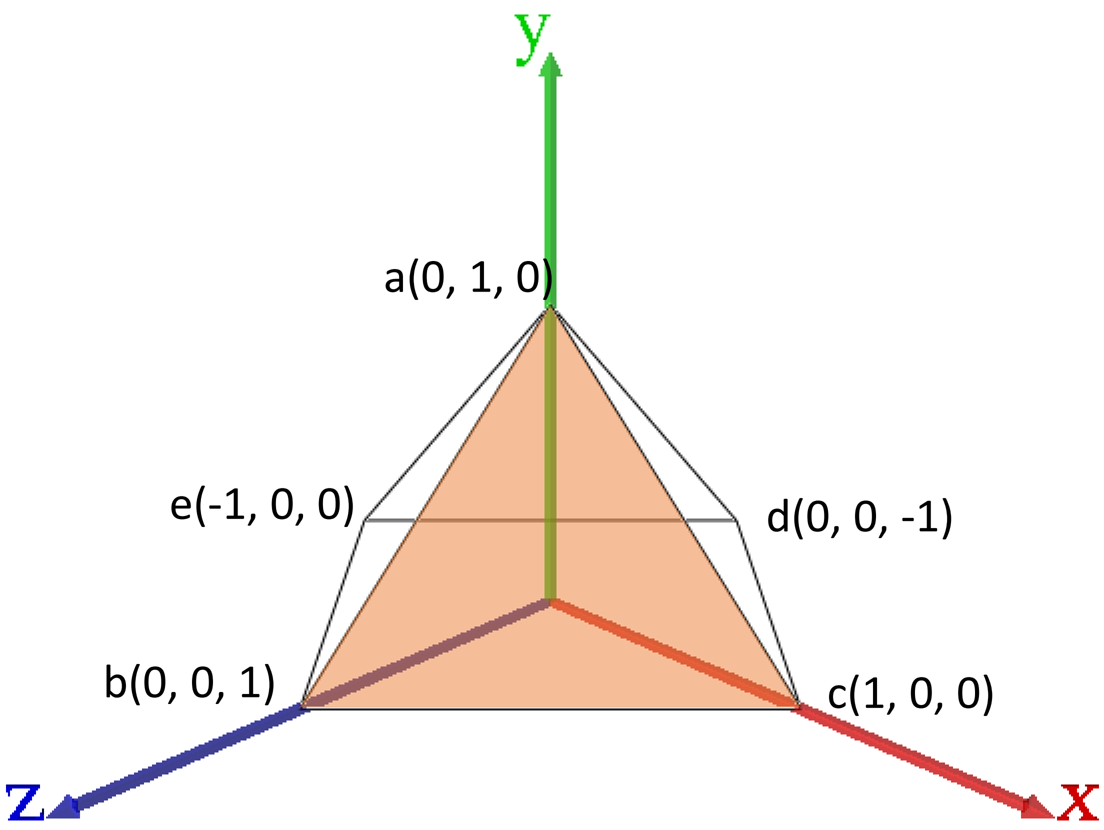
Example: Pyramid



Face

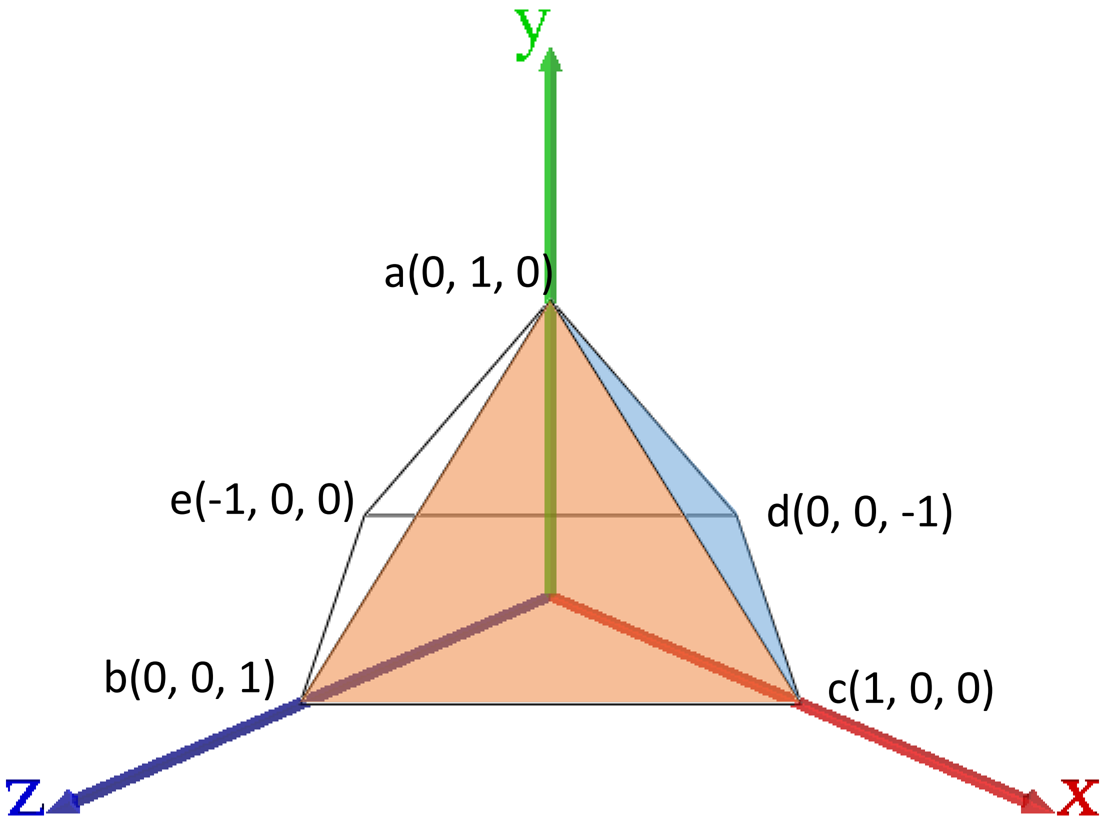


Face



For example: (0, 1, 0) (0, 0, 1) (1, 0, 0)

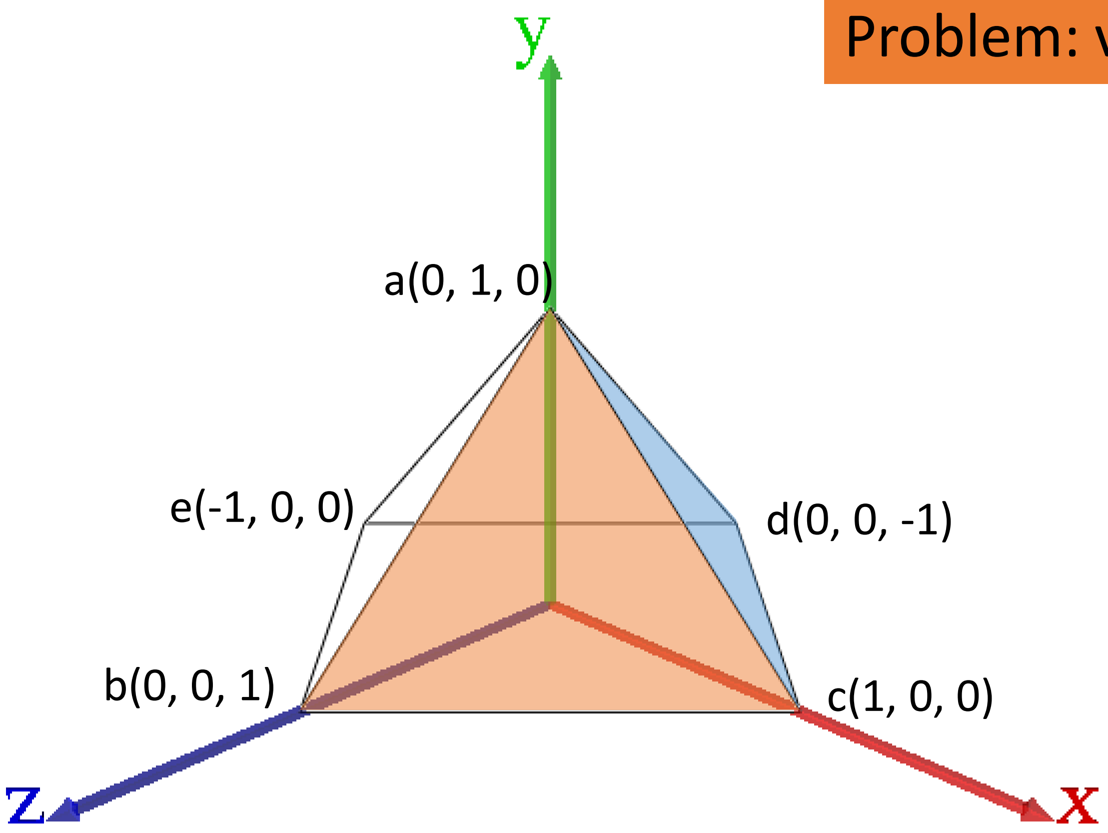
Face



For example: (0, 1, 0) (0, 0, 1) (1, 0, 0) or (0, 1, 0) (1, 0, 0) (0, 0, -1)

Face

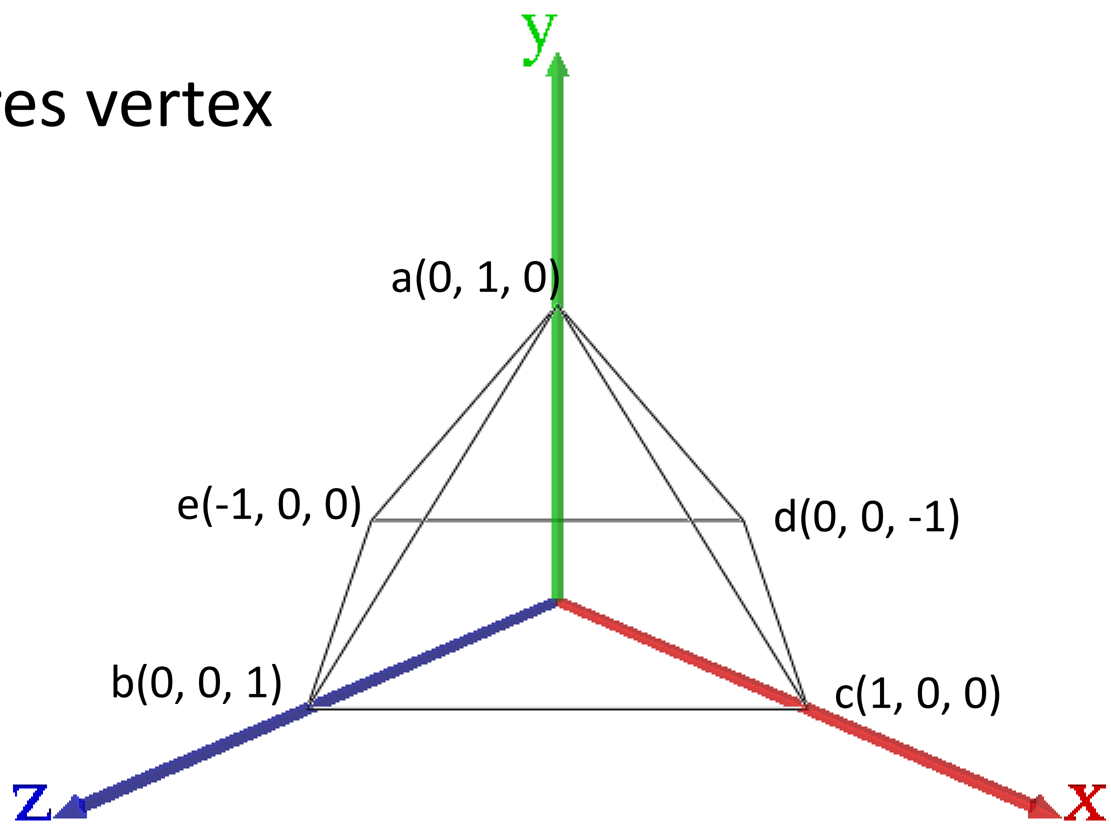
Problem: vertex duplication!



For example: (0, 1, 0) (0, 0, 1) (1, 0, 0) or (0, 1, 0) (1, 0, 0) (0, 0, -1)

Solution: new data structure

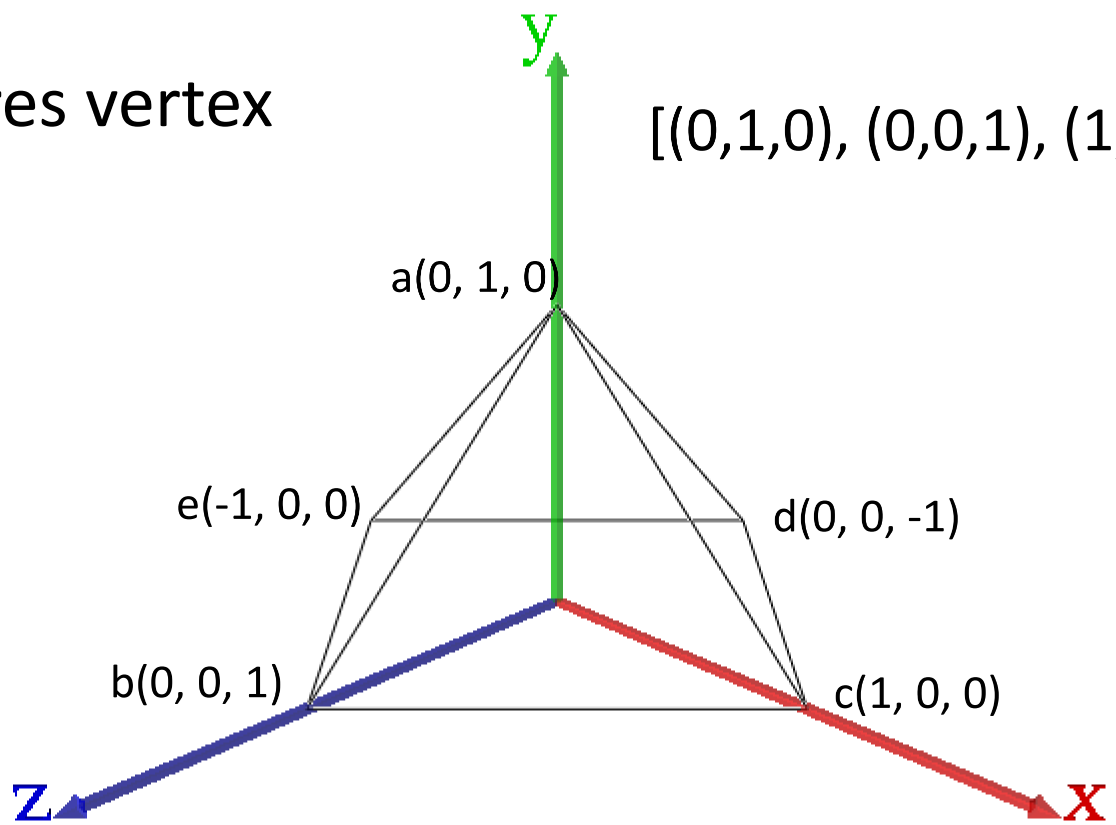
Vertex buffer stores vertex information



Solution: new data structure

Vertex buffer stores vertex information

[(0,1,0), (0,0,1), (1,0,0), (0,0,-1), (-1,0,0)]

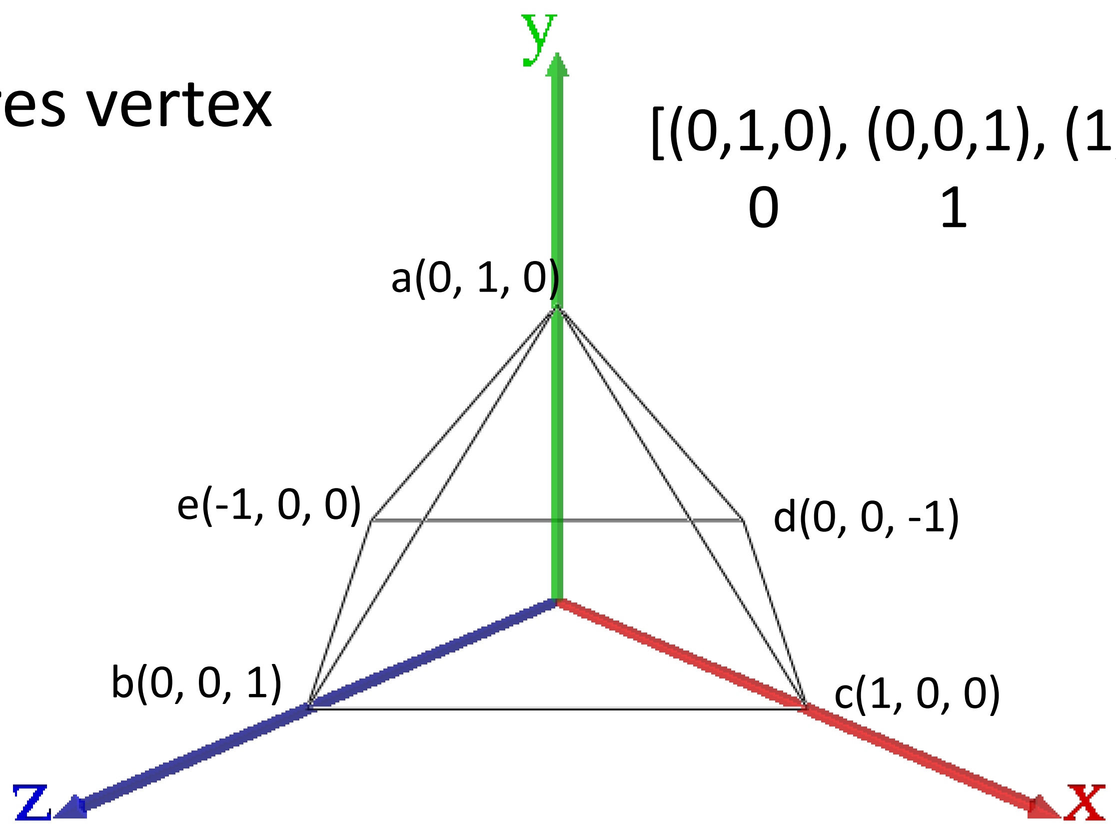


Solution: new data structure

Vertex buffer stores vertex information

[(0,1,0), (0,0,1), (1,0,0), (0,0,-1), (-1,0,0)]

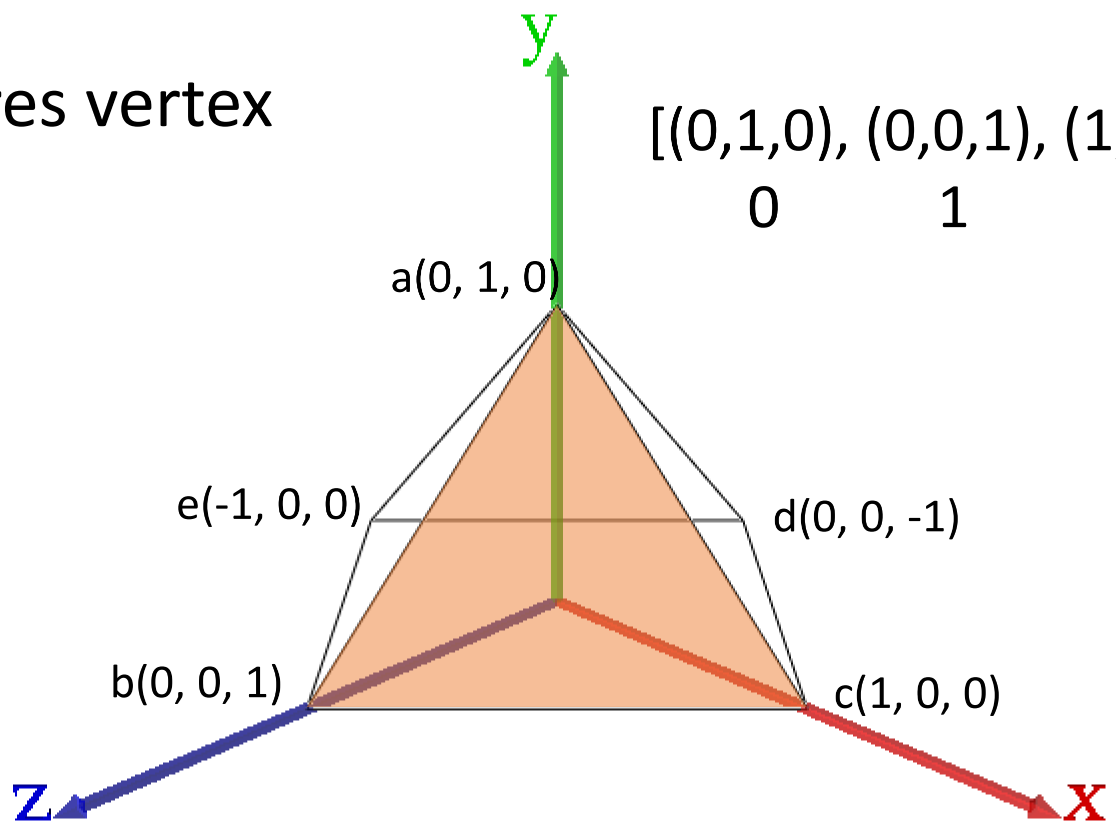
0 1 2 3 4



Solution: new data structure

Vertex buffer stores vertex information

[(0,1,0), (0,0,1), (1,0,0), (0,0,-1), (-1,0,0)]
0 1 2 3 4

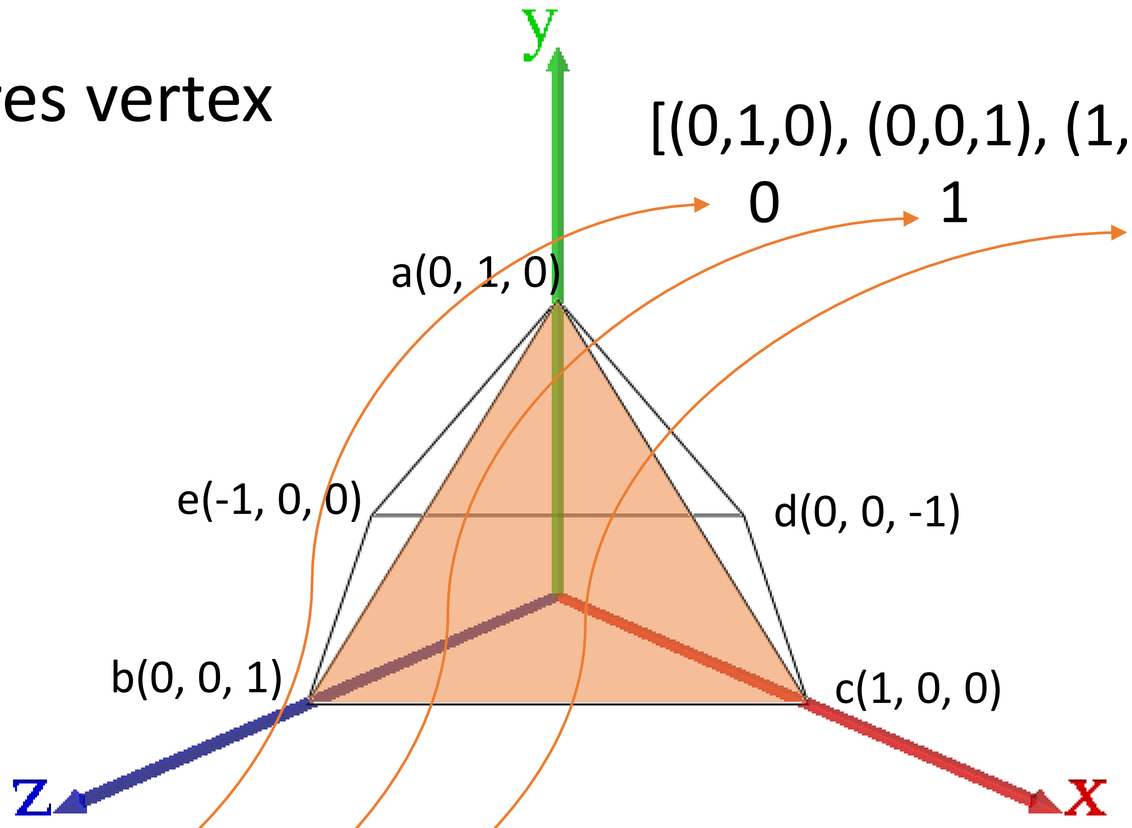


For example: (0, 1, 2)

Solution: new data structure

Vertex buffer stores vertex information

[(0,1,0), (0,0,1), (1,0,0), (0,0,-1), (-1,0,0)]
0 1 2 3 4

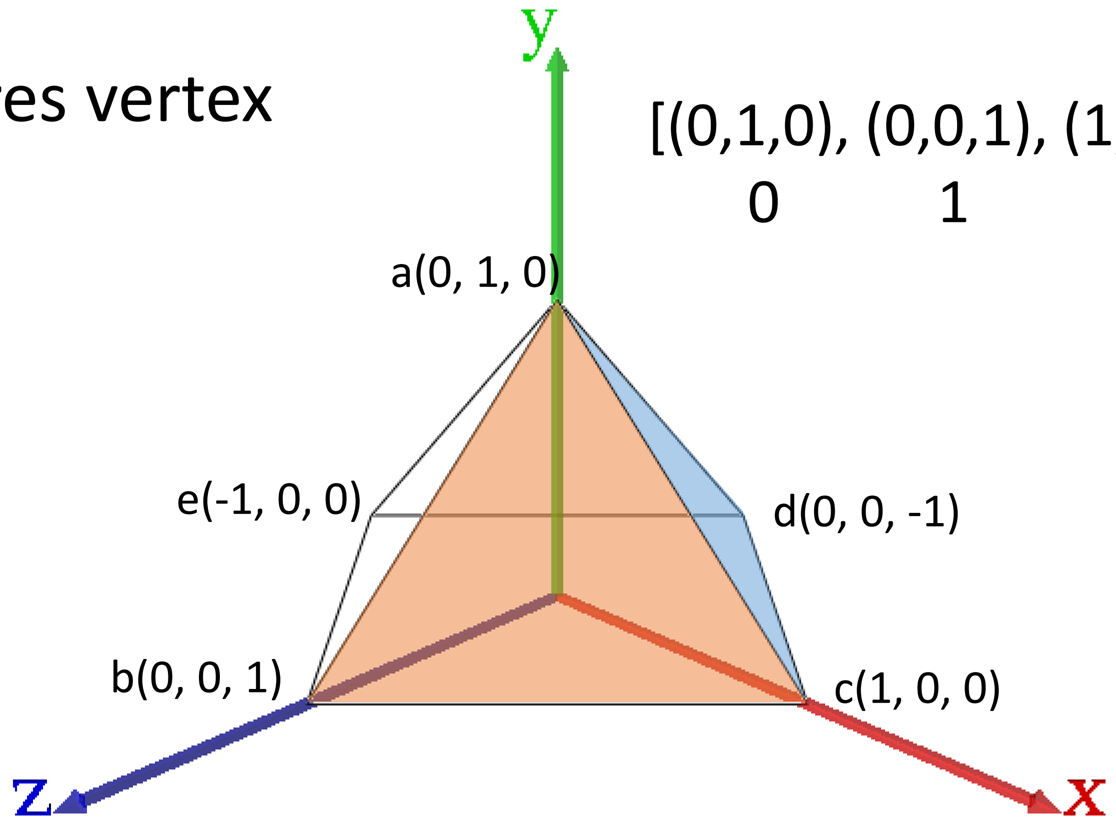


For example: (0, 1, 2)

Solution: new data structure

Vertex buffer stores vertex information

[(0,1,0), (0,0,1), (1,0,0), (0,0,-1), (-1,0,0)]
0 1 2 3 4

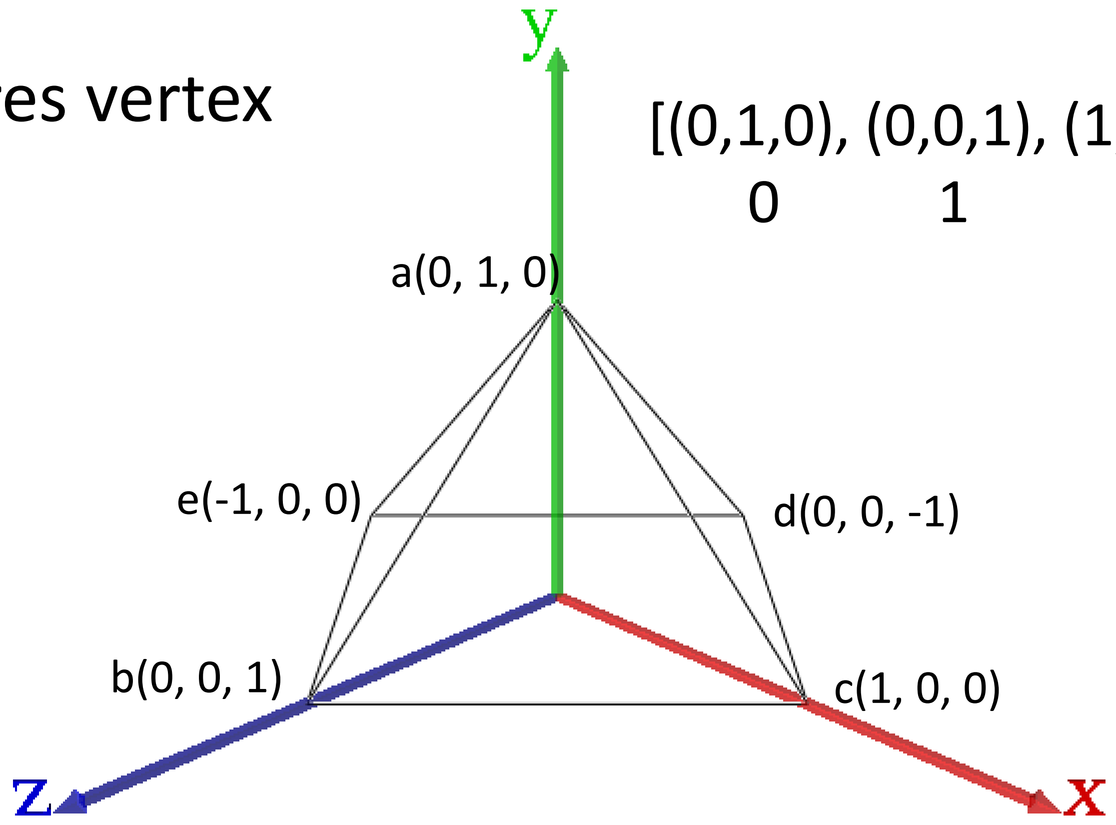


For example: (0, 1, 2) or(0, 2, 3)

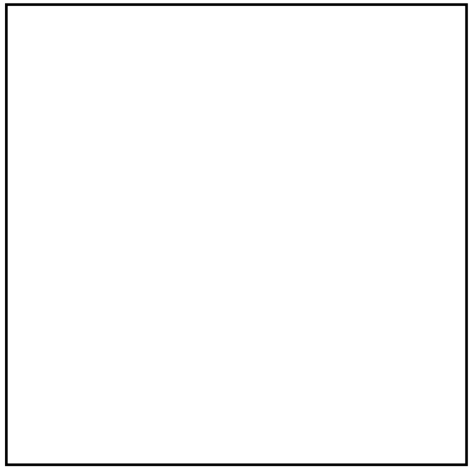
Solution: new data structure

Vertex buffer stores vertex information

[(0,1,0), (0,0,1), (1,0,0), (0,0,-1), (-1,0,0)]
0 1 2 3 4

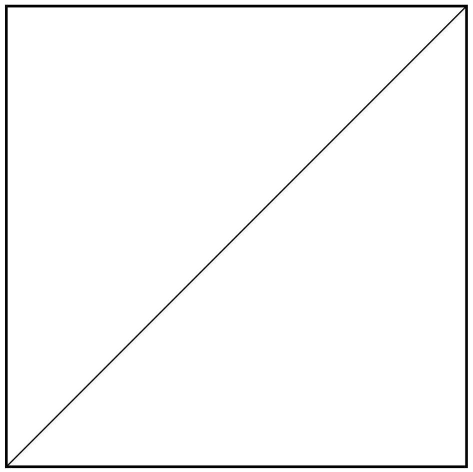


Triangles



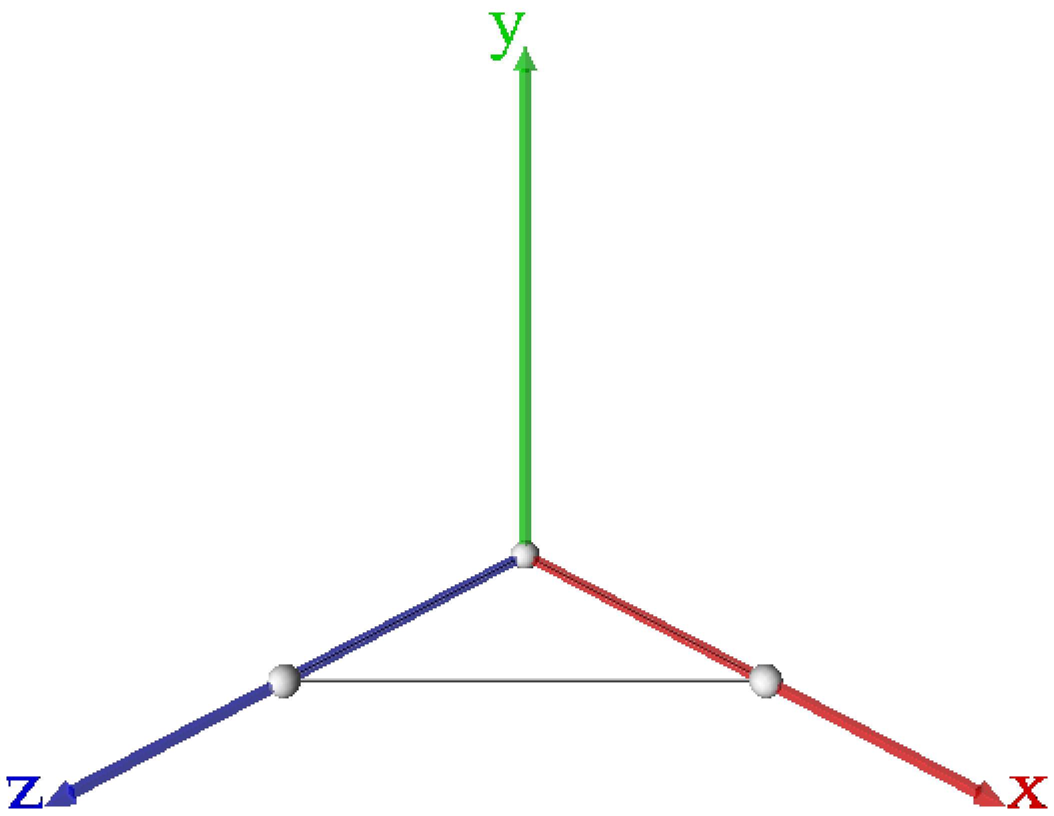
bottom of the pyramid

Triangles

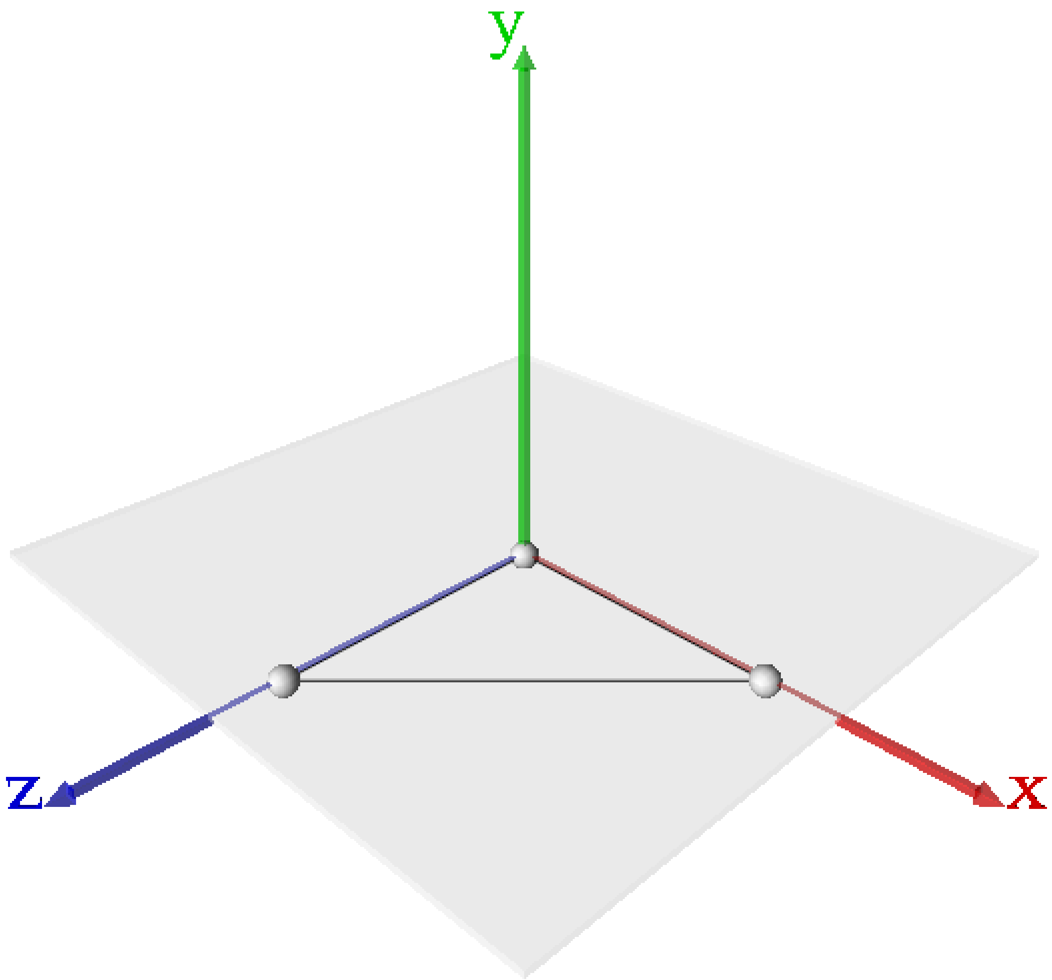


bottom of the pyramid

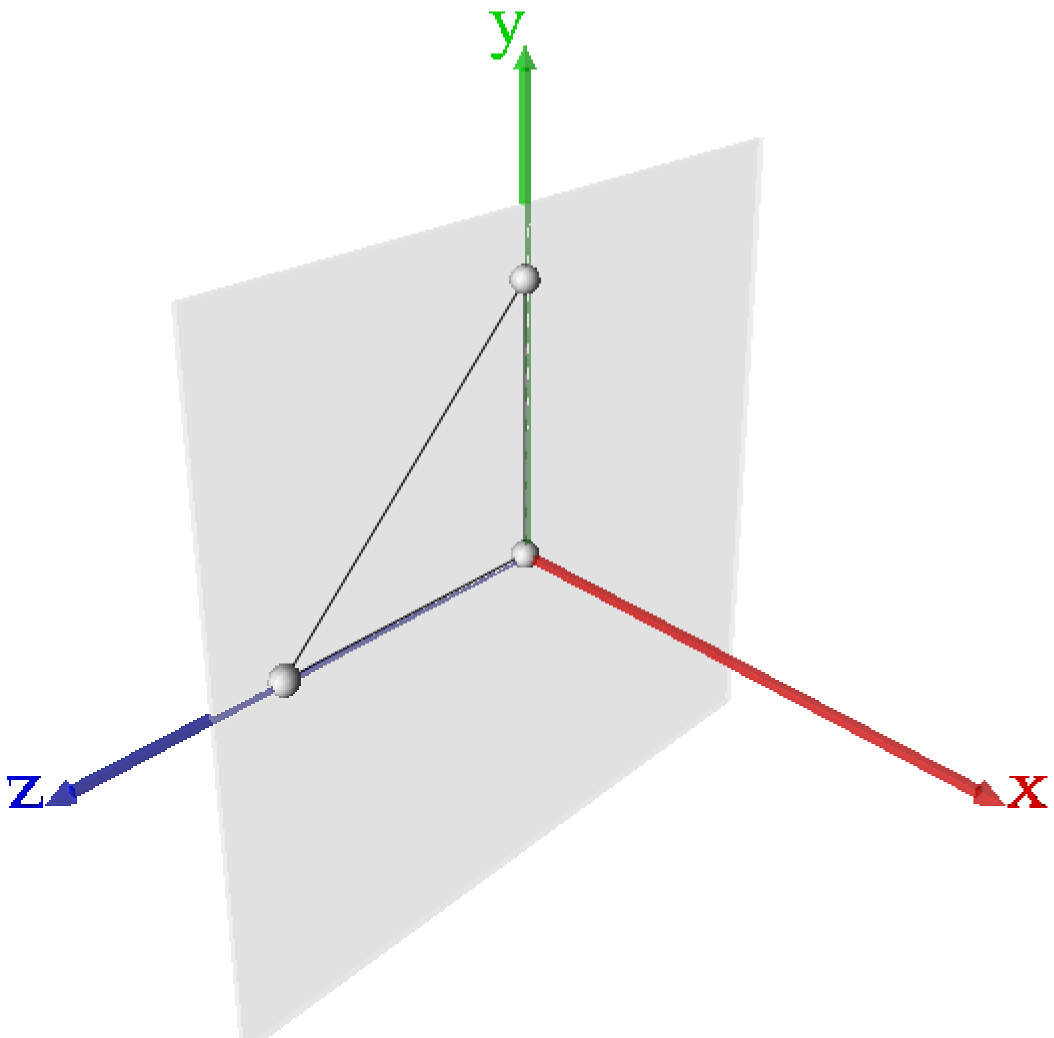
Triangles - coplanarity



Triangles - coplanarity

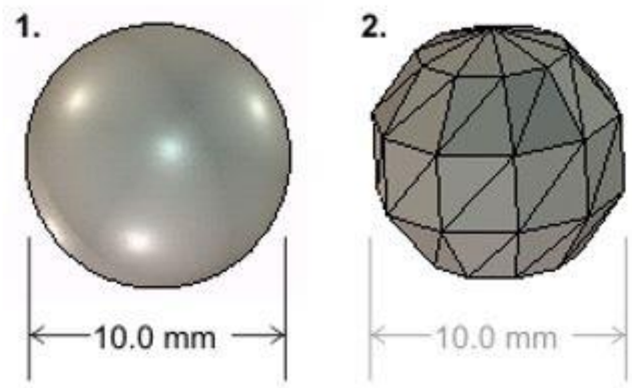


Triangles - coplanarity

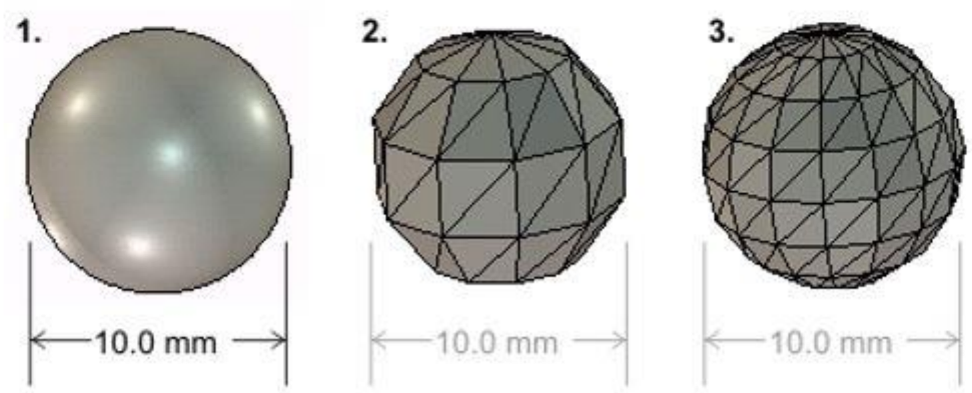


Shape approximation with triangles

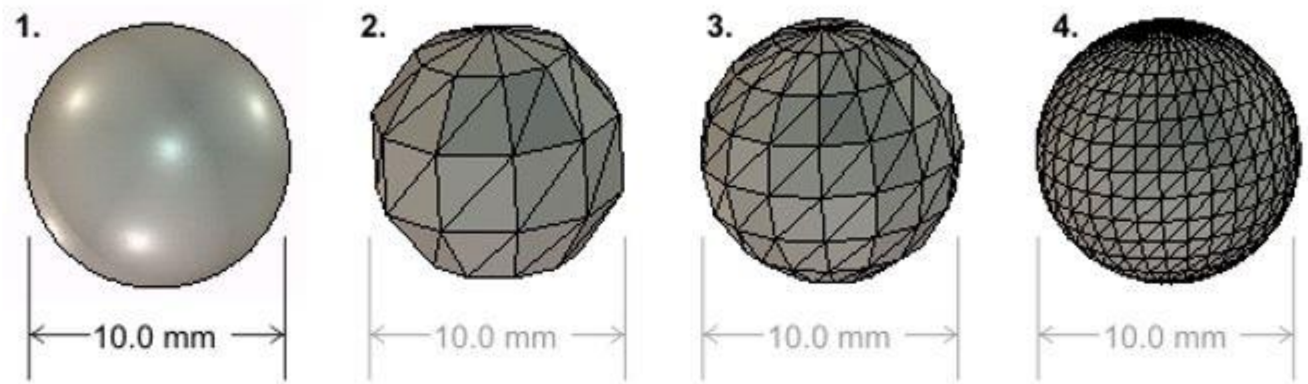
Shape approximation with triangles



Shape approximation with triangles



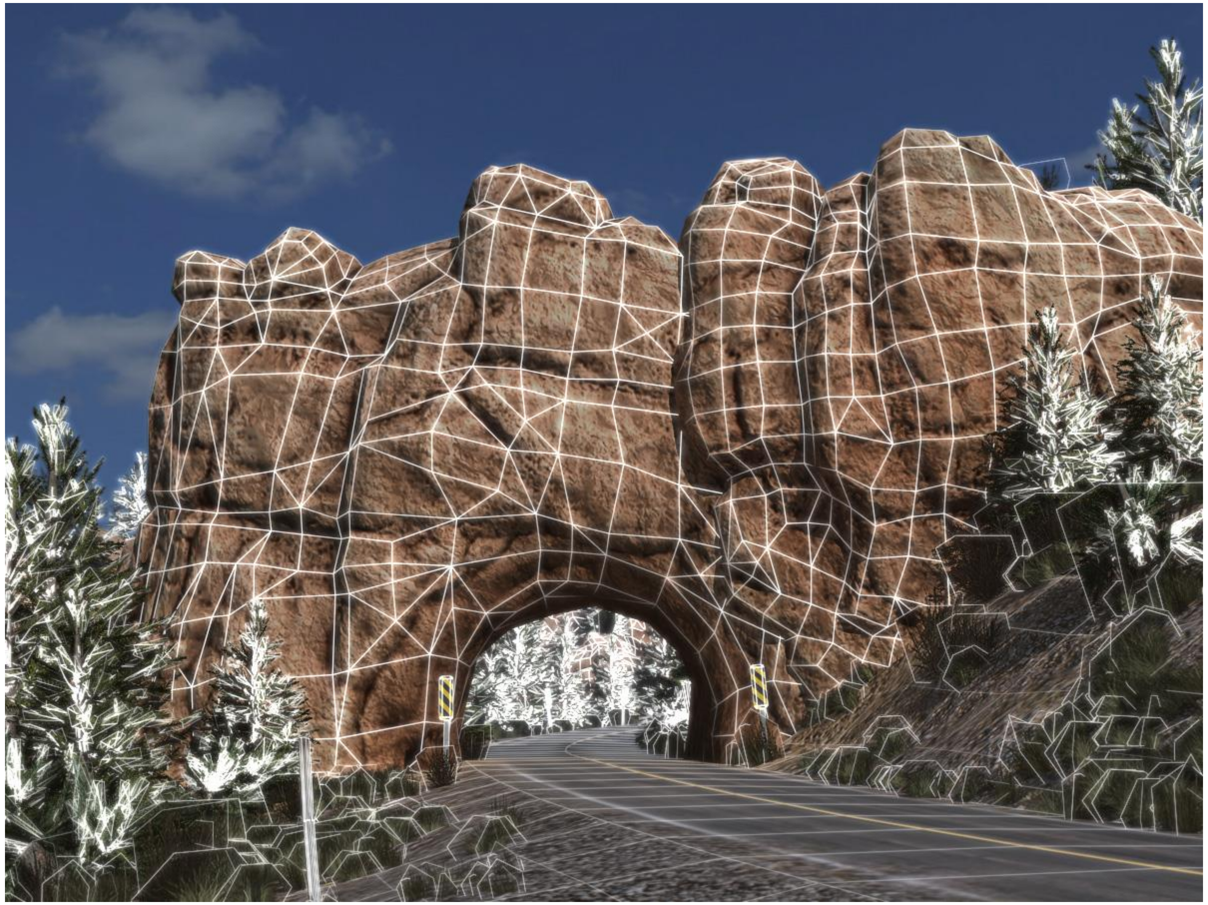
Shape approximation with triangles



World of polygons (triangles)

http://www.jeroenbackx.com/

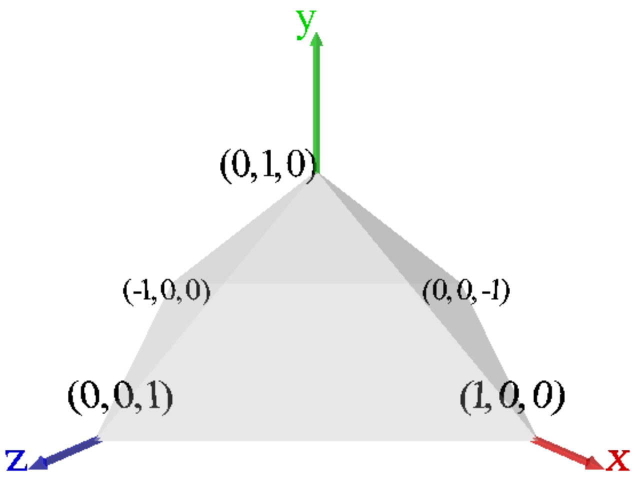
World of polygons (triangles)



http://www.jeroenbackx.com/

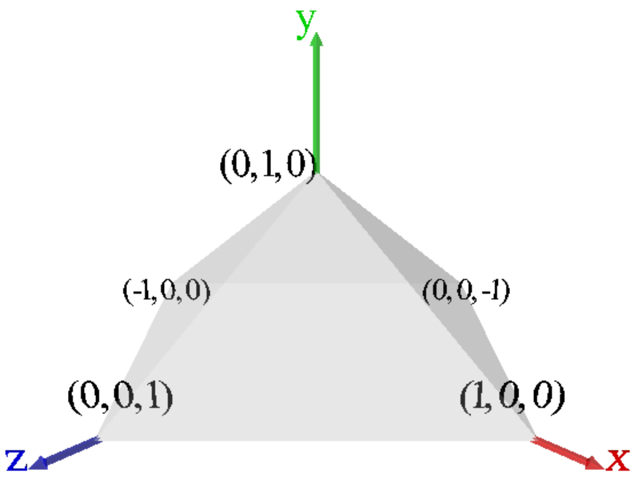
Translation of vertices?

Vertex transformations

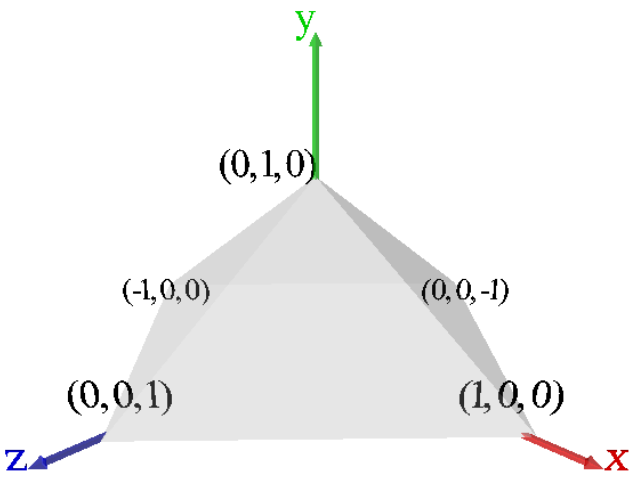


Scaling

Vertex transformations

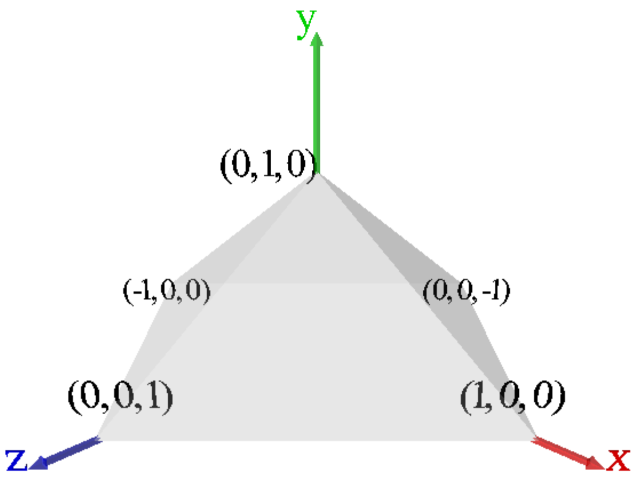


Scaling

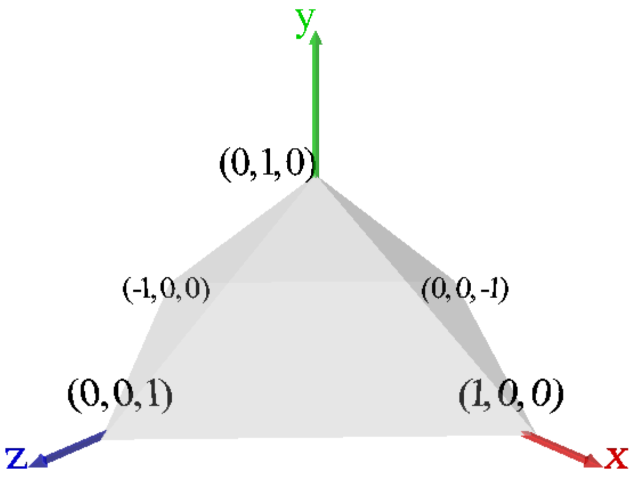


Rotation

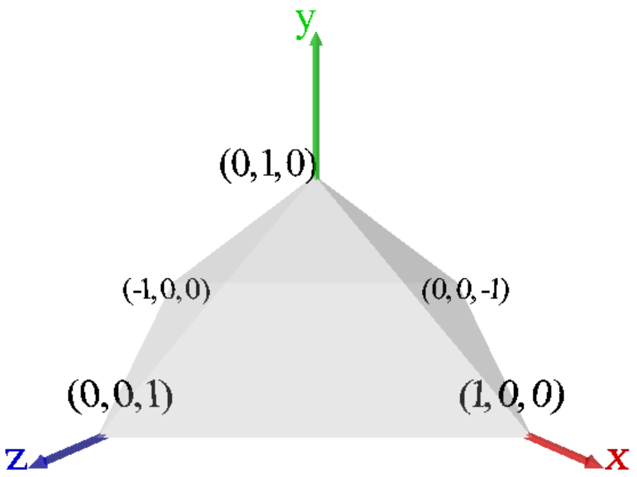
Vertex transformations



Scaling



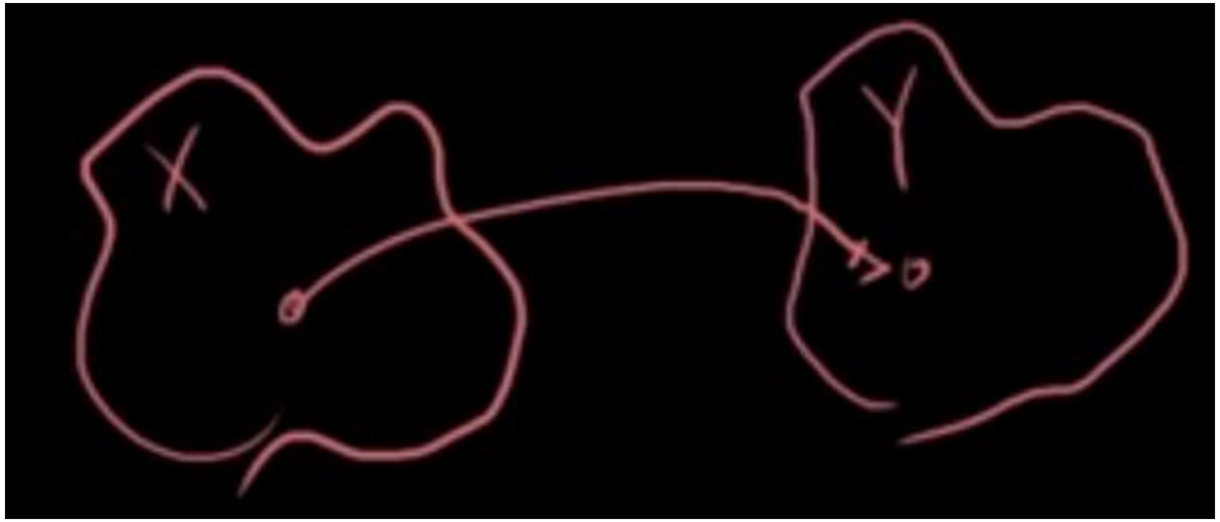
Rotation



Translation

Transformation

- $f: X \rightarrow Y$

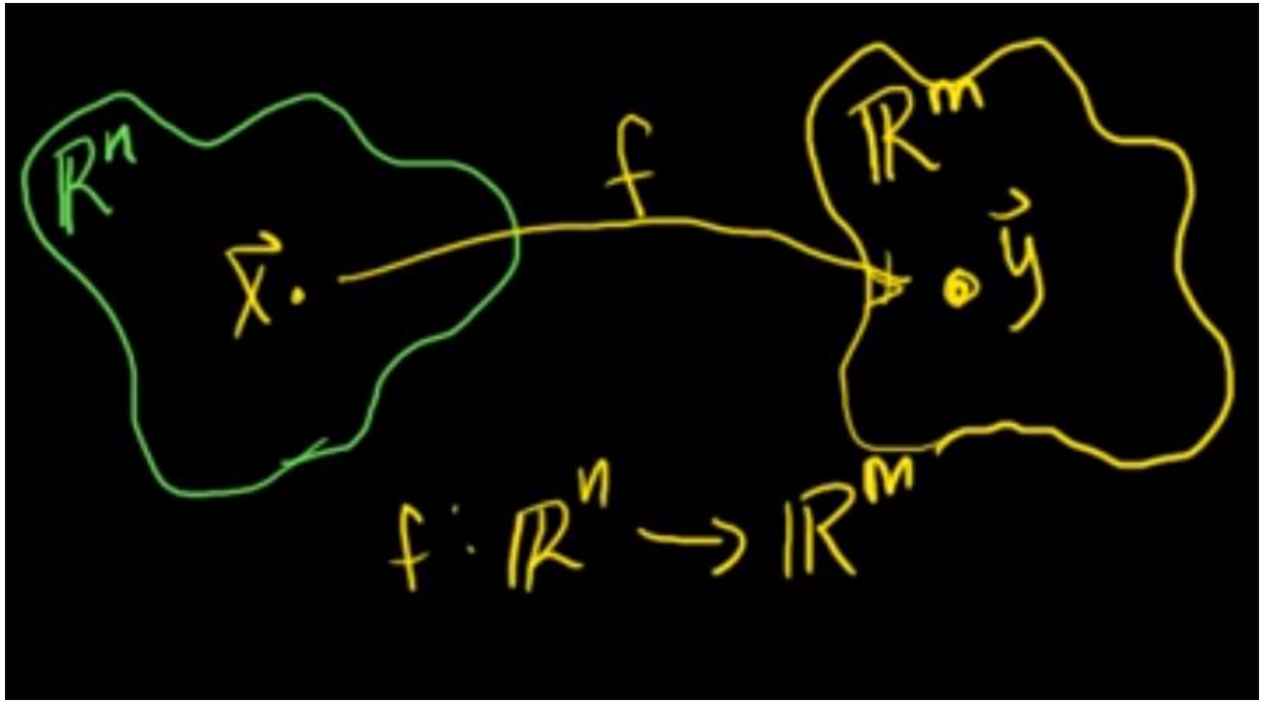


Transformation

- $f: X \rightarrow Y$

Vector-valued functions

$\mathbb{R}^n = \{(x_1, \dots, x_n): x_1, \dots, x_n \in \mathbb{R}\}$



Transformation

$f: \mathbb{R}^3 \to \mathbb{R}^2$

Transformation

$f: \mathbb{R}^3 \to \mathbb{R}^2$

$$f\left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}\right) = \begin{bmatrix} x_1 + 2x_2 \\ 3x_3 \end{bmatrix}$$

Transformation

$f: \mathbb{R}^3 \rightarrow \mathbb{R}^2$

$$f\left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}\right) = \begin{bmatrix} x_1 + 2x_2 \\ 3x_3 \end{bmatrix}$$

$$f\left(\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}\right)$$

Transformation

$f: \mathbb{R}^3 \to \mathbb{R}^2$

$$f\left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}\right) = \begin{bmatrix} x_1 + 2x_2 \\ 3x_3 \end{bmatrix}$$

$$f\left(\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} 1 + 2 \cdot 1 \\ 3 \cdot 1 \end{bmatrix}$$

Transformation

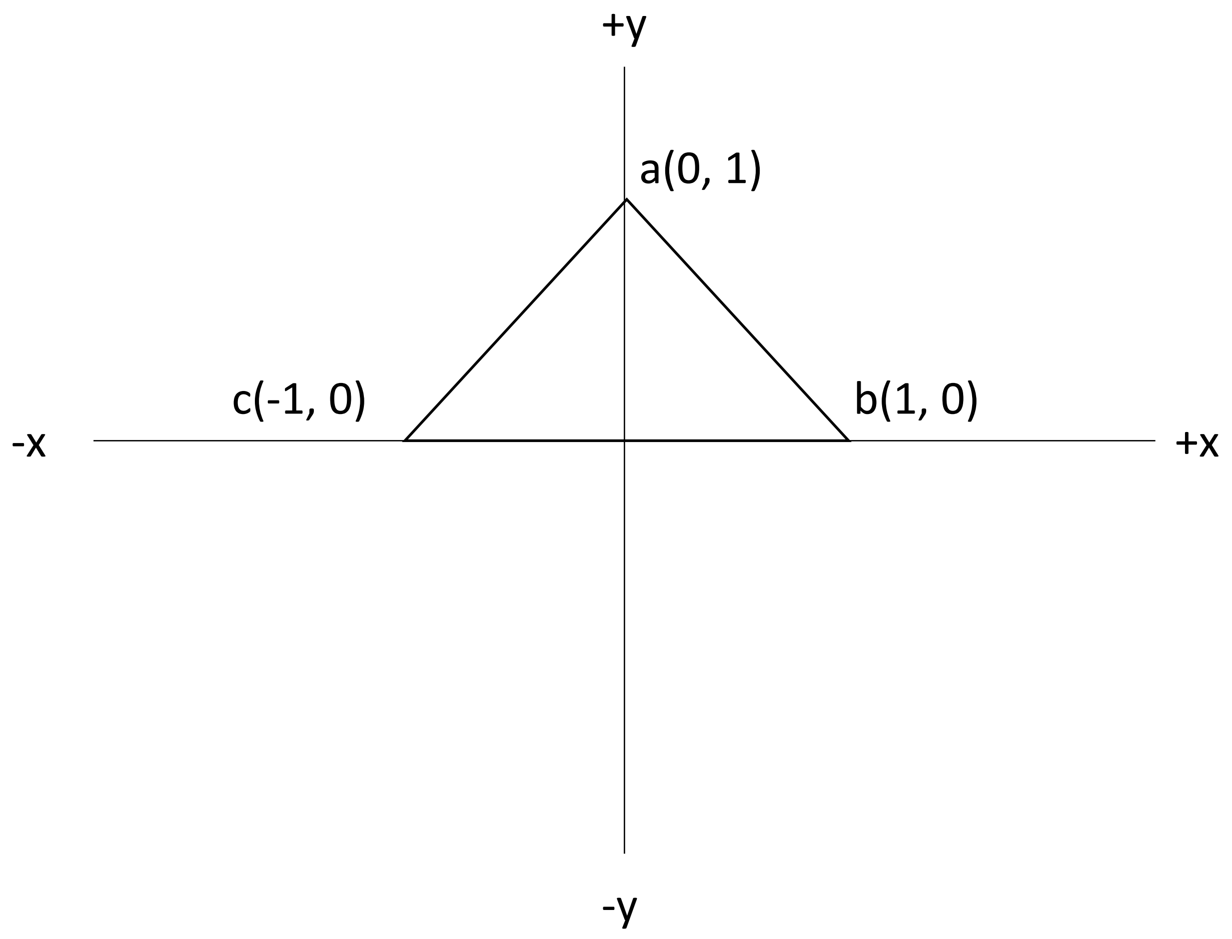
$f : \mathbb{R}^3 \to \mathbb{R}^2$

$$f\left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}\right) = \begin{bmatrix} x_1 + 2x_2 \\ 3x_3 \end{bmatrix}$$

$$f\left(\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} 1 + 2 \cdot 1 \\ 3 \cdot 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$$

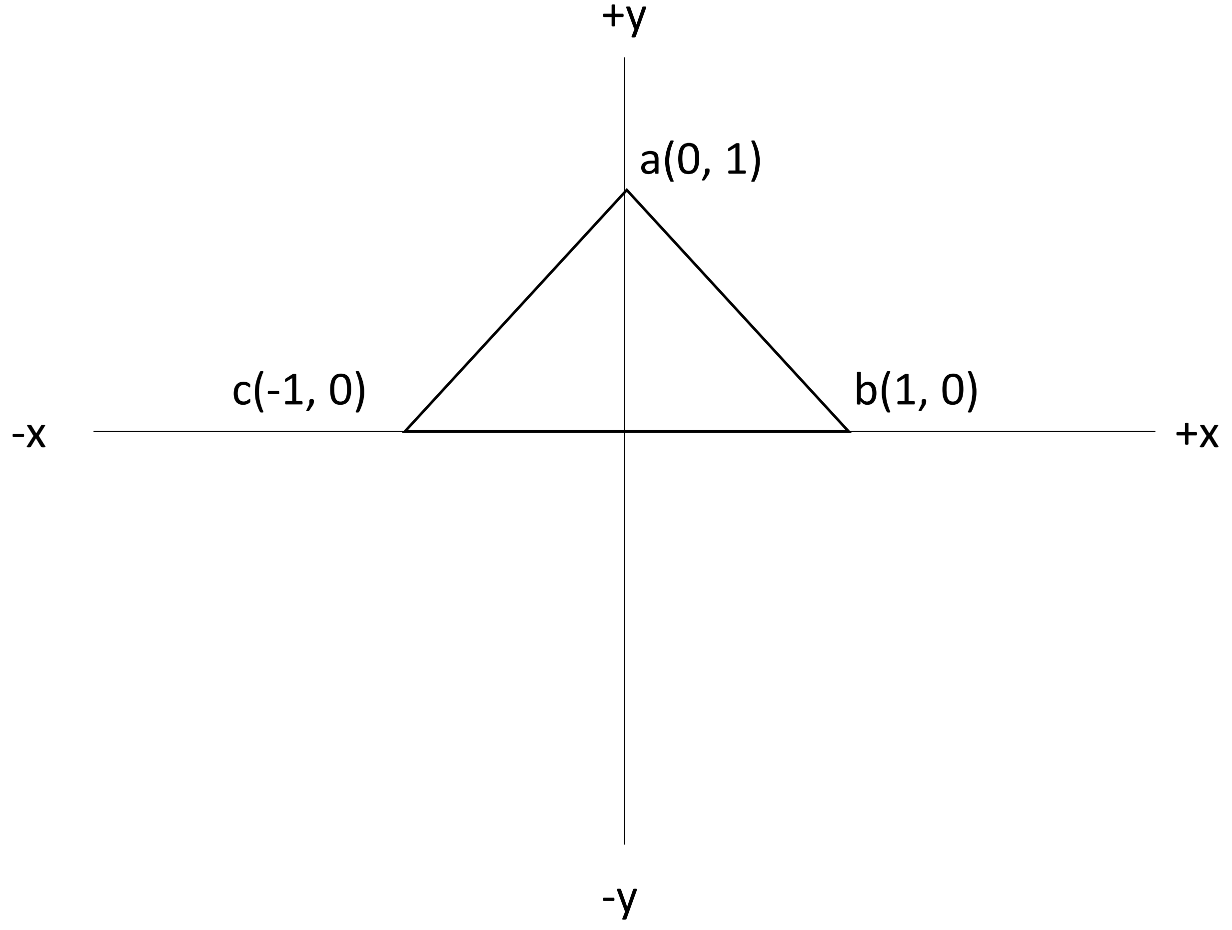
Scaling

$$f\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} x \cdot s_x \\ y \cdot s_y \end{bmatrix}$$



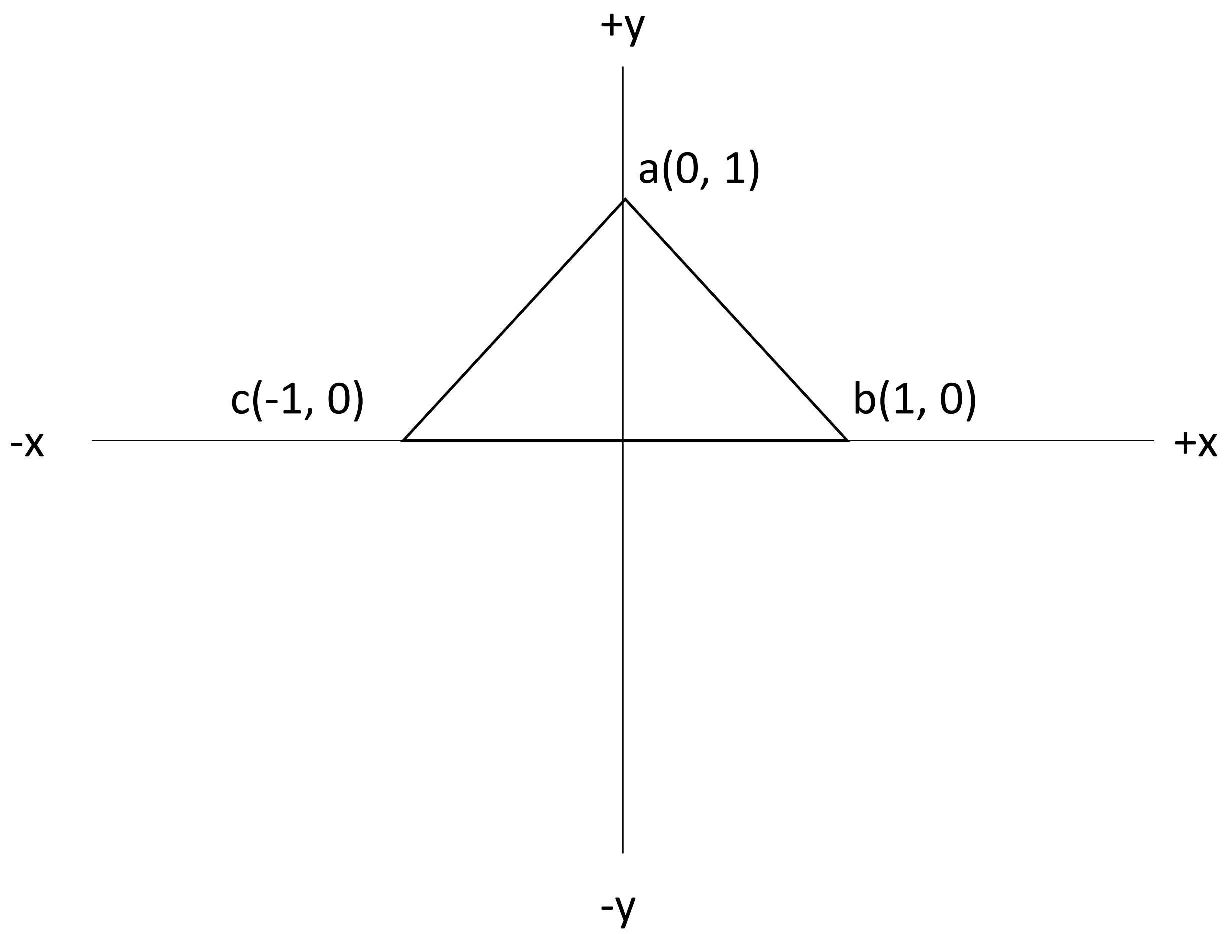
Scaling

$$f\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} x \cdot 2 \\ y \cdot 1 \end{bmatrix}$$



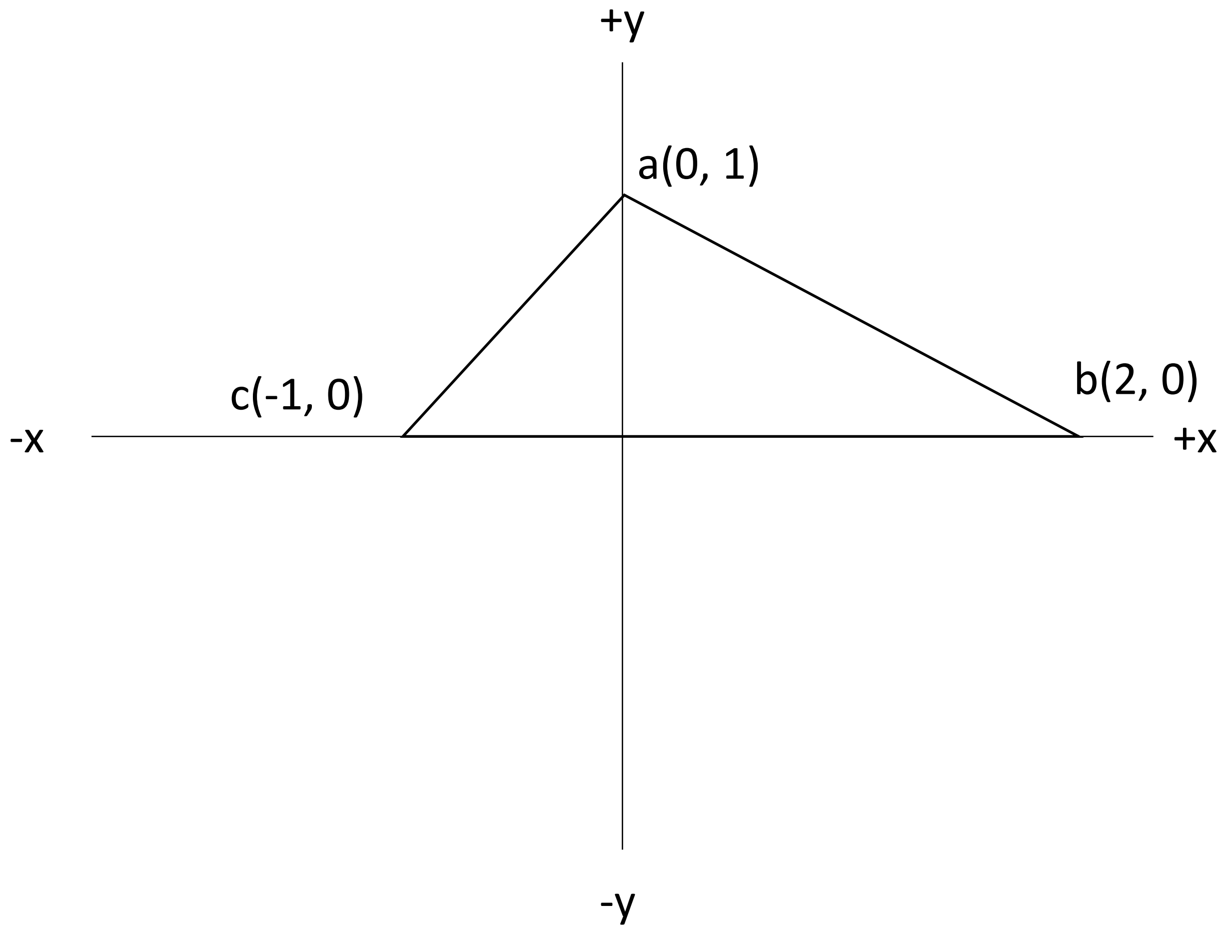
Scaling

$$f(\vec{b}) = \begin{bmatrix} 1 \cdot 2 \\ 0 \cdot 1 \end{bmatrix} = \begin{vmatrix} 2 \\ 0 \end{vmatrix}$$



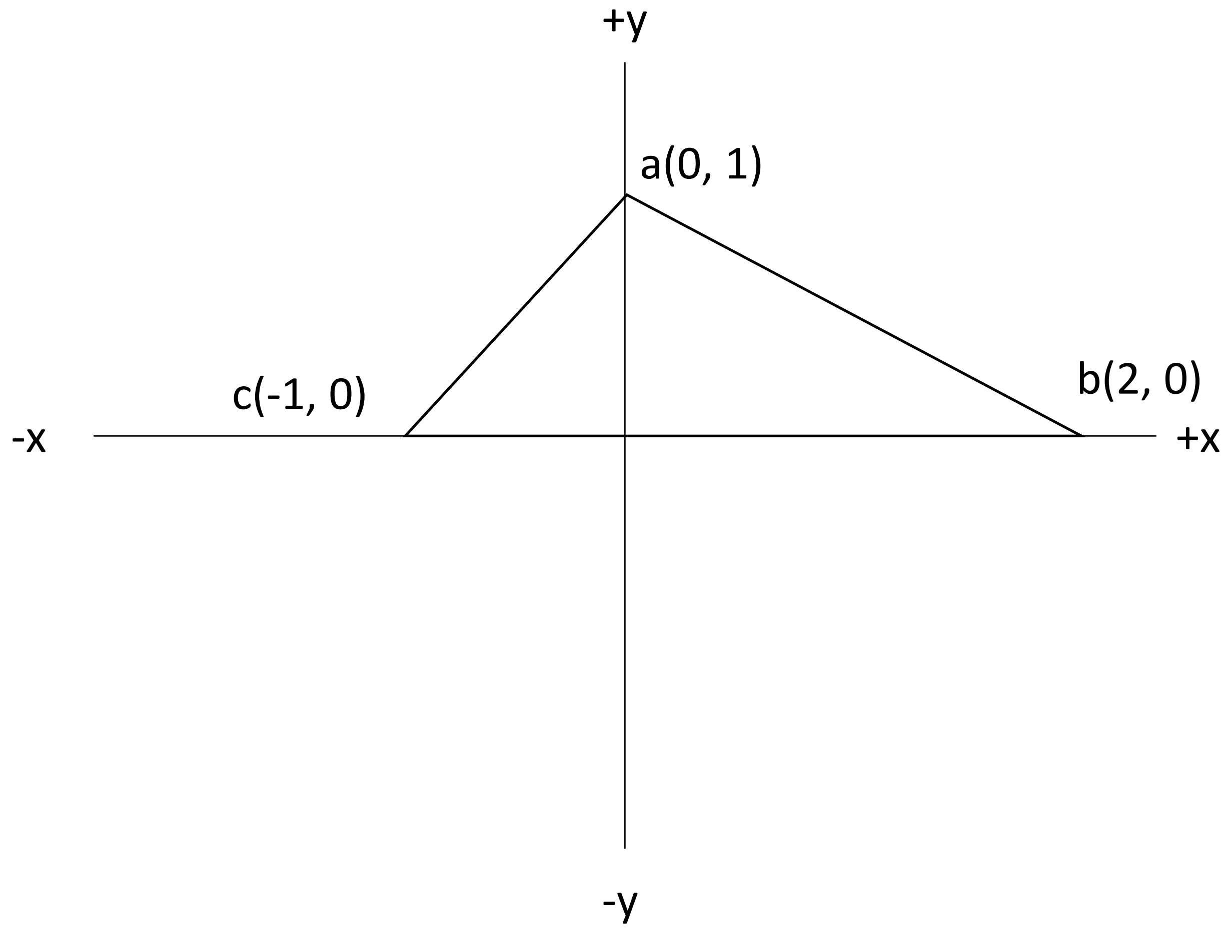
Scaling

$$f(\vec{b}) = \begin{bmatrix} 1 \cdot 2 \\ 0 \cdot 1 \end{bmatrix} = \begin{vmatrix} 2 \\ 0 \end{vmatrix}$$



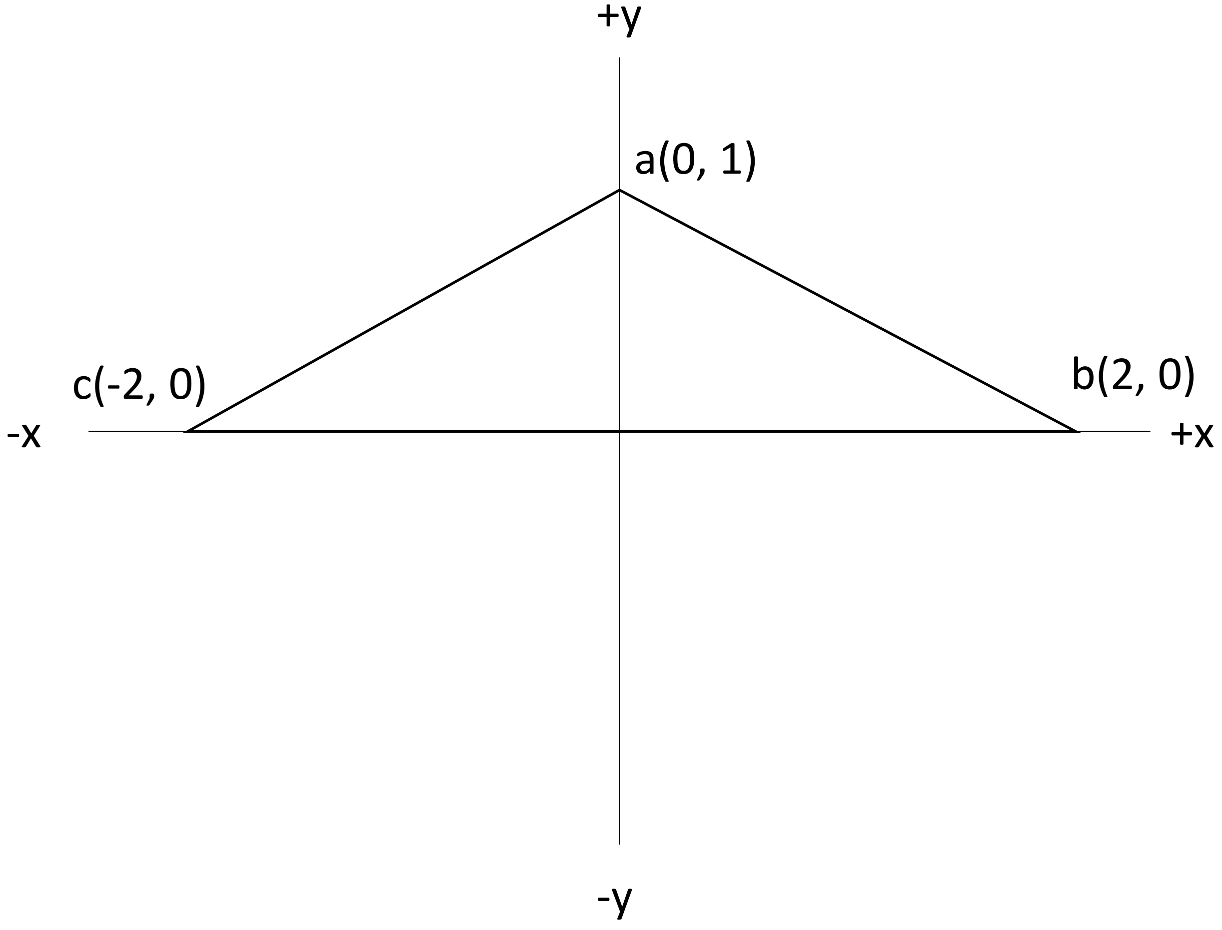
Scaling

$$f(\vec{c}) = \begin{bmatrix} -1 \cdot 2 \\ 0 \cdot 1 \end{bmatrix} = \begin{vmatrix} -2 \\ 0 \end{vmatrix}$$



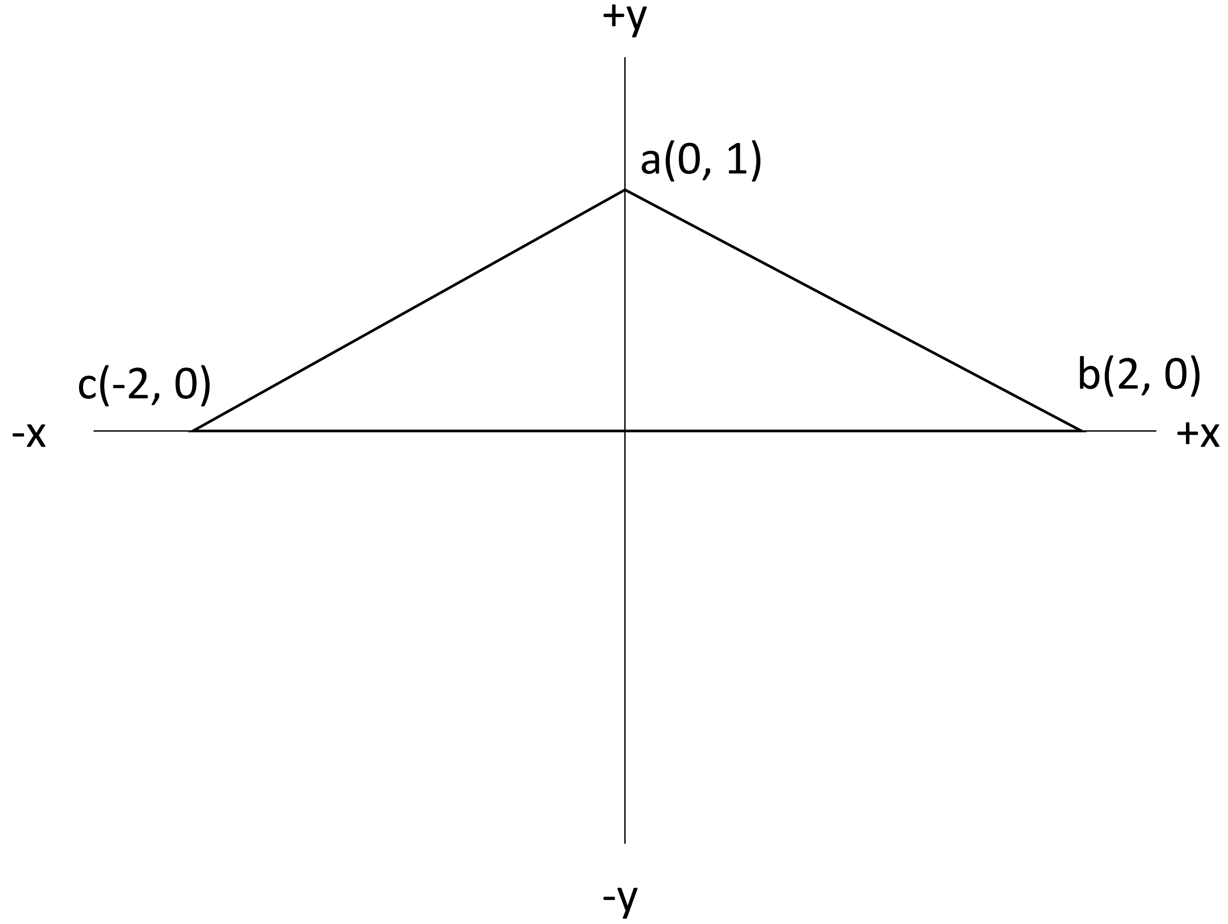
Scaling

$$f(\vec{c}) = \begin{bmatrix} -1 \cdot 2 \\ 0 \cdot 1 \end{bmatrix} = \begin{vmatrix} -2 \\ 0 \end{vmatrix}$$



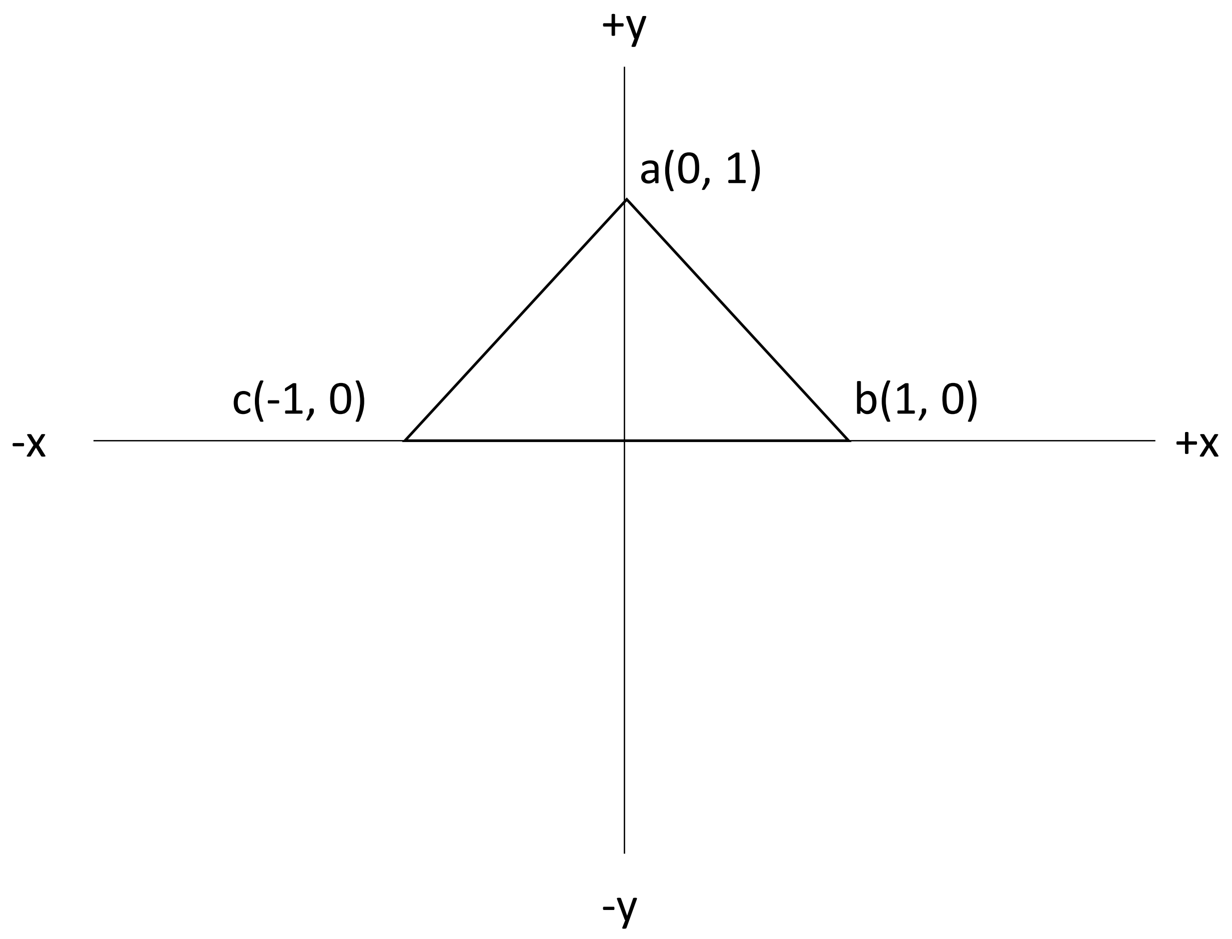
Scaling

$$f(\vec{a}) = \begin{bmatrix} 0 \cdot 2 \\ 1 \cdot 1 \end{bmatrix} = \begin{vmatrix} 0 \\ 1 \end{vmatrix}$$



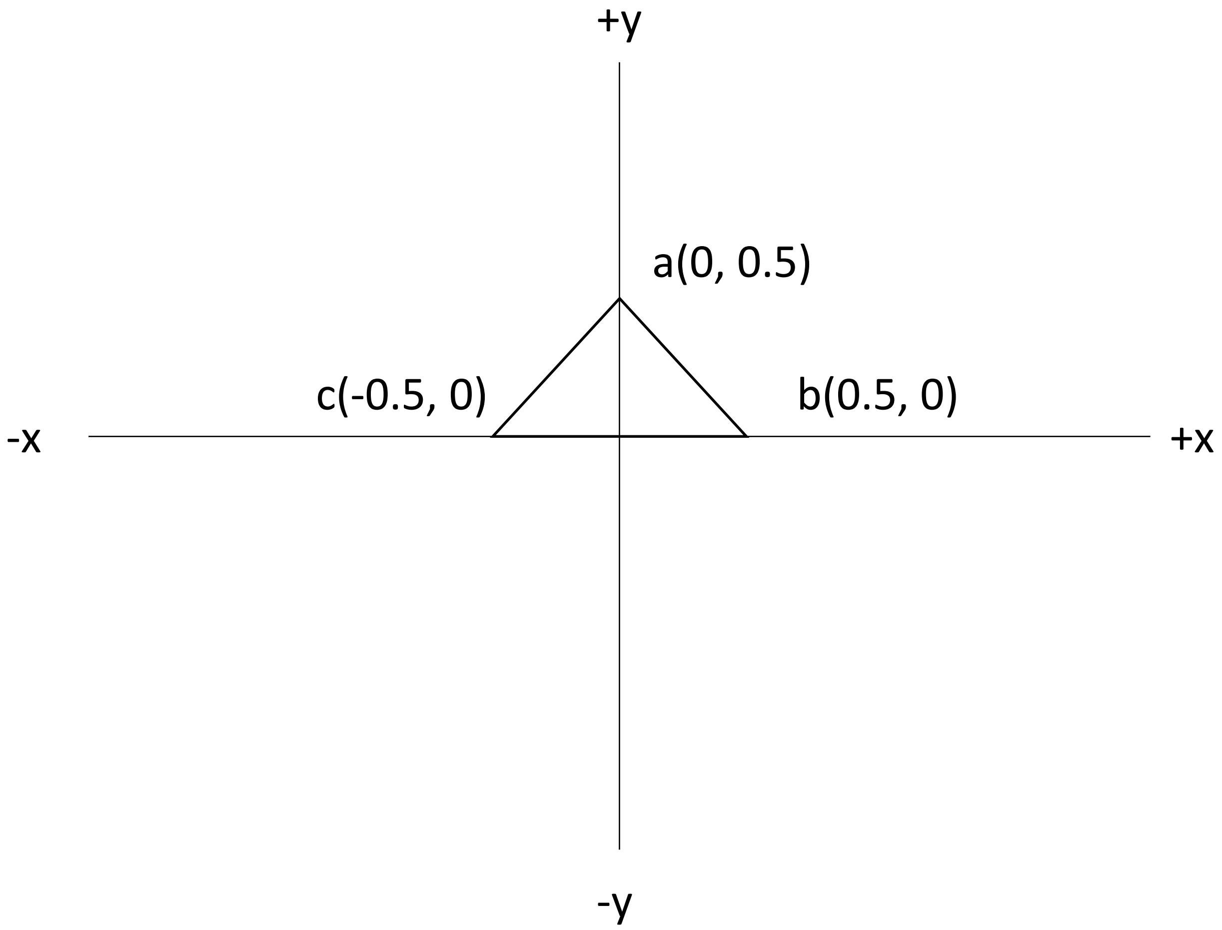
Scaling - XY

$$f\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} x \cdot 0.5 \\ y \cdot 0.5 \end{bmatrix}$$



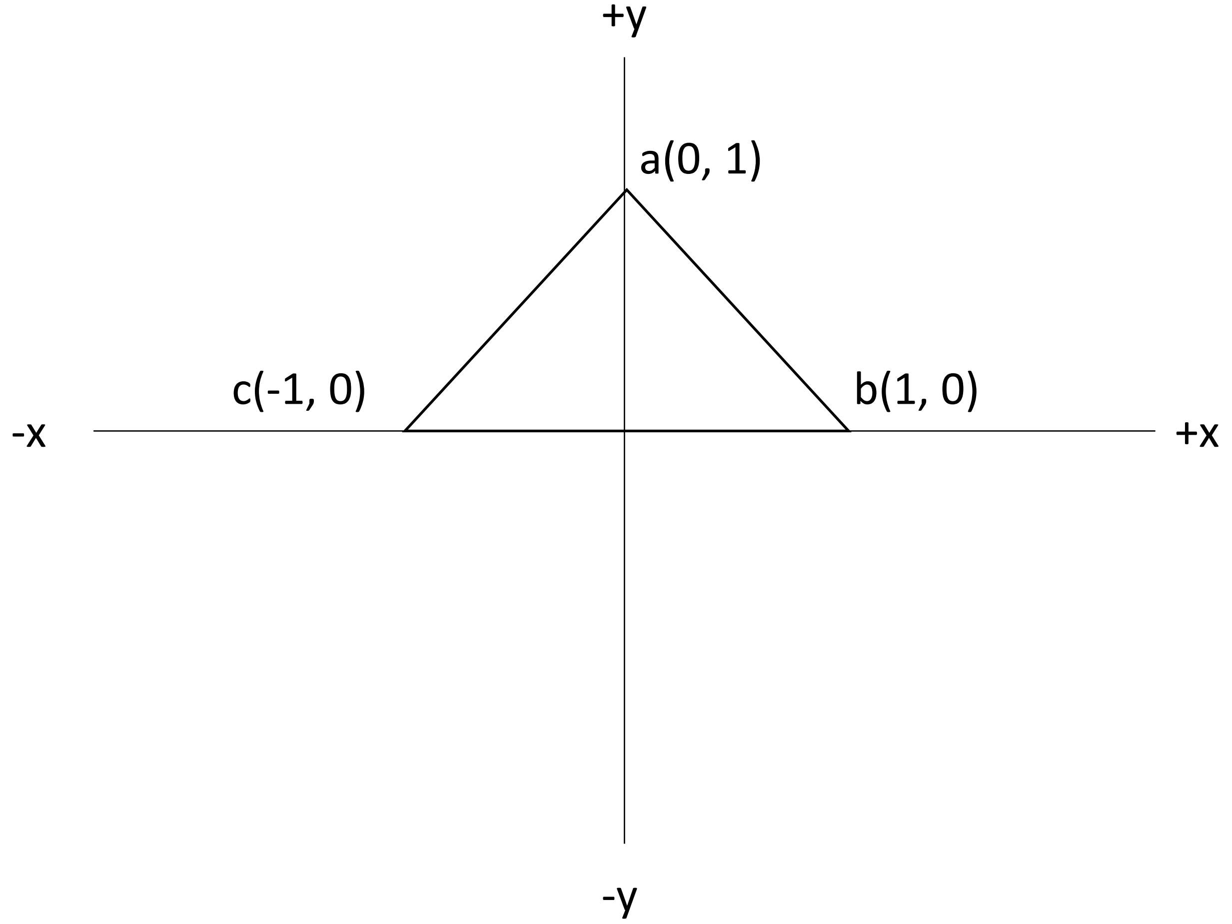
Scaling - XY

$$f\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} x \cdot 0.5 \\ y \cdot 0.5 \end{bmatrix}$$



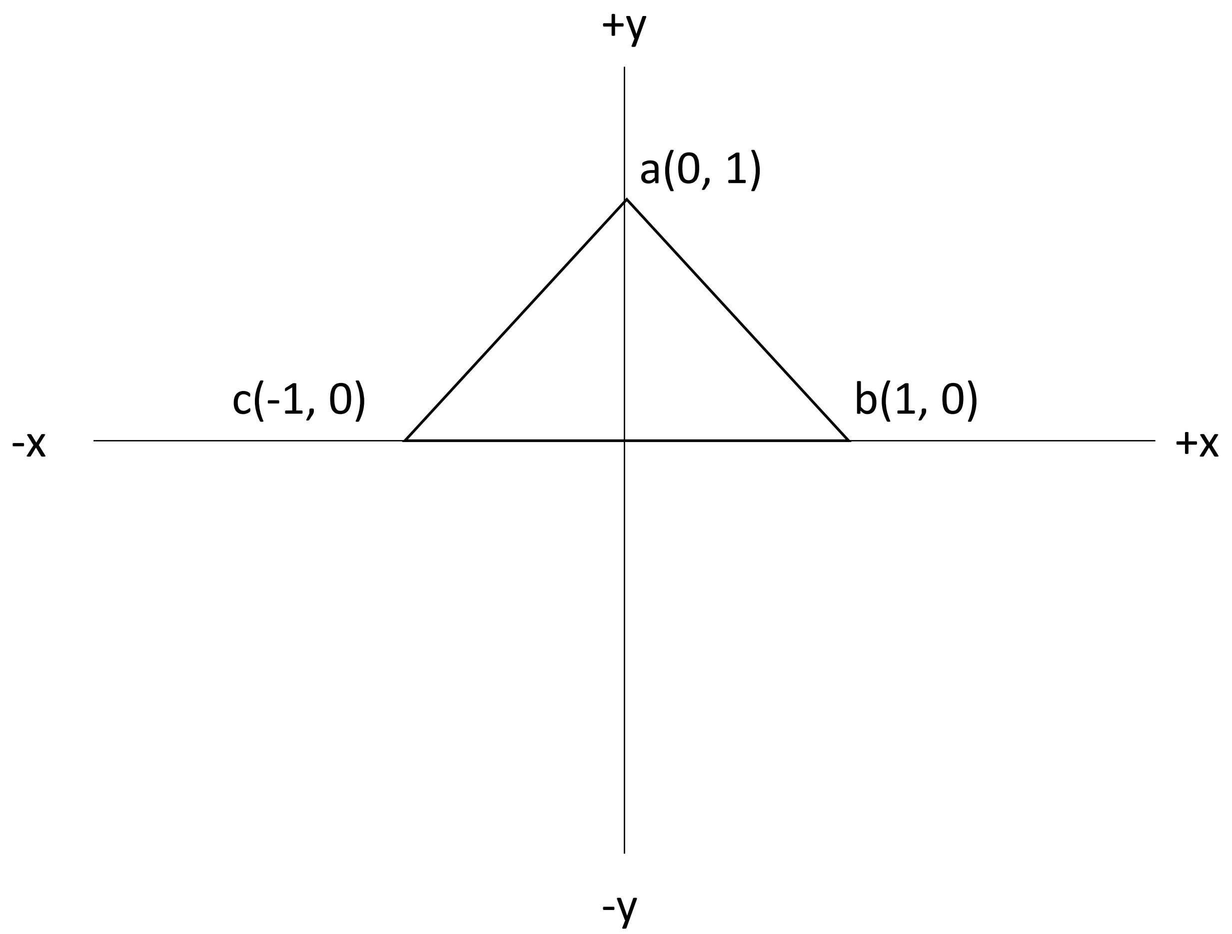
Translation

$$f\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} x + T_x \\ y + T_y \end{bmatrix}$$



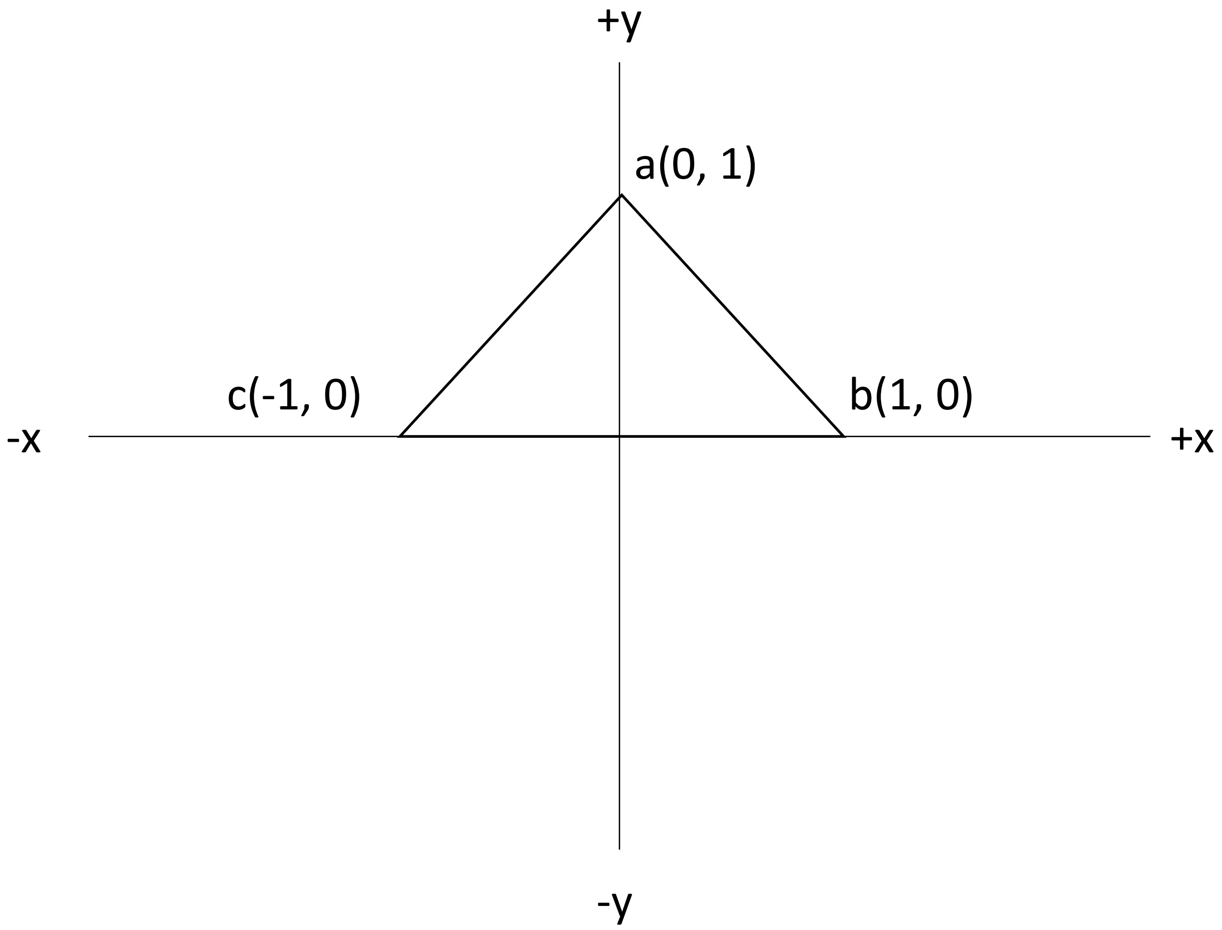
Translation

$$f\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} x + 1 \\ y \end{bmatrix}$$



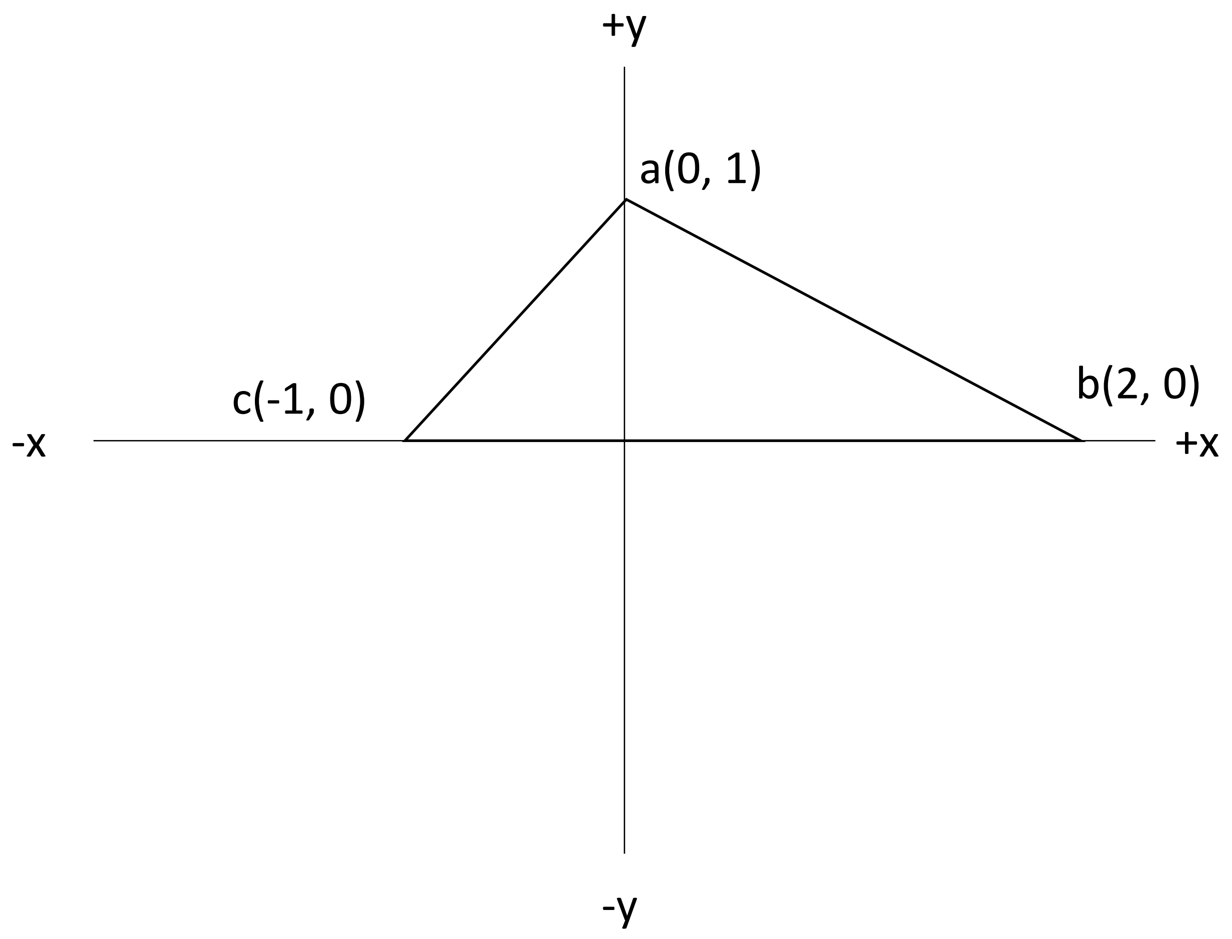
Translation

$$f(\vec{b}) = \begin{bmatrix} x+1 \\ y \end{bmatrix} = \begin{vmatrix} 2 \\ 0 \end{vmatrix}$$



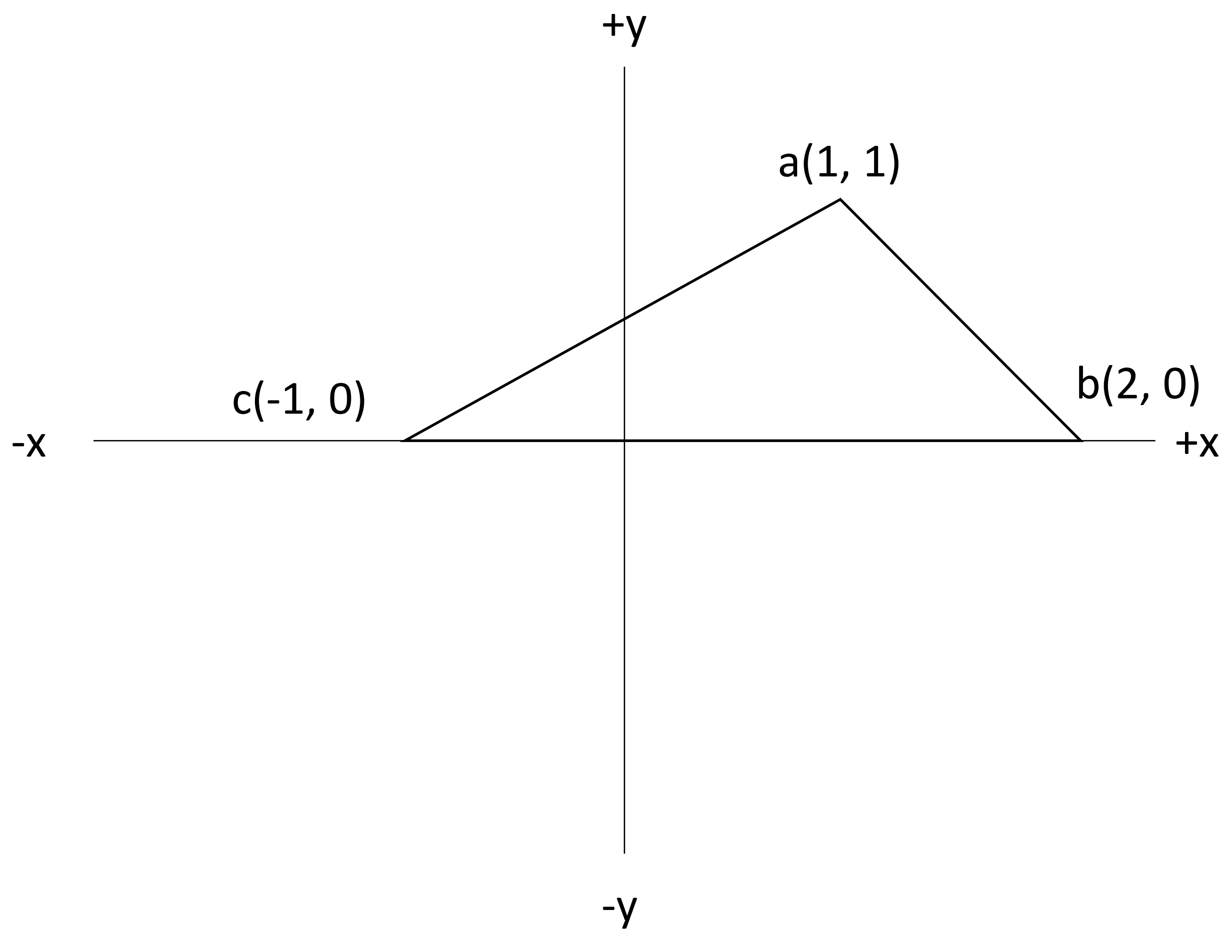
Translation

$$f(\vec{b}) = \begin{bmatrix} 1+1 \\ 0 \end{bmatrix} = \begin{vmatrix} 2 \\ 0 \end{vmatrix}$$



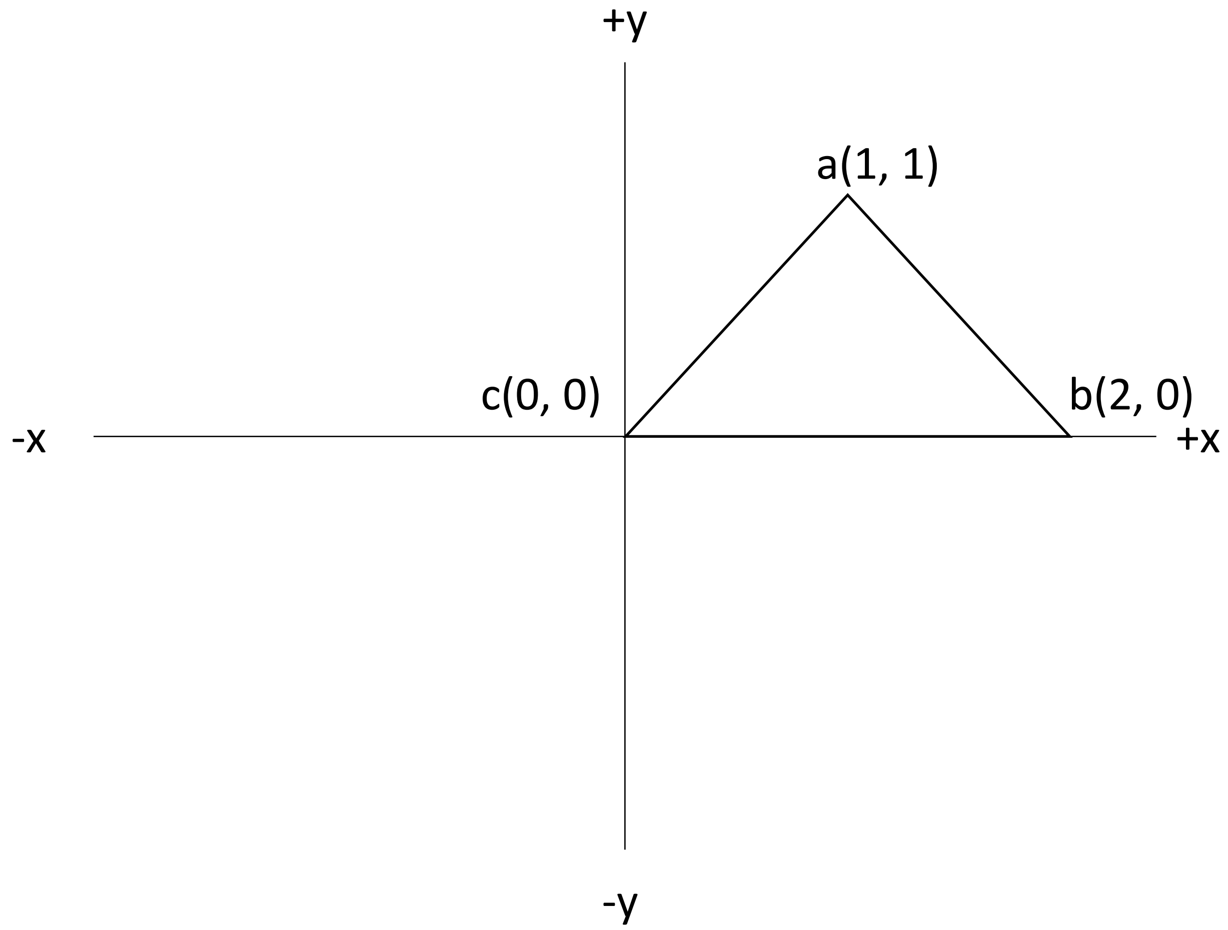
Translation

$$f(\vec{a}) = \begin{bmatrix} 0+1 \\ 1 \end{bmatrix} = \begin{vmatrix} 1 \\ 1 \end{vmatrix}$$



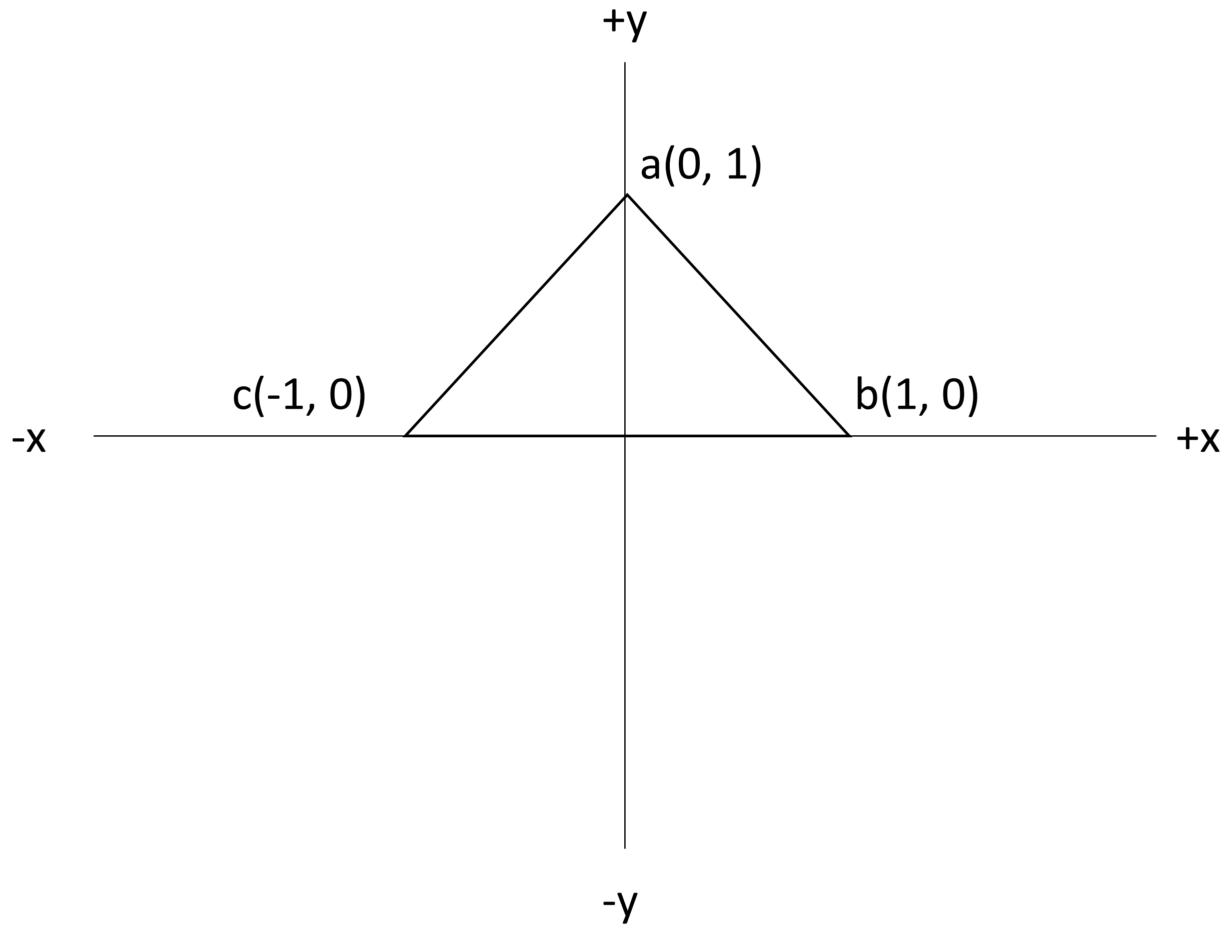
Translation

$$f(\vec{c}) = \begin{bmatrix} -1 + 1 \\ 0 \end{bmatrix} = \begin{vmatrix} 0 \\ 0 \end{vmatrix}$$



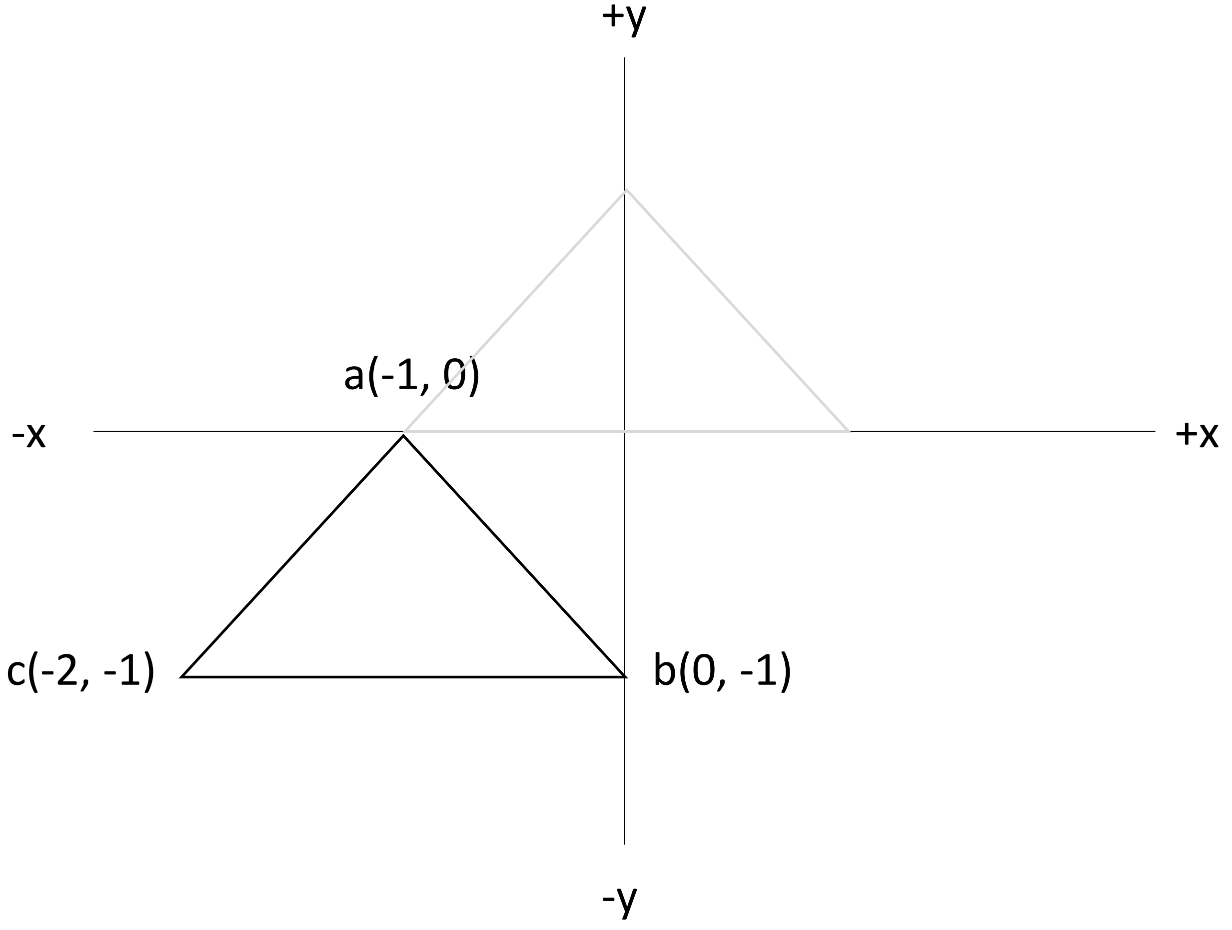
Translation - XY

$$f\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} x - 1 \\ y - 1 \end{bmatrix}$$

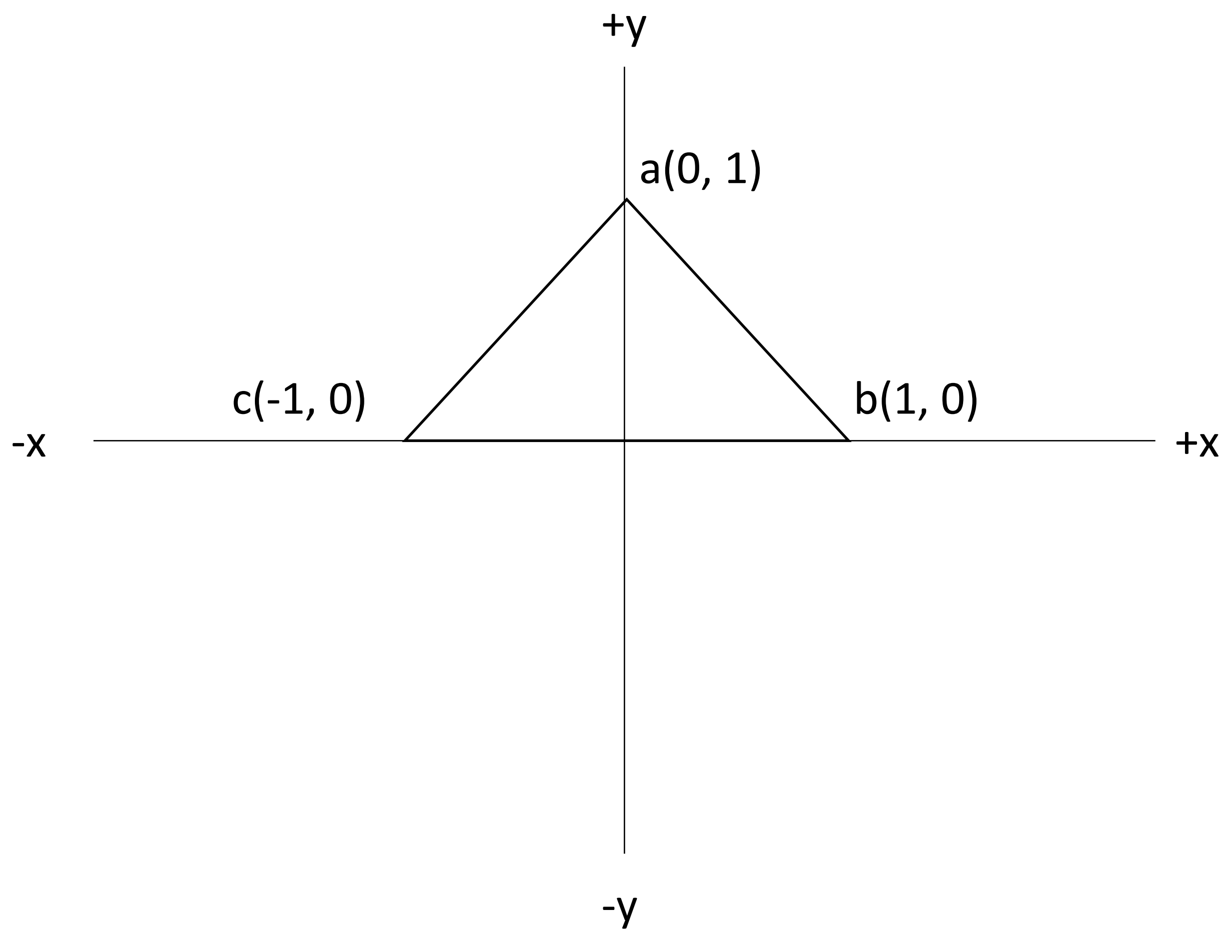


Translation - XY

$$f\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} x - 1 \\ y - 1 \end{bmatrix}$$

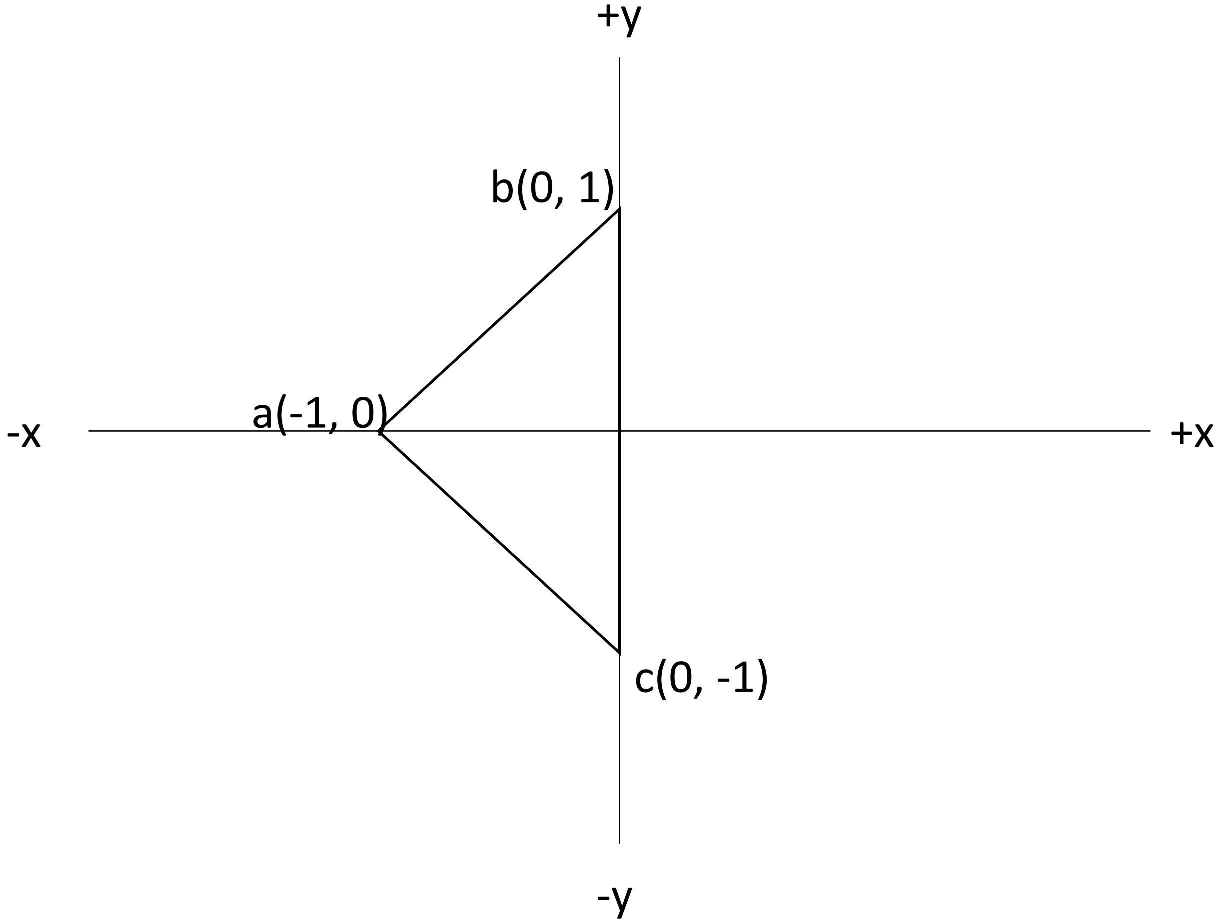


Rotation



Rotation

$\delta = 90°$

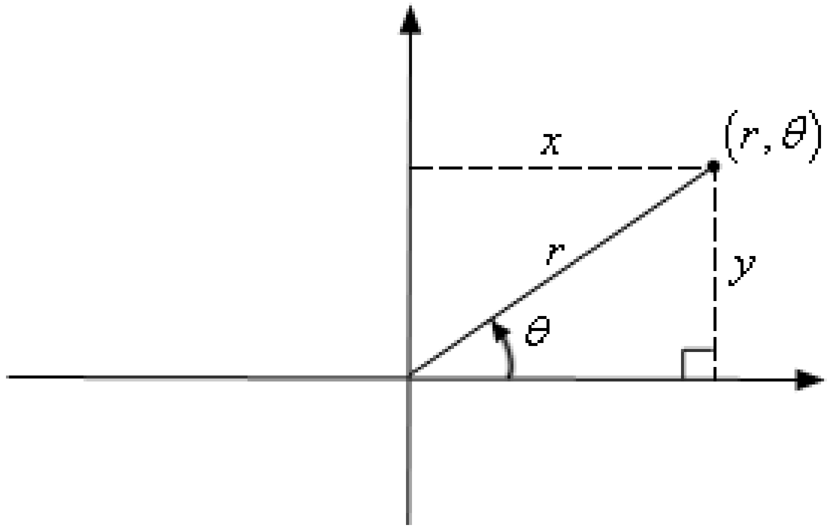


Rotation

Polar coordinates :

$x \quad = r \cdot \cos(\theta)$

$y \quad = r \cdot \sin(\theta)$

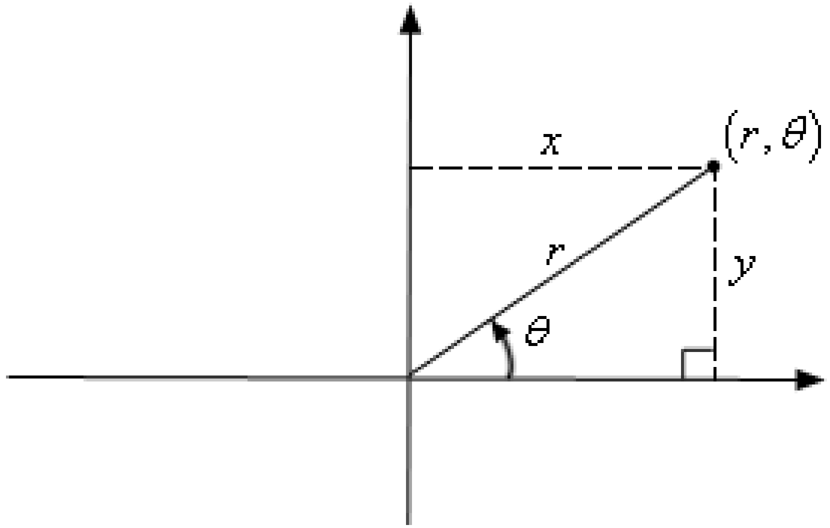


Rotation

Polar coordinates :

$x \quad = r \cdot \cos(\theta)$

$y \quad = r \cdot \sin(\theta)$



$x' = r \cdot \cos(\theta + \delta)$

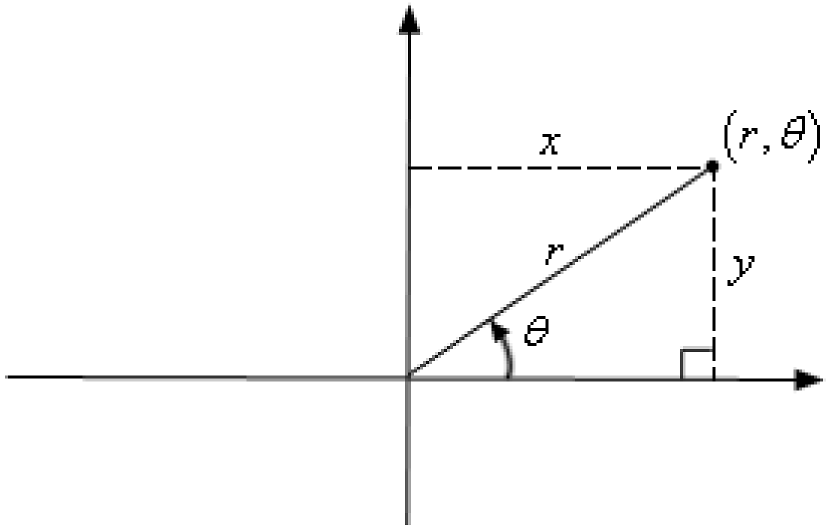
$y' = r \cdot \sin(\theta + \delta)$

Rotation

Polar coordinates :

$x \quad = r \cdot \cos(\theta)$

$y \quad = r \cdot \sin(\theta)$



$x' = r \cdot \cos(\theta + \ \delta) = r\cos(\theta)\cos(\delta) - r\sin(\theta)\sin(\delta)$

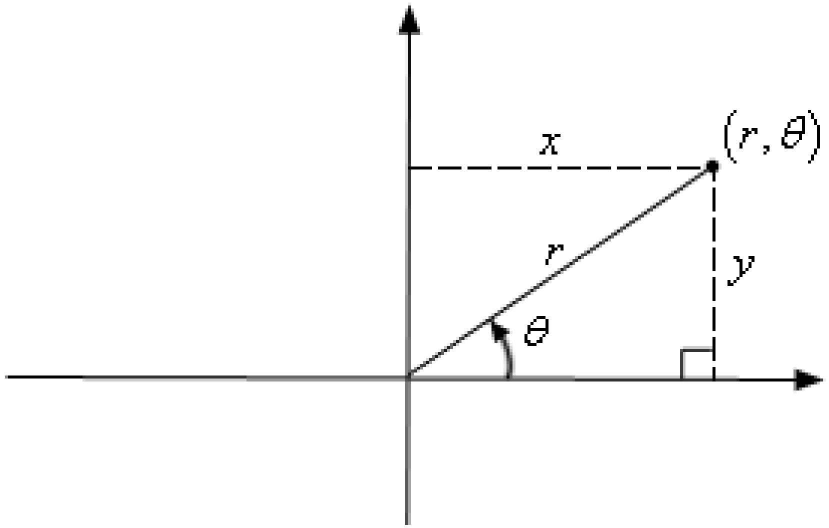
$y' = r \cdot \sin(\theta + \ \delta) = r\sin(\theta)\cos(\delta) + r\cos(\theta)\sin(\delta)$

Rotation

Polar coordinates :

$x \quad = r \cdot \cos(\theta)$

$y \quad = r \cdot \sin(\theta)$



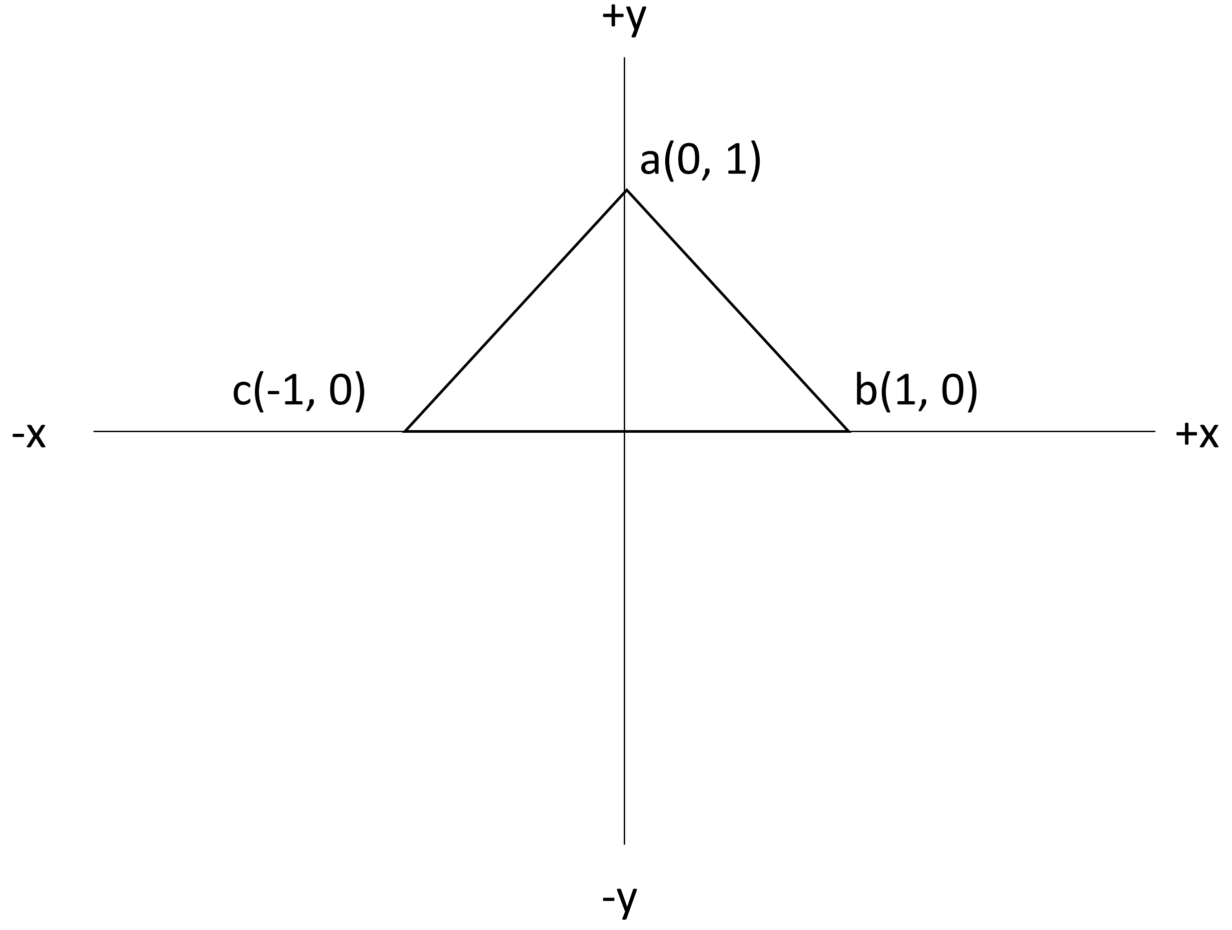
$$x' = r \cdot \cos(\theta + \ \delta) = r\cos(\theta)\cos(\delta) - r\sin(\theta)\sin(\delta)$$
$$y' = r \cdot \sin(\theta + \ \delta) = r\sin(\theta)\cos(\delta) + r\cos(\theta)\sin(\delta)$$

$$x' = x \cdot \cos(\delta) - y \cdot \sin(\delta)$$
$$y' = x \cdot \sin(\delta) + y \cdot \cos(\delta)$$

Rotation

$$f\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} x \cdot \cos(\theta) - y \cdot \sin(\theta) \\ x \cdot \sin(\theta) + y \cdot \cos(\theta) \end{bmatrix}$$

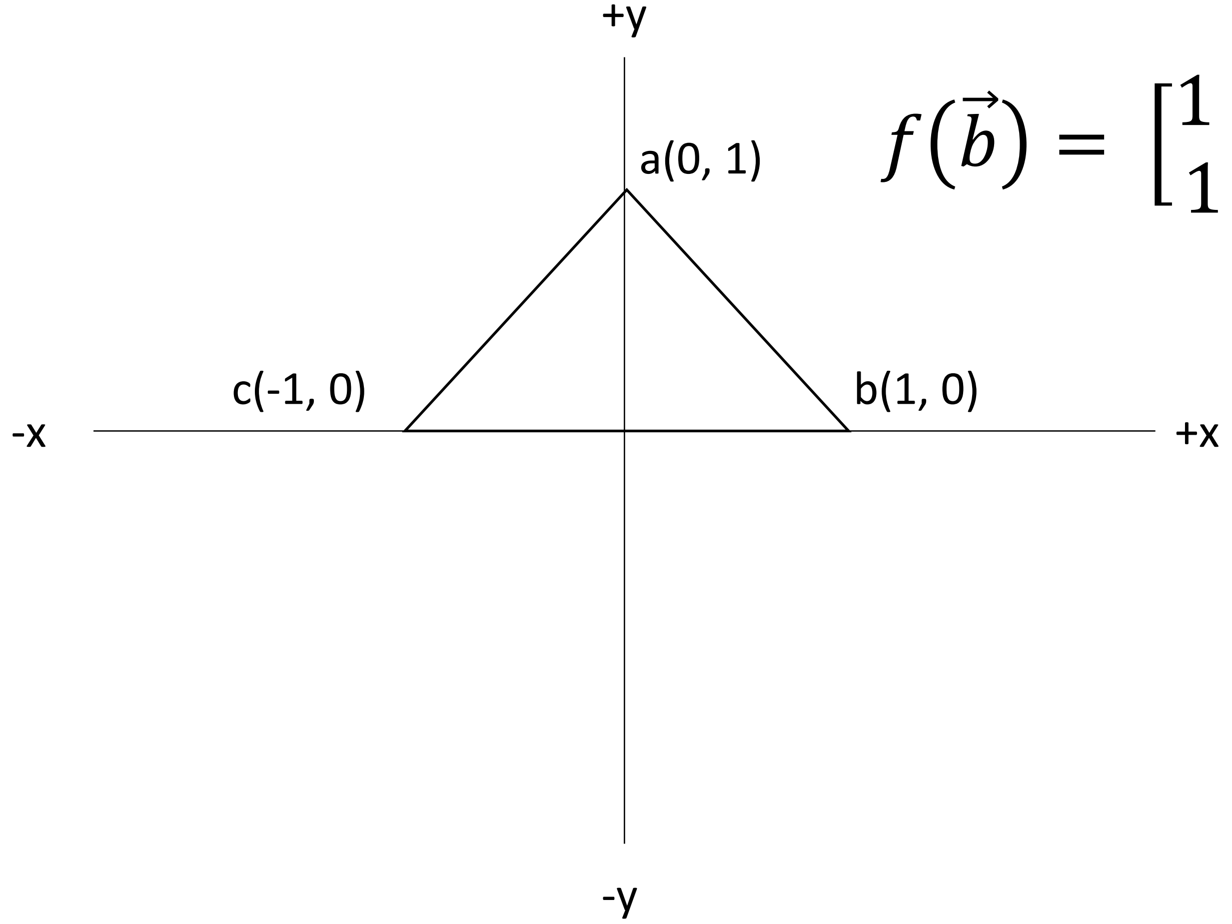
$\Theta = 90°$



Rotation

$$f\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} x \cdot \cos(\theta) - y \cdot \sin(\theta) \\ x \cdot \sin(\theta) + y \cdot \cos(\theta) \end{bmatrix}$$

$\Theta = 90°$



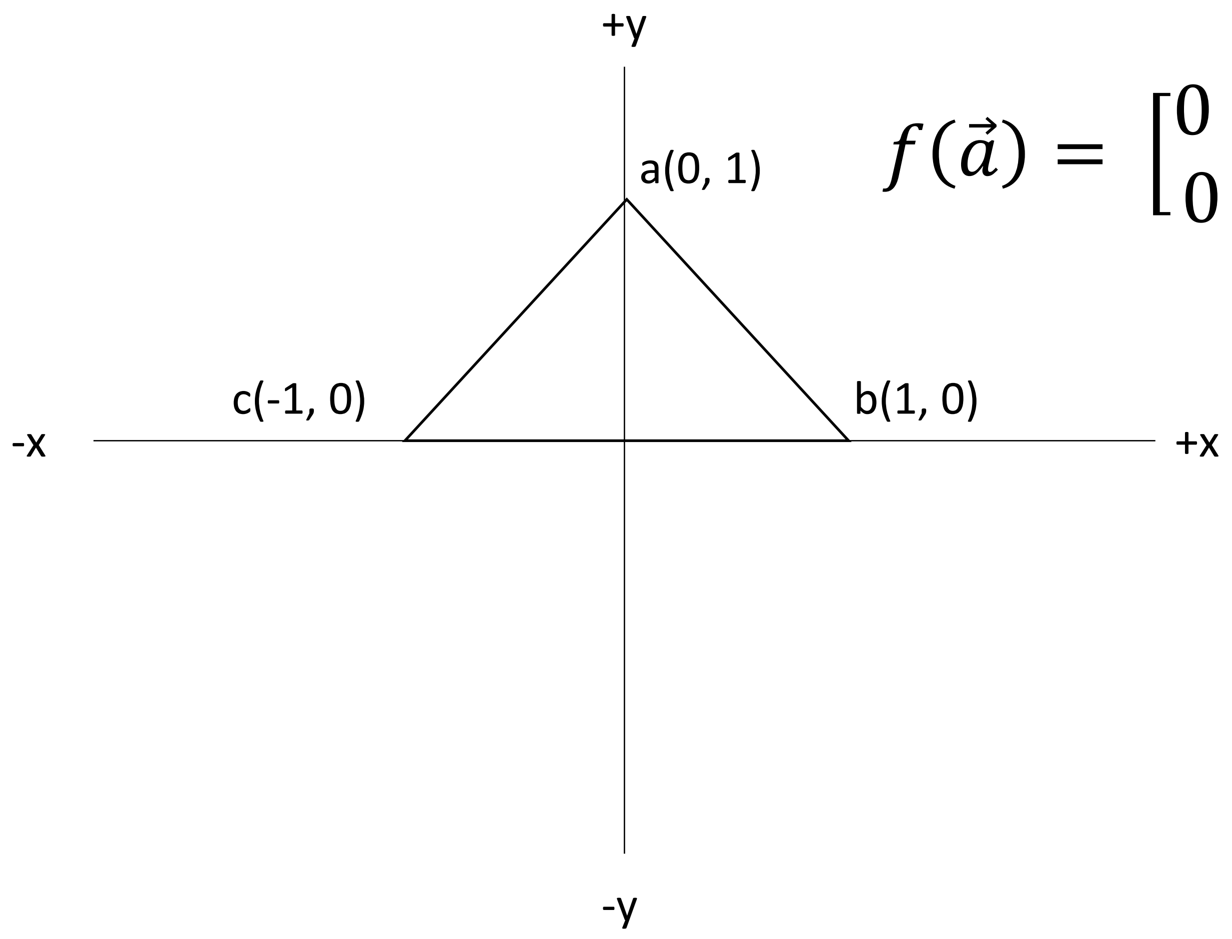
$$f(\vec{b}) = \begin{bmatrix} 1 \cdot 0 - 0 \cdot 1 \\ 1 \cdot 1 + 0 \cdot 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Rotation

$$f\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} x \cdot \cos(\theta) - y \cdot \sin(\theta) \\ x \cdot \sin(\theta) + y \cdot \cos(\theta) \end{bmatrix}$$

$$\Theta = 90°$$

$$f(\vec{a}) = \begin{bmatrix} 0 \cdot 0 - 1 \cdot 1 \\ 0 \cdot 1 + 1 \cdot 0 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

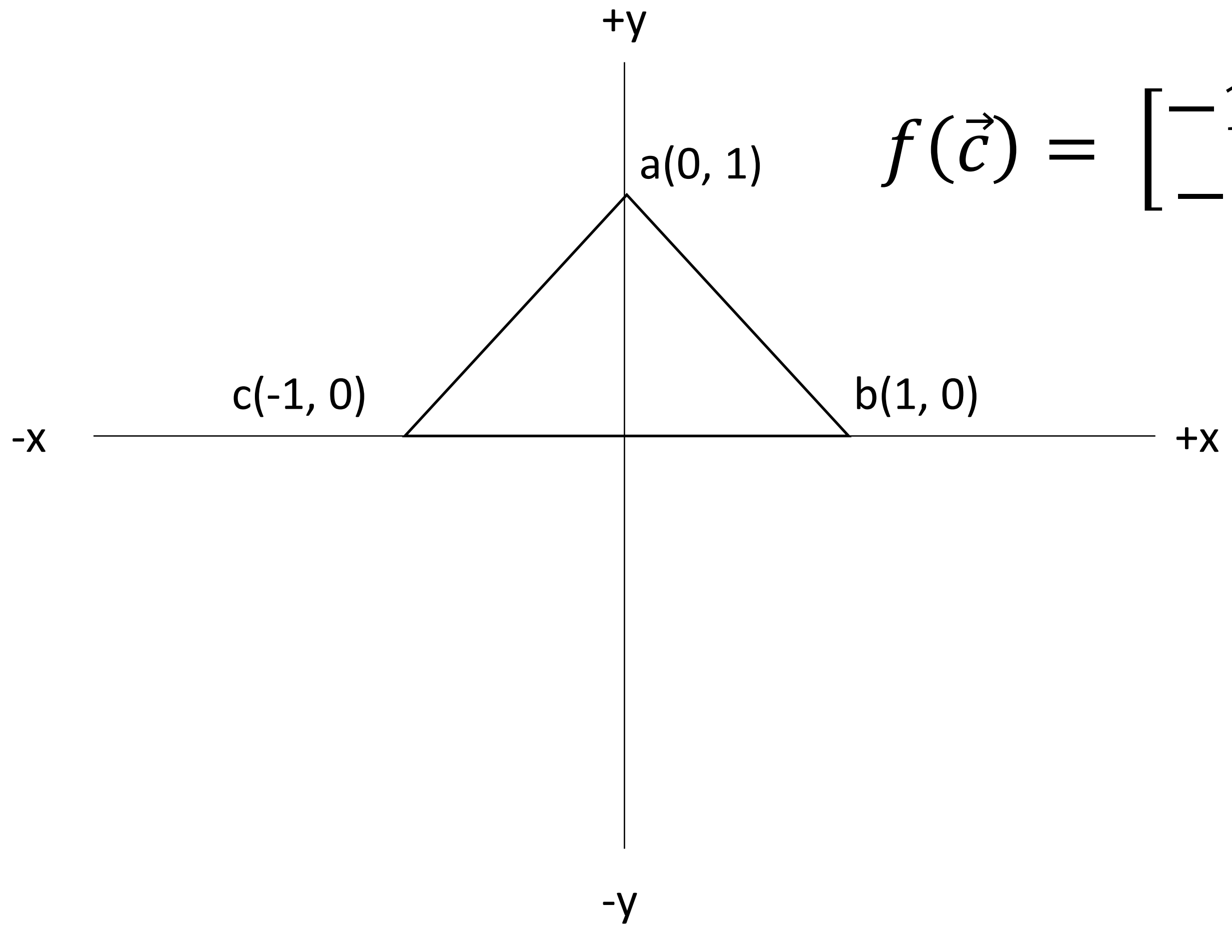


Rotation

$$\Theta = 90°$$

$$f\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} x \cdot \cos(\theta) - y \cdot \sin(\theta) \\ x \cdot \sin(\theta) + y \cdot \cos(\theta) \end{bmatrix}$$

$$f(\vec{c}) = \begin{bmatrix} -1 \cdot 0 - 0 \cdot 1 \\ -1 \cdot 1 + 0 \cdot 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

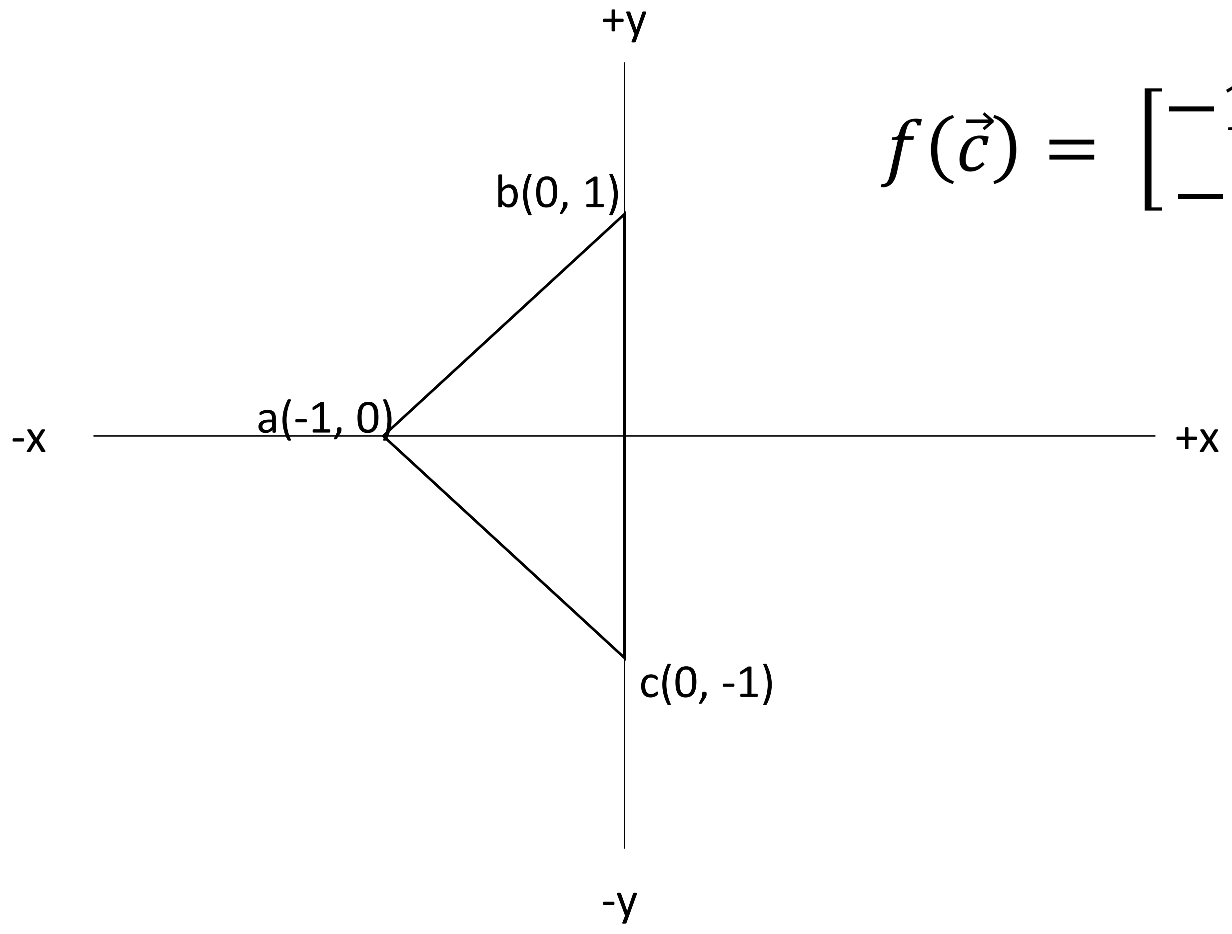


Rotation

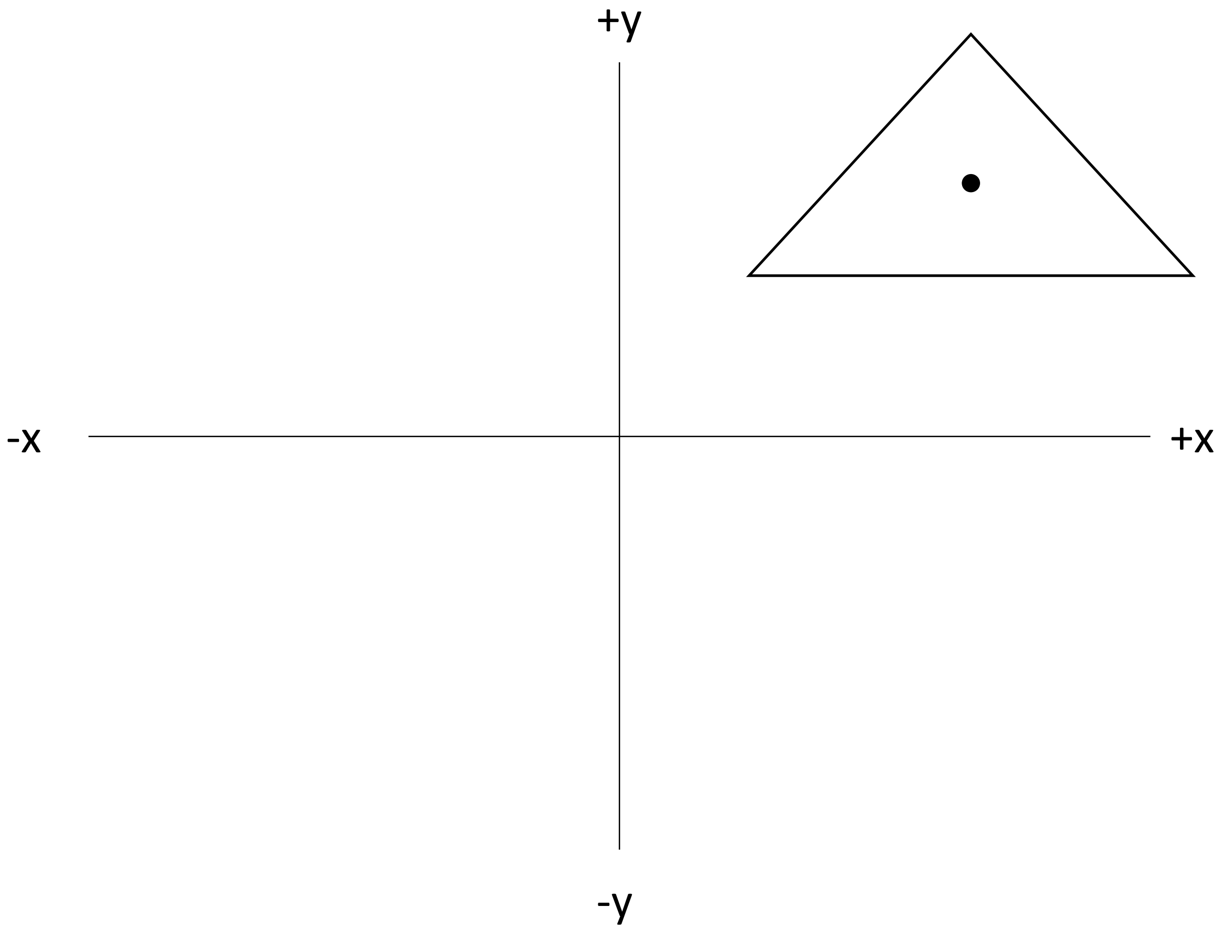
$\Theta = 90°$

$$f\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} x \cdot \cos(\theta) - y \cdot \sin(\theta) \\ x \cdot \sin(\theta) + y \cdot \cos(\theta) \end{bmatrix}$$

$$f(\vec{c}) = \begin{bmatrix} -1 \cdot 0 - 0 \cdot 1 \\ -1 \cdot 1 + 0 \cdot 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

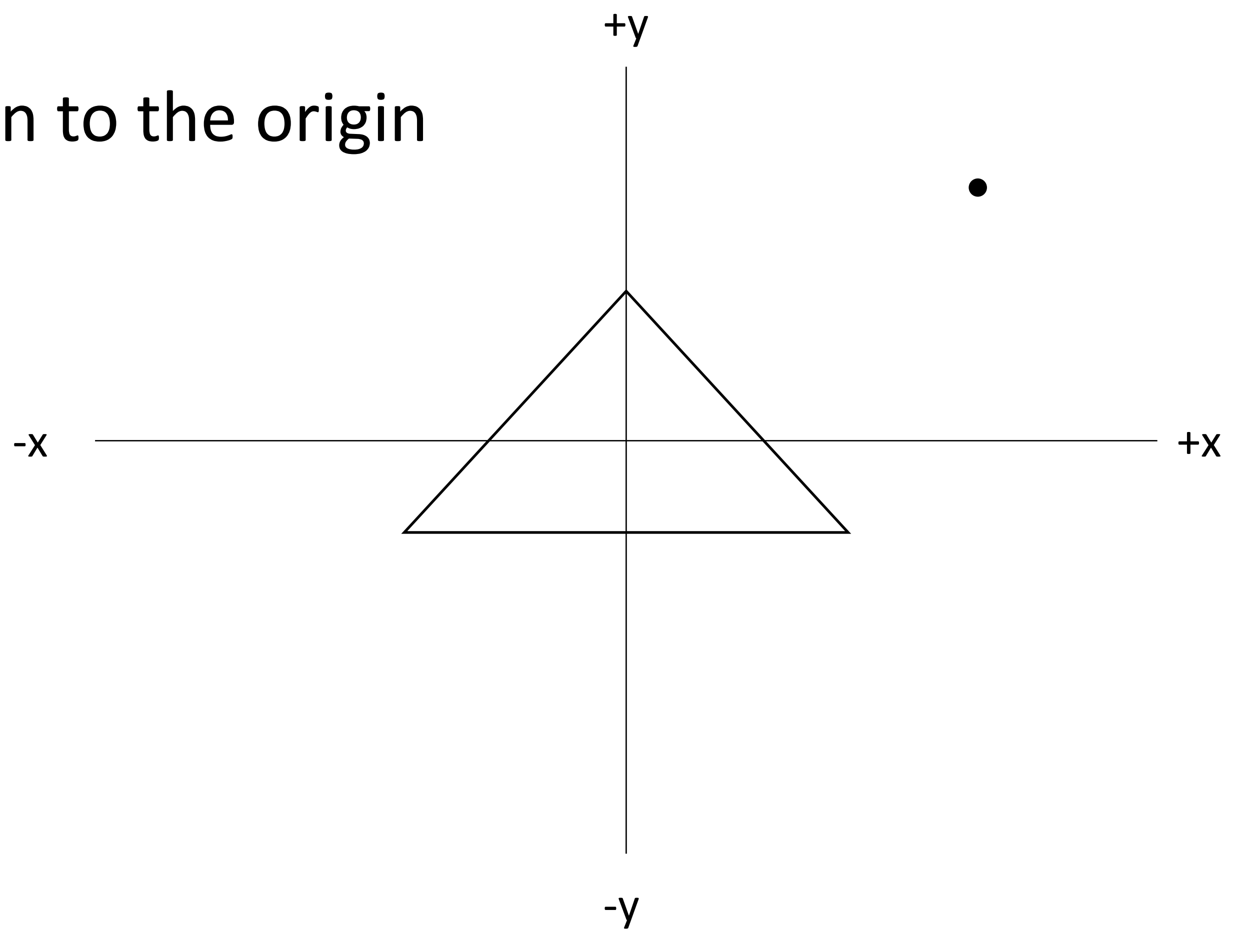


Rotation around the geometric center



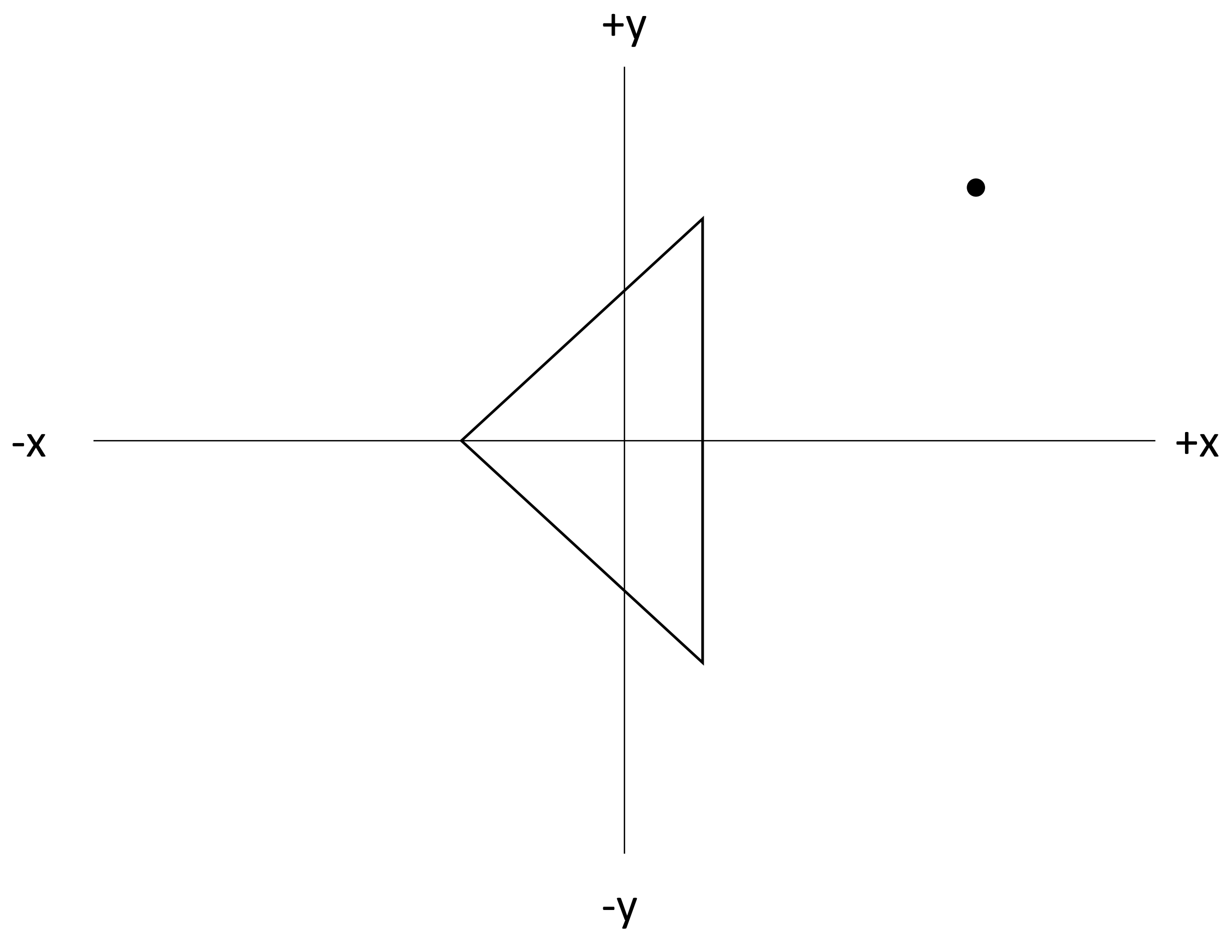
Rotation around the geometric center

Translation to the origin



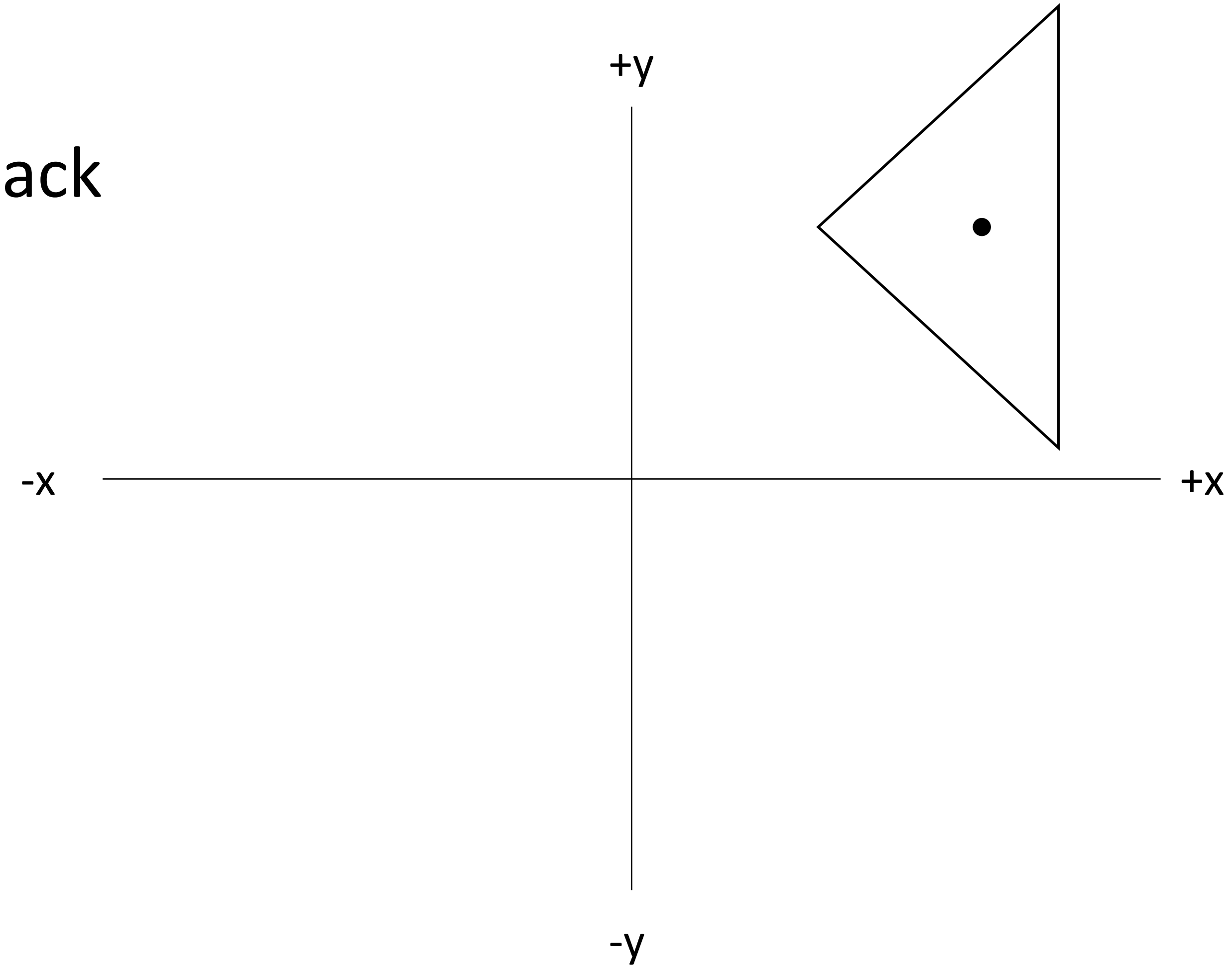
Rotation around the geometric center

Rotation

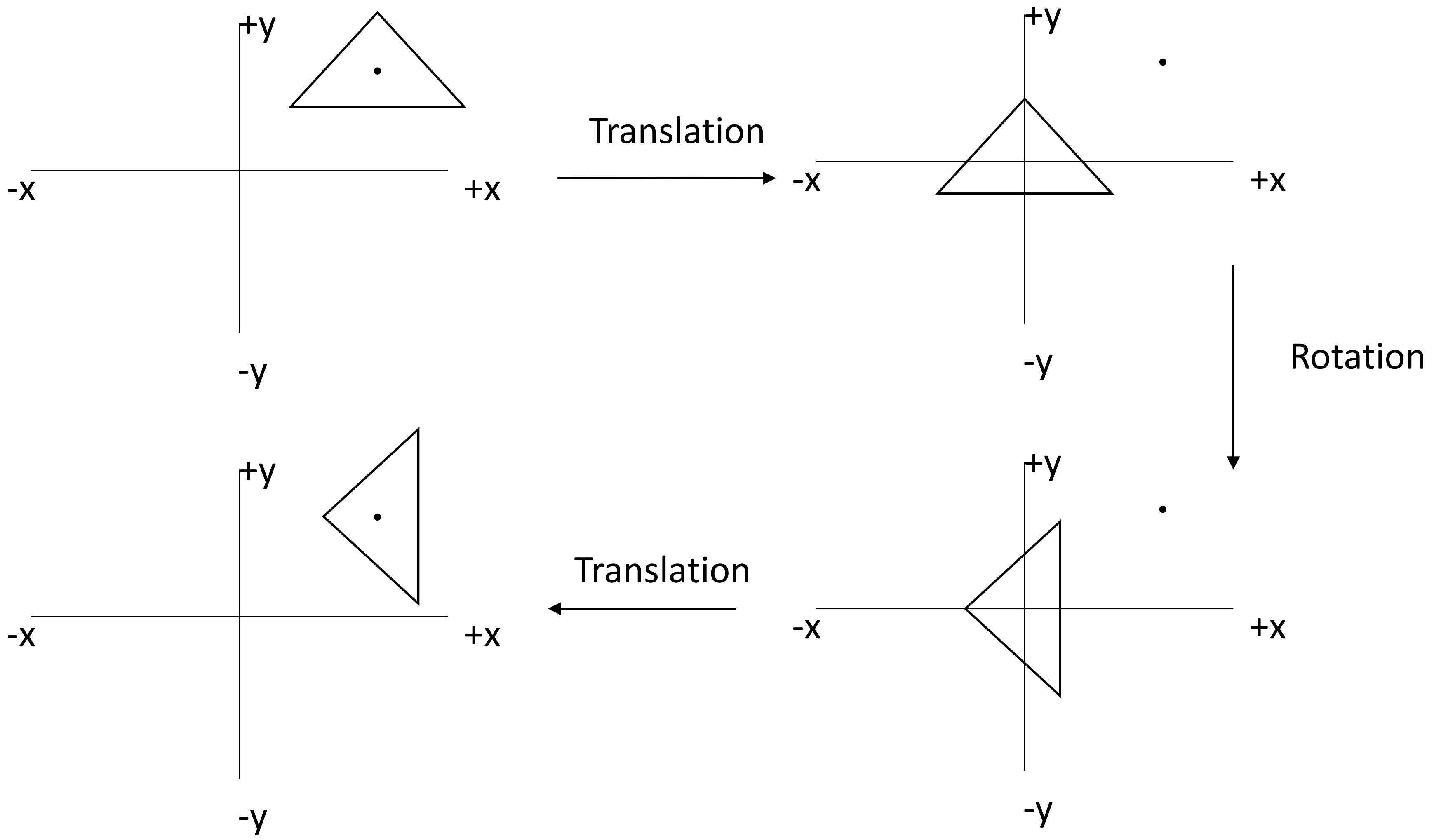


Rotation around the geometric center

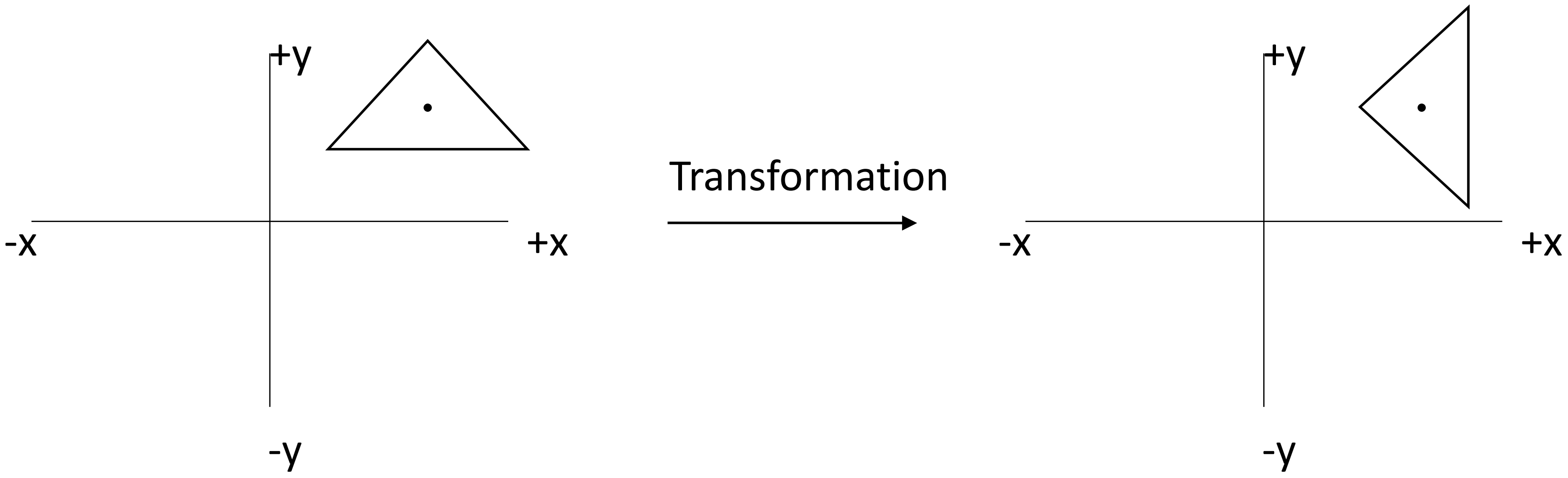
Translation back



How to apply transformations instantaneously



How to apply transformations instantaneously?



Transformation Matrices: Associative

- Let x be a vertex
- A and B transformation matrices and C the product of A and B
- Then $A(Bx) = (AB)x = Cx$

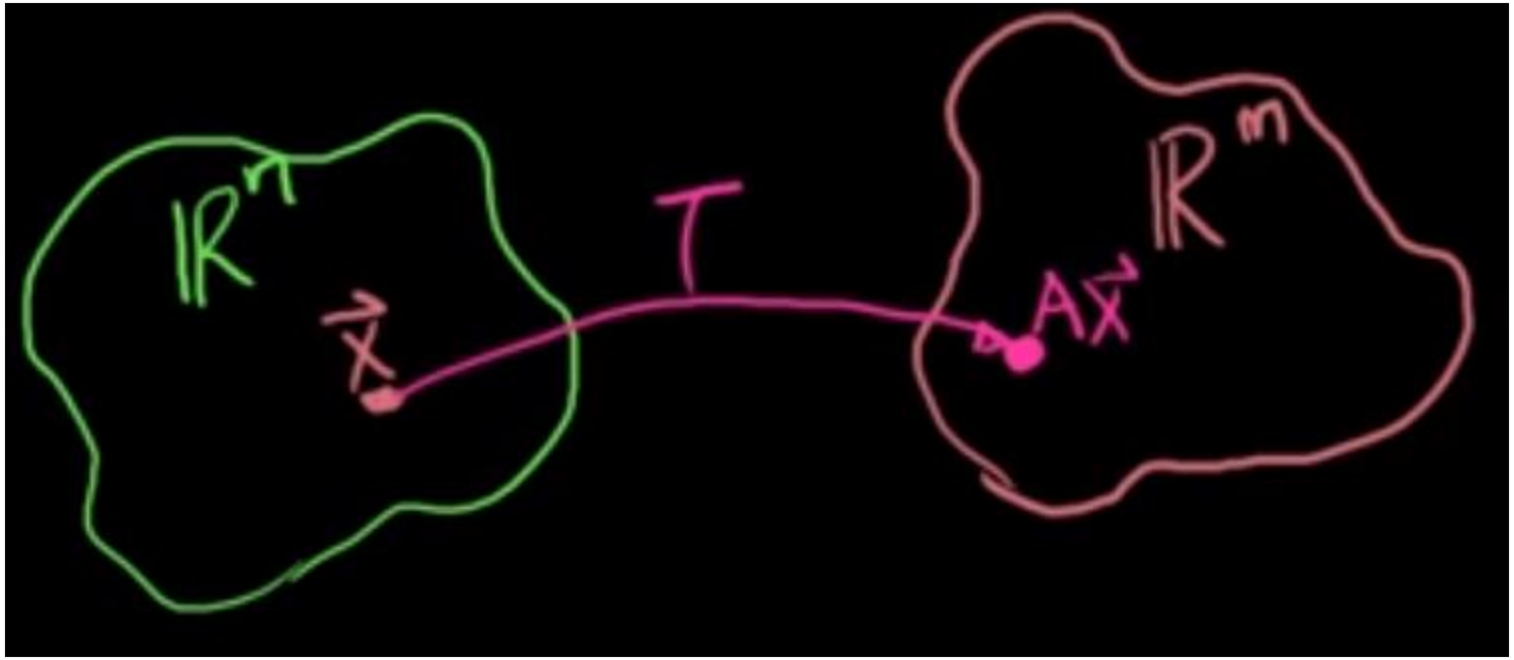
Transformation Matrices: Associative

- Let x be a vertex
- A and B transformation matrices and C the product of A and B
- Then A(Bx) = (AB)x = Cx
  - → applies transformations in a single Matrix-vector multiplication
- If we have a scene composed of millions of vertices this is a significant optimization

Matrix-Vector Product as a Transformation

$T: \mathbb{R}^n \to \mathbb{R}^m$

$T(\vec{x}) = A\vec{x}$



Matrix-Vector Product as a Transformation

$T: \mathbb{R}^2 \to \mathbb{R}^2$

Matrix-Vector Product as a Transformation

$T: \mathbb{R}^2 \to \mathbb{R}^2$

$$B = \begin{bmatrix} 2 & -1 \\ 3 & 4 \end{bmatrix}$$

Matrix-Vector Product as a Transformation

$T: \mathbb{R}^2 \to \mathbb{R}^2$

$$B = \begin{bmatrix} 2 & -1 \\ 3 & 4 \end{bmatrix}$$

$T(\vec{x}) = B\vec{x}$

Matrix-Vector Product as a Transformation

$T: \mathbb{R}^2 \rightarrow \mathbb{R}^2$

$$B = \begin{bmatrix} 2 & -1 \\ 3 & 4 \end{bmatrix}$$

$$T(\vec{x}) = B\vec{x} = \begin{bmatrix} 2 & -1 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Matrix-Vector Product as a Transformation

$T: \mathbb{R}^2 \to \mathbb{R}^2$

$$B = \begin{bmatrix} 2 & -1 \\ 3 & 4 \end{bmatrix}$$

$$T(\vec{x}) = B\vec{x} = \begin{bmatrix} 2 & -1 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2x_1 - x_2 \\ 3x_1 + 4x_2 \end{bmatrix}$$

Matrix-Vector Product as a Transformation

$T: \mathbb{R}^2 \to \mathbb{R}^2$

$$B = \begin{bmatrix} 2 & -1 \\ 3 & 4 \end{bmatrix}$$

$$T(\vec{x}) = B\vec{x} = \begin{bmatrix} 2 & -1 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2x_1 - x_2 \\ 3x_1 + 4x_2 \end{bmatrix}$$

$$T\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} 2x_1 - x_2 \\ 3x_1 + 4x_2 \end{bmatrix}$$

Scaling 2D

Scaling matrix: $\begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$

$$P = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} s_x \cdot x + 0 \cdot y \\ 0 \cdot x + s_y \cdot y \end{bmatrix} = \begin{bmatrix} s_x \cdot x \\ s_y \cdot y \end{bmatrix}$$

Rotation 2D

Rotation matrix: $\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$

$$P = \begin{bmatrix} \cos(\Pi/4) & -\sin(\Pi/4) \\ \sin(\Pi/4) & \cos(\Pi/4) \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \cdot 0 - 0 \cdot 1 \\ 1 \cdot 1 + 0 \cdot 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Concatenation of transformation matrices

$$P' = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Concatenation of transformation matrices

$$P' = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} s_x x \\ s_y y \end{bmatrix}$$

Concatenation of transformation matrices

$$P' = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} s_x x \\ s_y y \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

Concatenation of transformation matrices

$$P' = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} s_x x \\ s_y y \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

$$P'' = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix}$$

Concatenation of transformation matrices

$$P' = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} s_x x \\ s_y y \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

$$P'' = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\theta) s_x\, x - \sin(\theta) s_y y \\ \sin(\theta)\, s_x\, x + \cos(\theta) s_y y \end{bmatrix}$$

Concatenation of transformation matrices

$$P' = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} s_x x \\ s_y y \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

$$P'' = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\theta) s_x\, x - \sin(\theta) s_y y \\ \sin(\theta)\, s_x\, x + \cos(\theta) s_y y \end{bmatrix}$$

This is identical to (associative multiplication of matrices)

$$P'' = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Concatenation of transformation matrices

$$P' = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} s_x x \\ s_y y \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

$$P'' = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\theta) s_x\, x - \sin(\theta) s_y y \\ \sin(\theta)\, s_x\, x + \cos(\theta) s_y y \end{bmatrix}$$

This is identical to (associative multiplication of matrices)

$$P'' = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos(\theta) s_x\, x - \sin(\theta) s_y y \\ \sin(\theta)\, s_x\, x + \cos(\theta) s_y y \end{bmatrix}$$

Translation

$$f\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} x + T_x \\ y + T_y \end{bmatrix}$$

Translation

$$f\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} x + T_x \\ y + T_y \end{bmatrix} \rightarrow$$

it is impossible to express such a transformation with 2D matrix multiplications

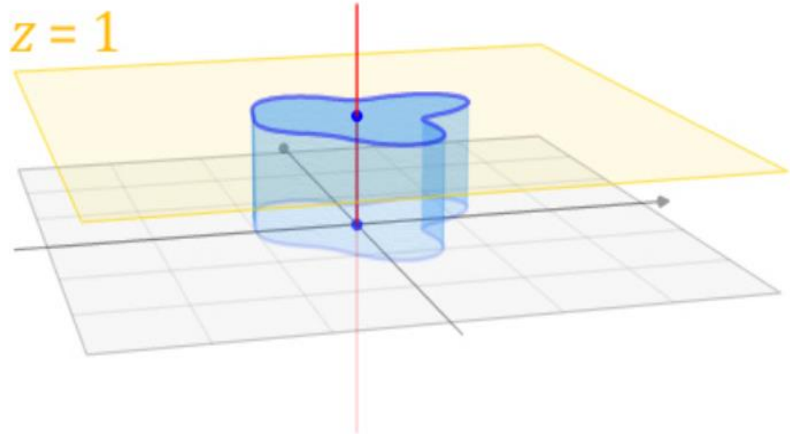
Translation

$$f\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} x + T_x \\ y + T_y \end{bmatrix}$$ → it is impossible to express such a transformation with 2D matrix multiplications

Hence, we embed 2D space in 3D where the third coordinate will be equal to 1. Our 2D space resides in the z = 1 plane.

$$\begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Geometric interpretation of 2D translation



Translation T(T_x, T_y)

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Translation T(T_x, T_y)

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Resulting in:

$$x' = x + 1 \cdot T_x$$
$$y' = y + 1 \cdot T_y$$
$$1 = 1$$

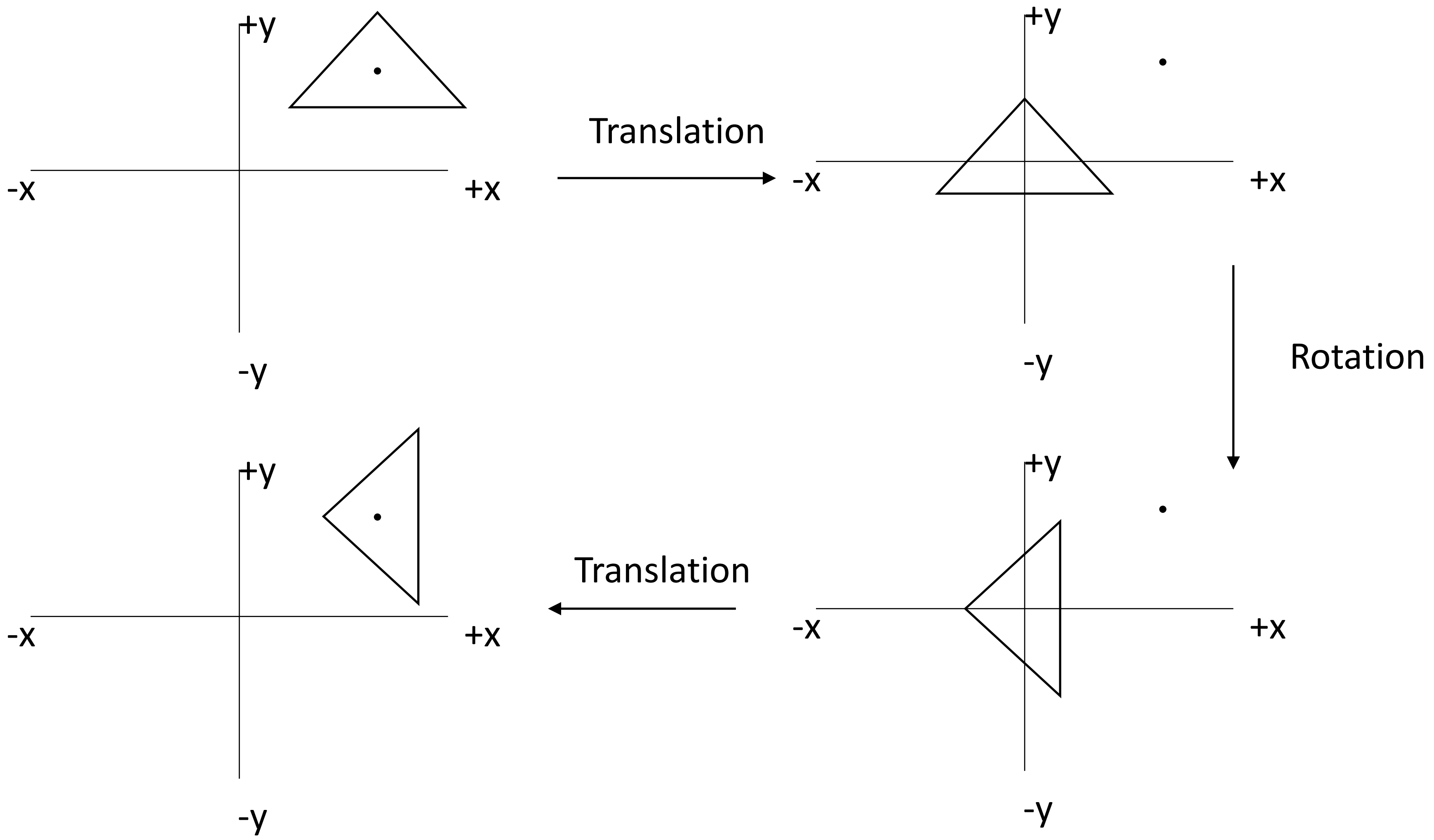
Scaling, Rotation and Translation in 2D

$$S(s_x, s_y) = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T(T_x, T_y) = \begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix}$$

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

How to apply transformations instantaneously



Rotation matrix M around $P_0(x_0, y_0)$

Rotation matrix M around $P_0(x_0, y_0)$

1. Translation to origin: $T(-x_0, -y_0)$

Rotation matrix M around $P_0(x_0, y_0)$

1. Translation to origin: $T(-x_0, -y_0)$
2. Rotation with angle θ: $R(\theta)$

Rotation matrix M around $P_0(x_0, y_0)$

1. Translation to origin: $T(-x_0, -y_0)$
2. Rotation with angle θ: $R(\theta)$
3. Translation to point P_0: $T(x_0, y_0)$

Rotation matrix M around $P_0(x_0, y_0)$

1. Translation to origin: $T(-x_0, -y_0)$
2. Rotation with angle θ: $R(\theta)$
3. Translation to point P_0: $T(x_0, y_0)$

$$M = T(x_0, y_0)\, R(\theta)\, T(-x_0, -y_0)$$

Rotation matrix M around $P_0(x_0, y_0)$

1. Translation to origin: $T(-x_0, -y_0)$
2. Rotation with angle θ: $R(\theta)$
3. Translation to point P_0: $T(x_0, y_0)$

$$M = T(x_0, y_0)\, R(\theta)\, T(-x_0, -y_0)$$

$$M = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Rotation matrix M around $P_0(x_0, y_0)$

1. Translation to origin: $T(-x_0, -y_0)$
2. Rotation with angle θ: $R(\theta)$
3. Translation to point P_0: $T(x_0, y_0)$

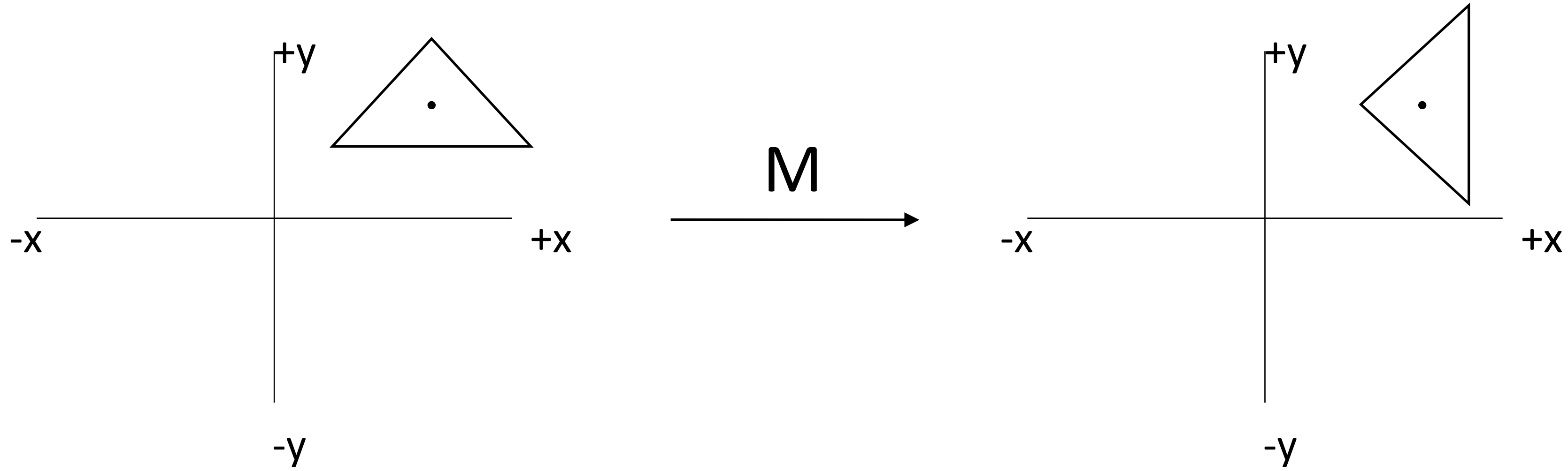
$$M = T(x_0, y_0)\, R(\theta)\, T(-x_0, -y_0)$$

$$M = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & -\cos(\theta)\, x_0 + \sin(\theta) y_0 + x_0 \\ \sin(\theta) & \cos(\theta) & \sin(\theta)\, x_0 - \cos(\theta) y_0 + y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

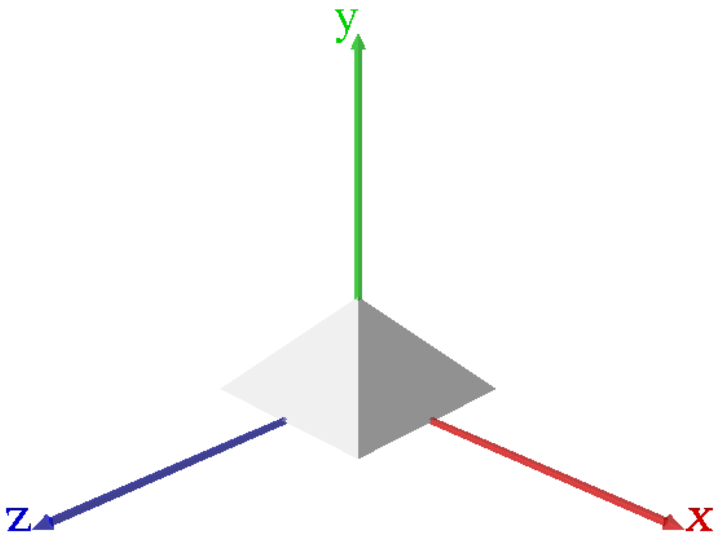
Rotation matrix M around $P_0(x_0, y_0)$

1. Translation to origin: $T(-x_0, -y_0)$
2. Rotation with angle θ: $R(\theta)$
3. Translation to point P_0: $T(x_0, y_0)$

$$M = T(x_0, y_0)\, R(\theta)\, T(-x_0, -y_0)$$

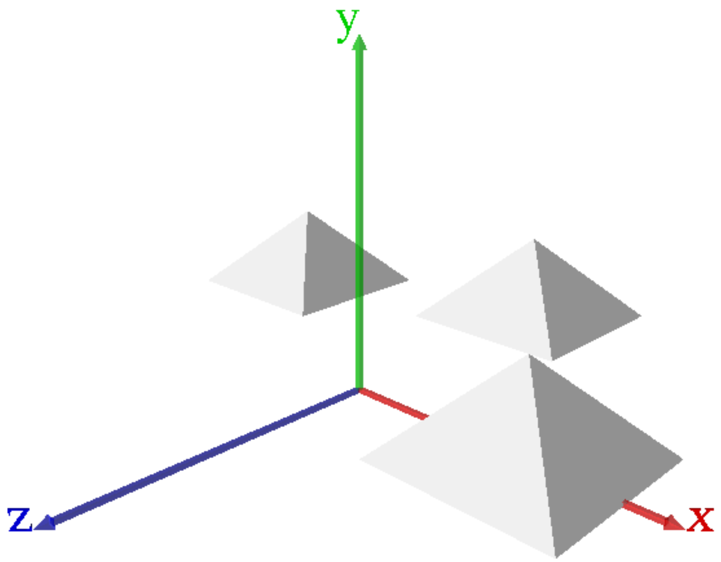


World space transformation



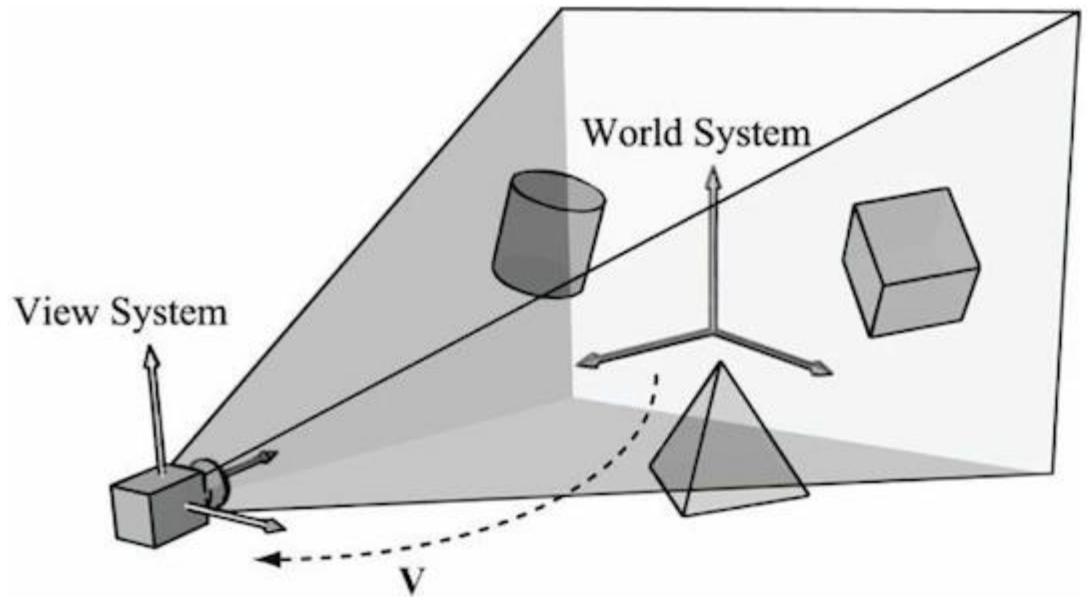
Object Space

Model Matrix



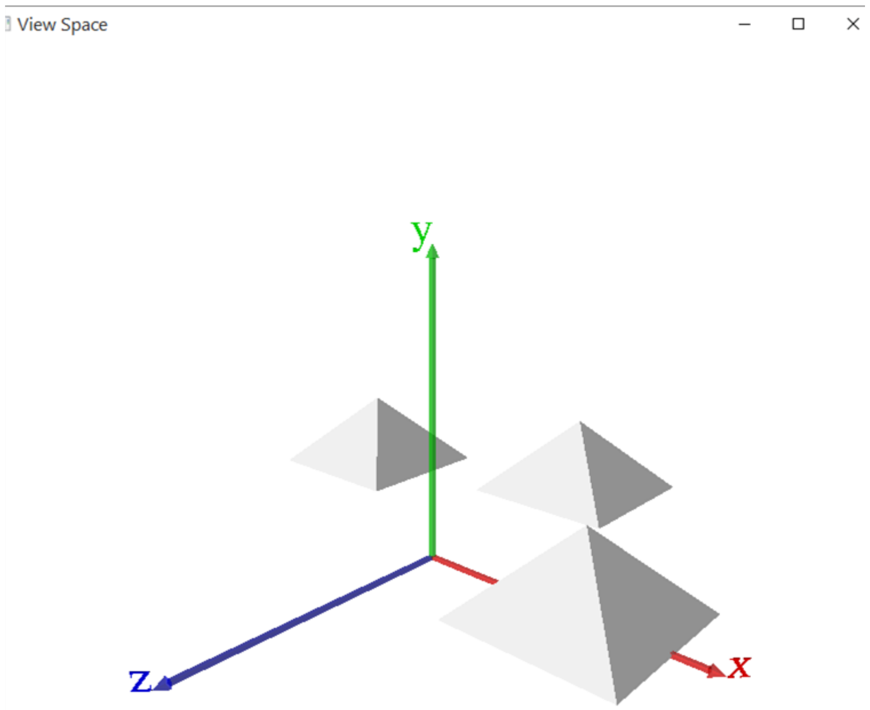
World Space

View (eye) space transformation

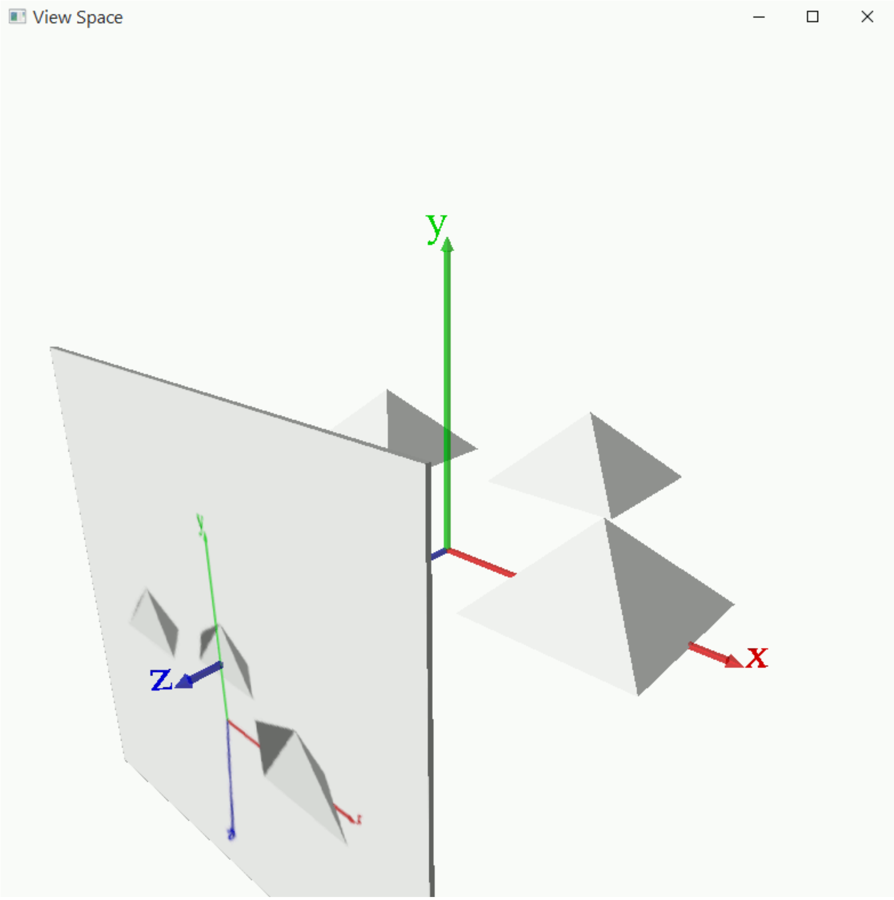


Define a “**view frustum**” that contains all visible objects

View (eye) space transformation

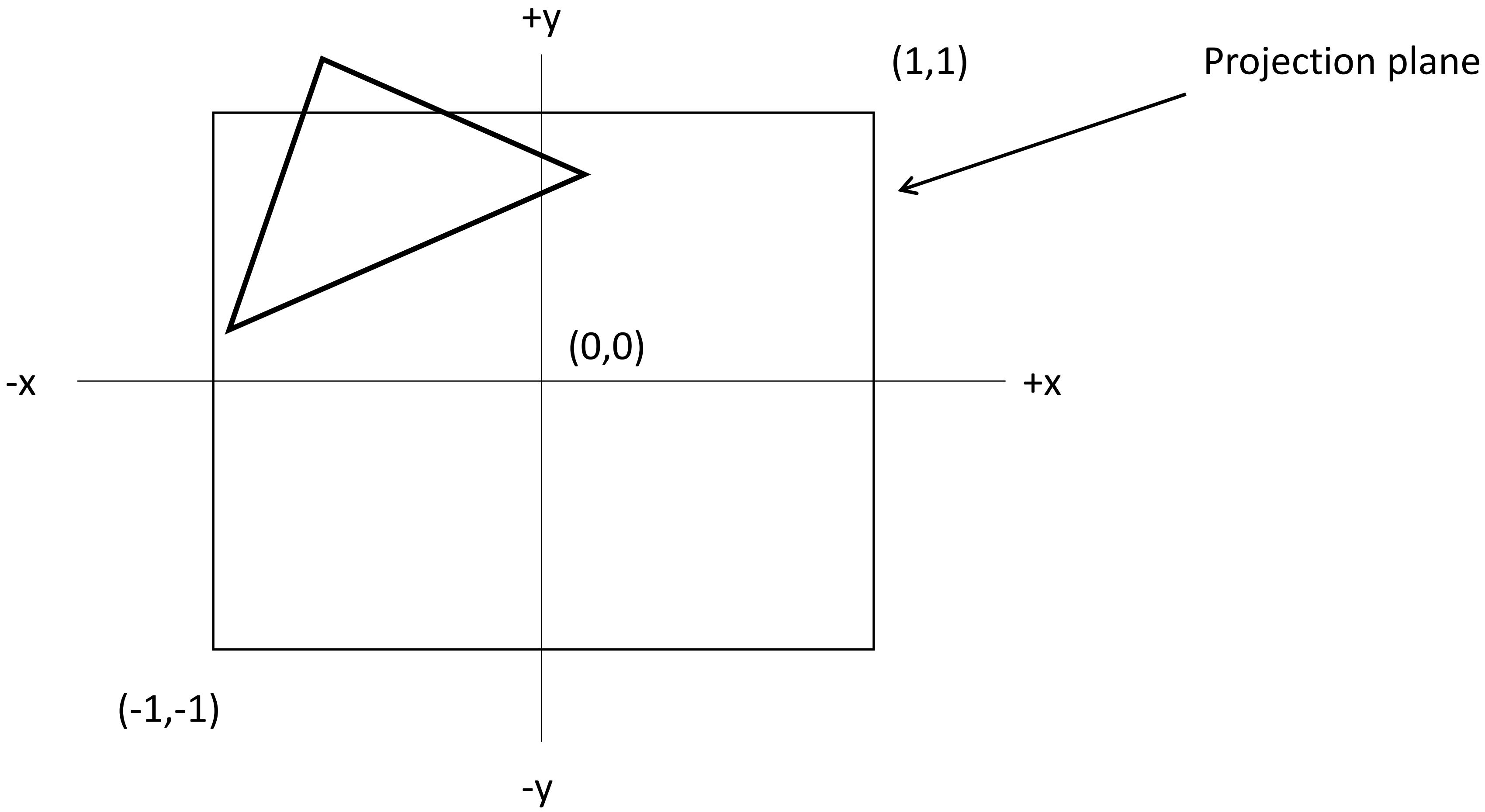


Projection transformation

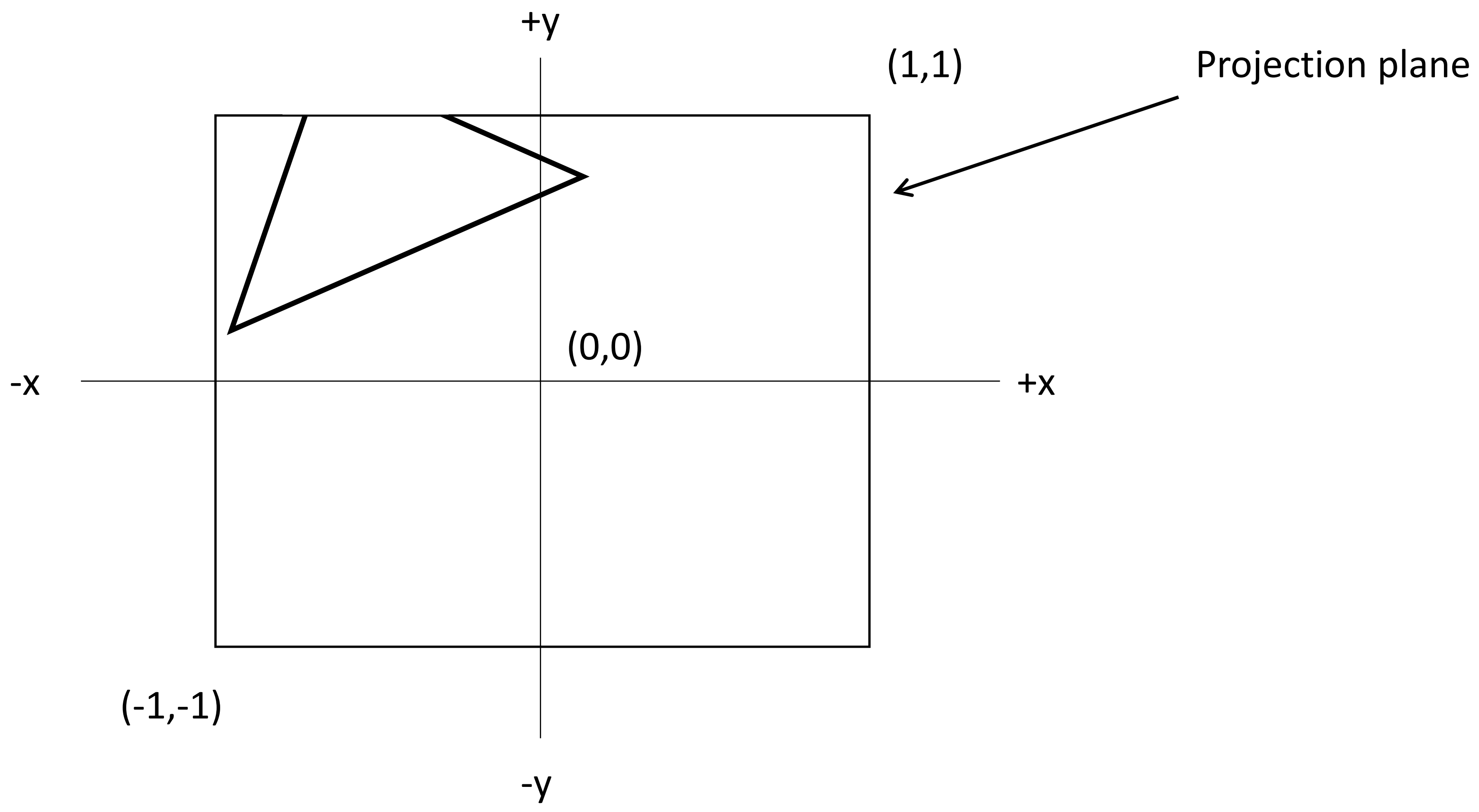


Project scene inside the view frustum onto a “**projection plane**”

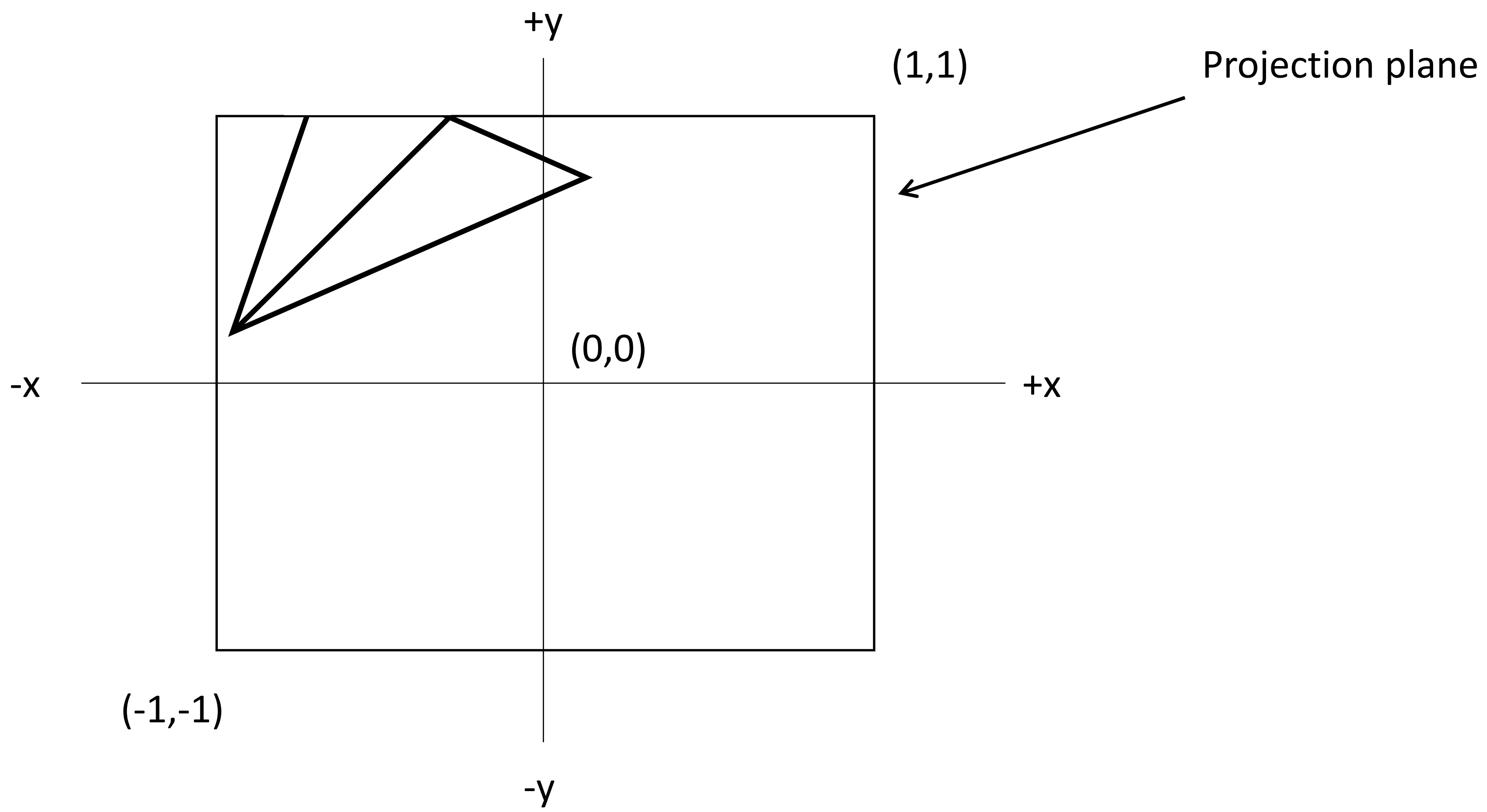
Clipping



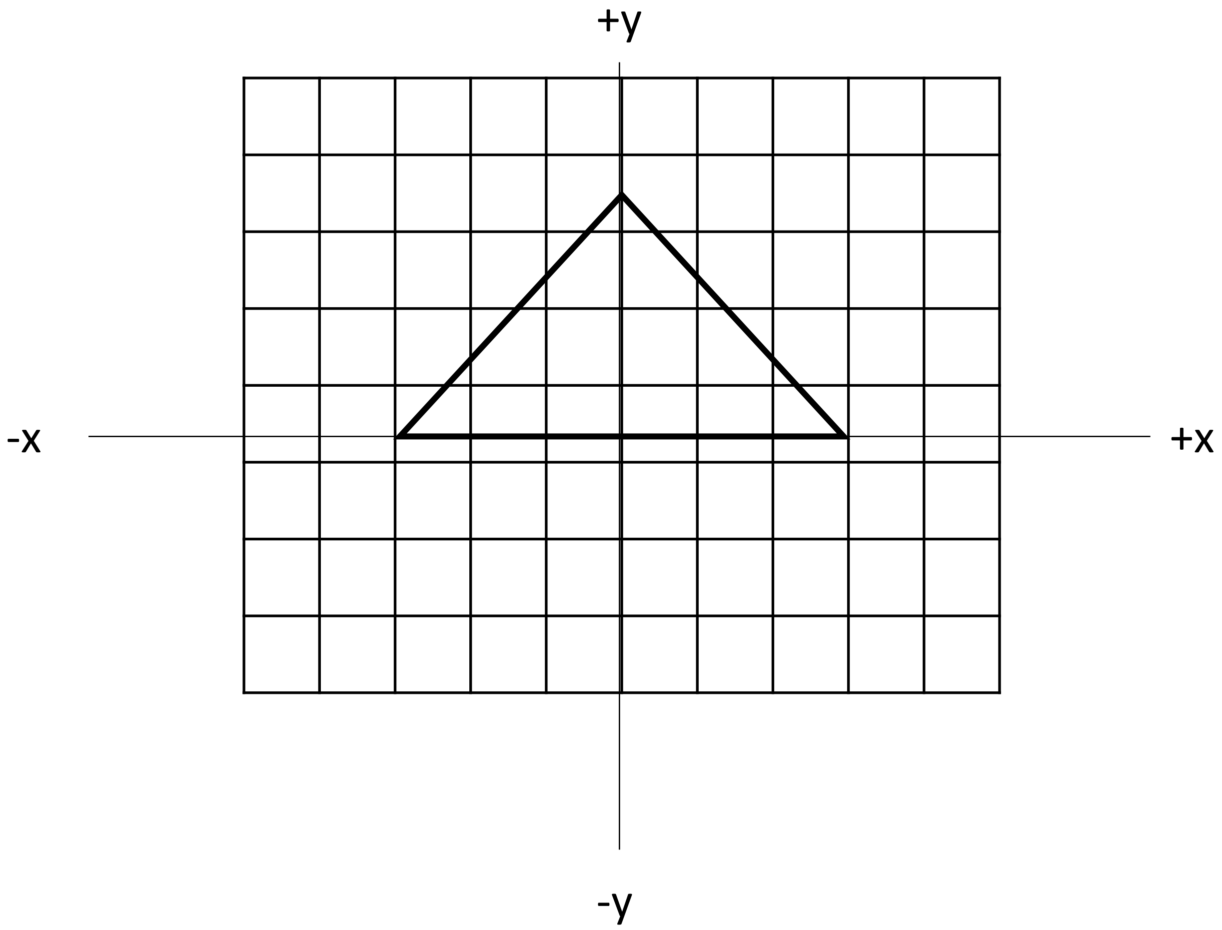
Clipping



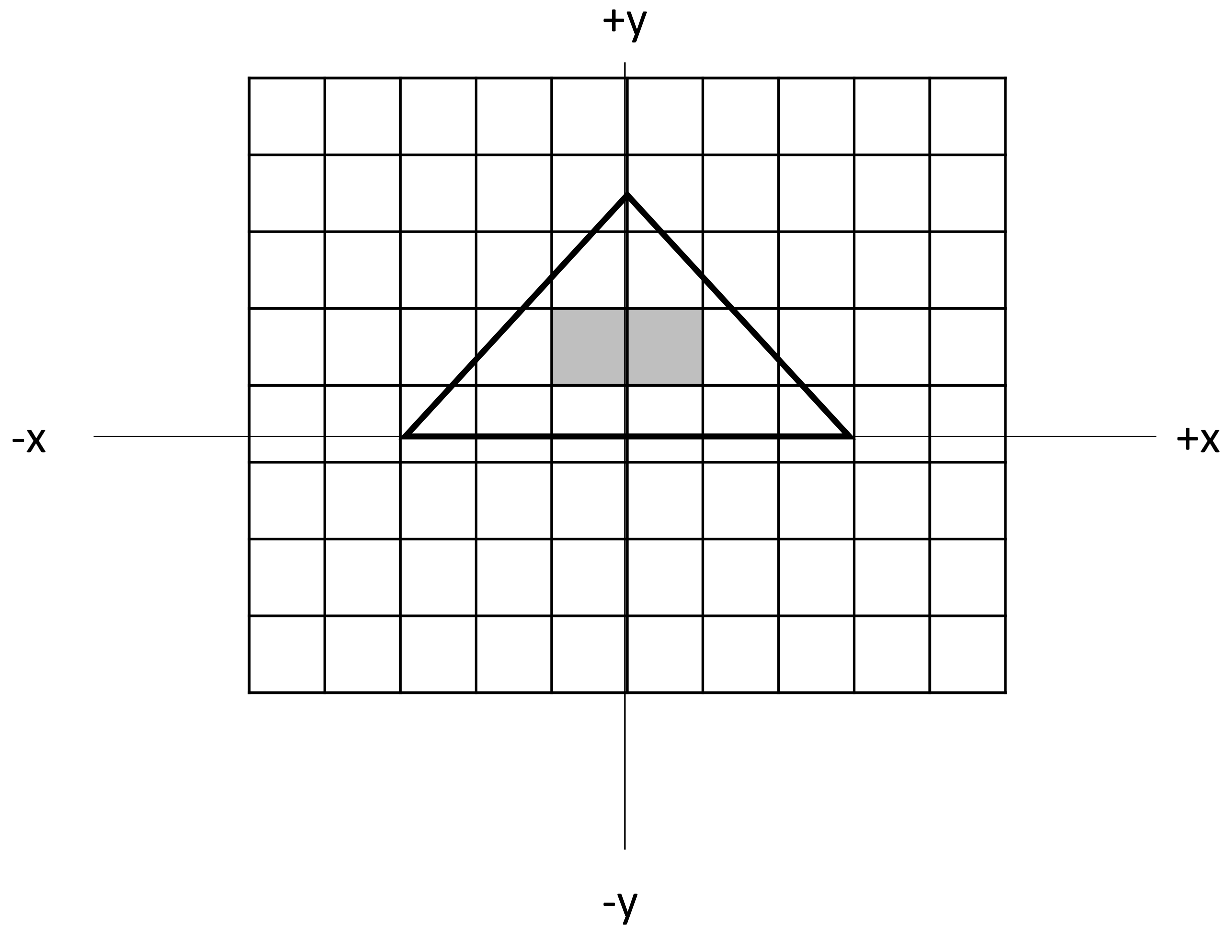
Clipping



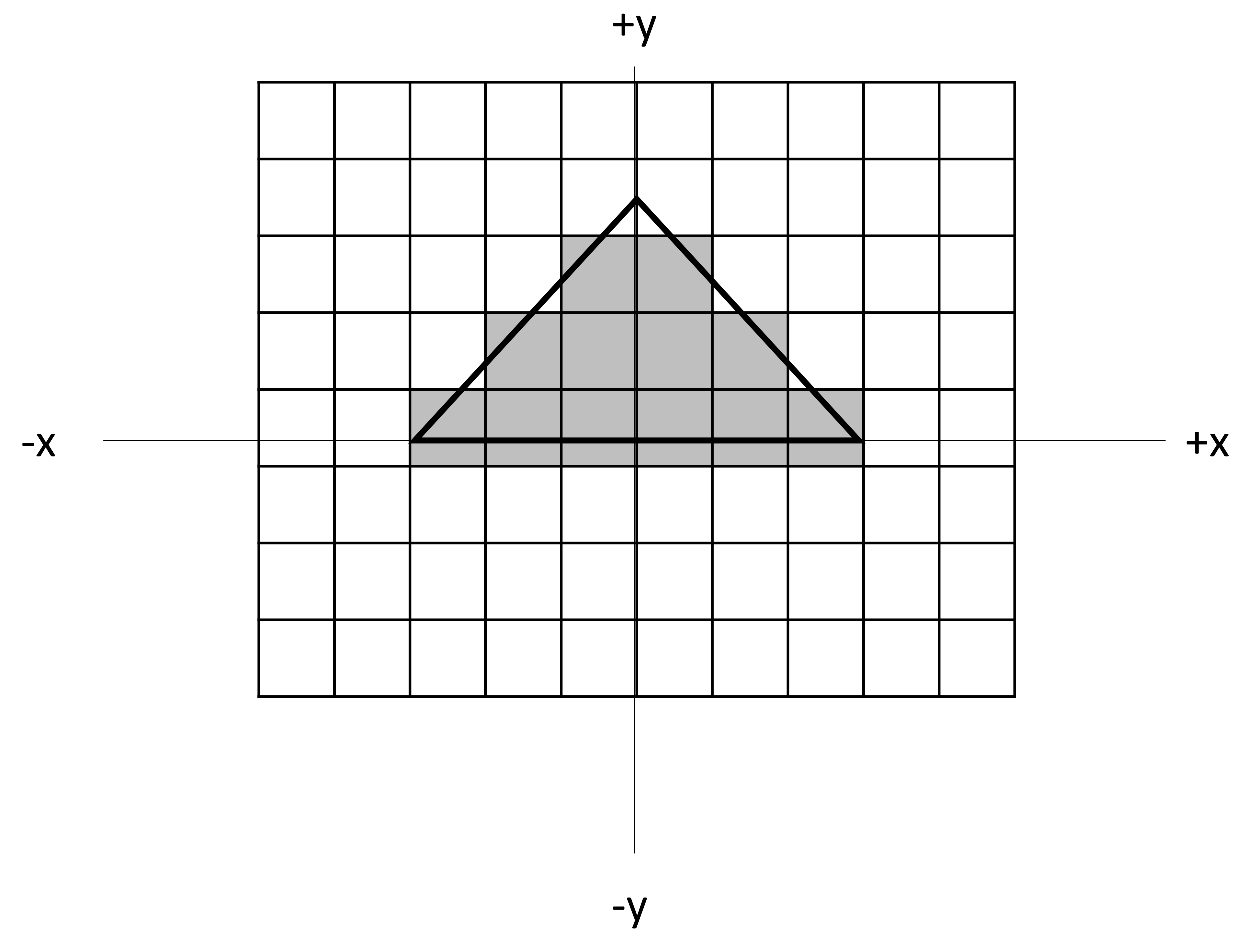
Scan conversion or rasterization



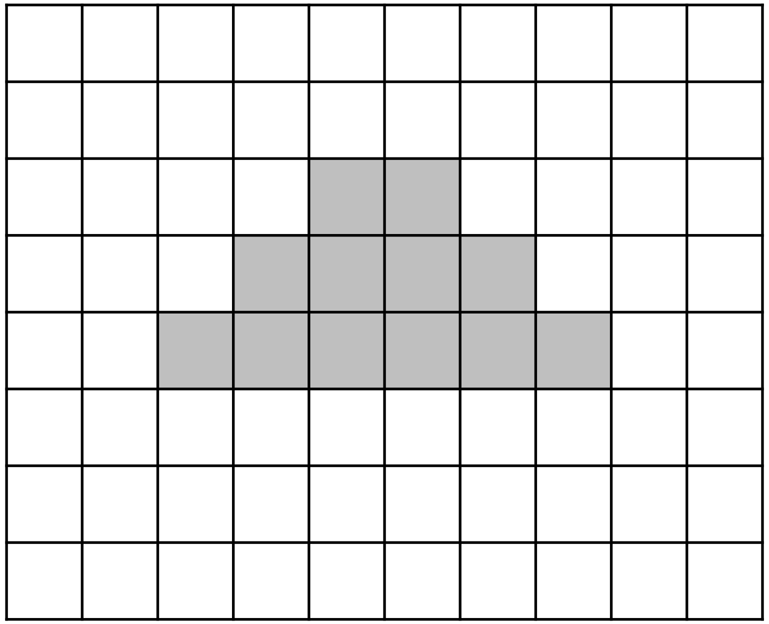
Scan conversion or rasterization



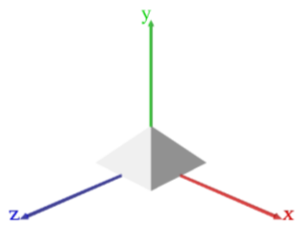
Scan conversion or rasterization



Scan conversion or rasterization



Simplified Rendering Pipeline

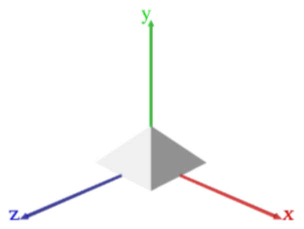


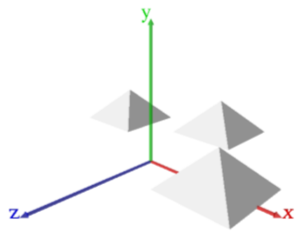
Object Space

Simplified Rendering Pipeline

Model Matrix

→



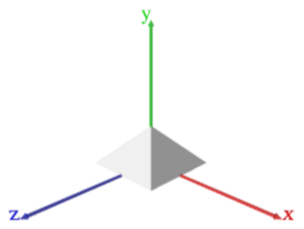


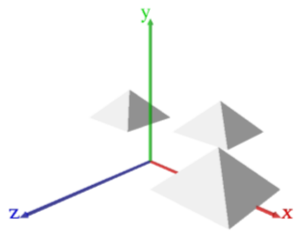
Object Space

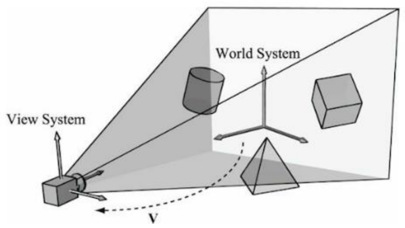
World Space

Simplified Rendering Pipeline

Model Matrix → View/Camera Matrix →



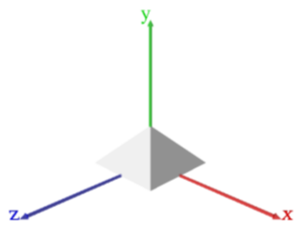




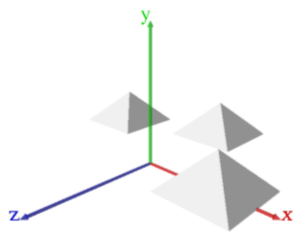
Object Space World Space View Space

Simplified Rendering Pipeline

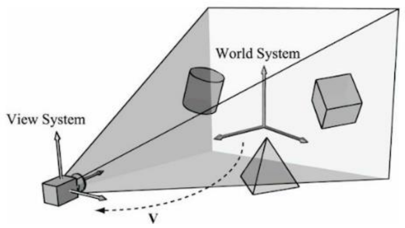
Model Matrix → View/Camera Matrix → Projection Matrix →



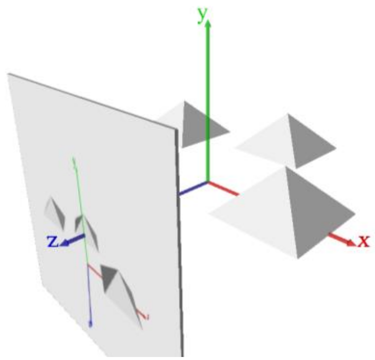
Object Space



World Space



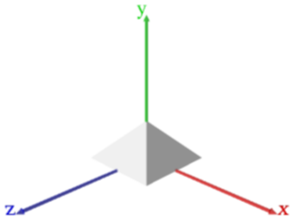
View Space

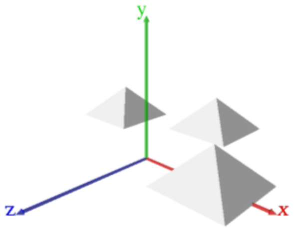


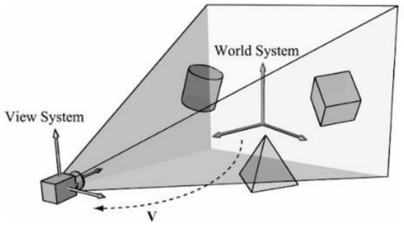
Clip Space

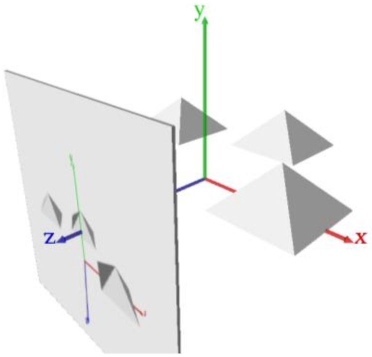
Simplified Rendering Pipeline

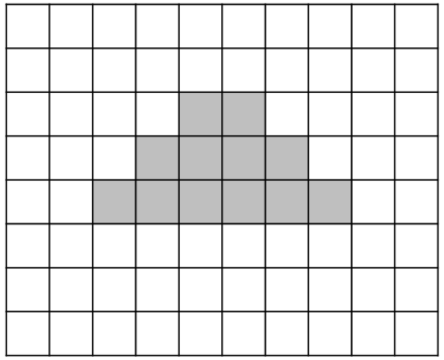
Model Matrix → View/Camera Matrix → Projection Matrix → Viewport Transform →









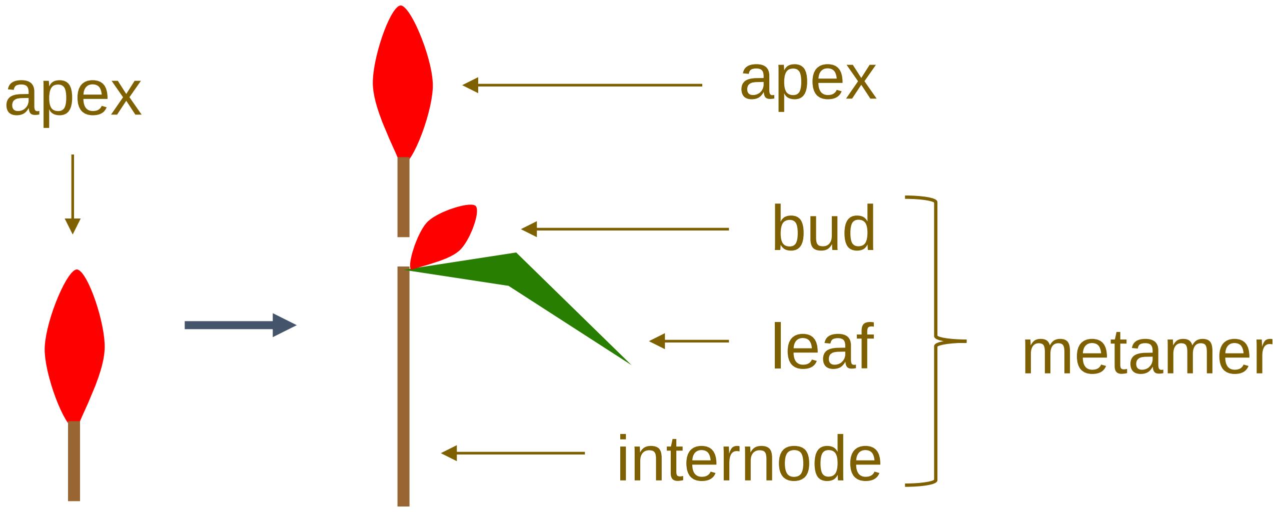


Object Space | World Space | View Space | Clip Space | Screen/Window Space

Mango tree growth

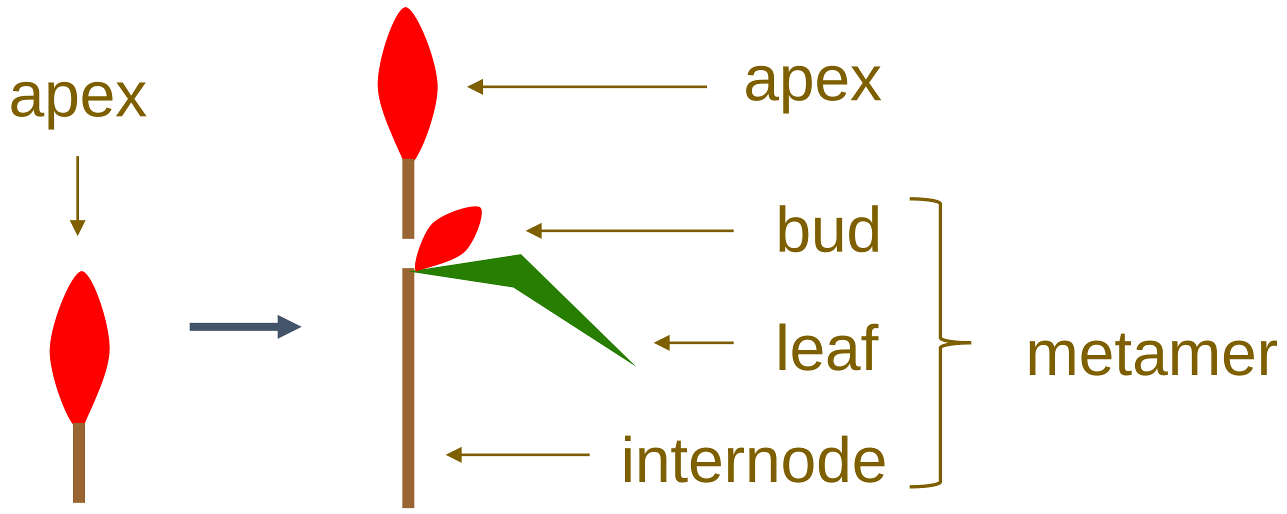
[Grechi et al., VPlants team, CIRAD, Montpellier]

The fundamental developmental scheme

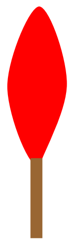


rule

The fundamental developmental scheme

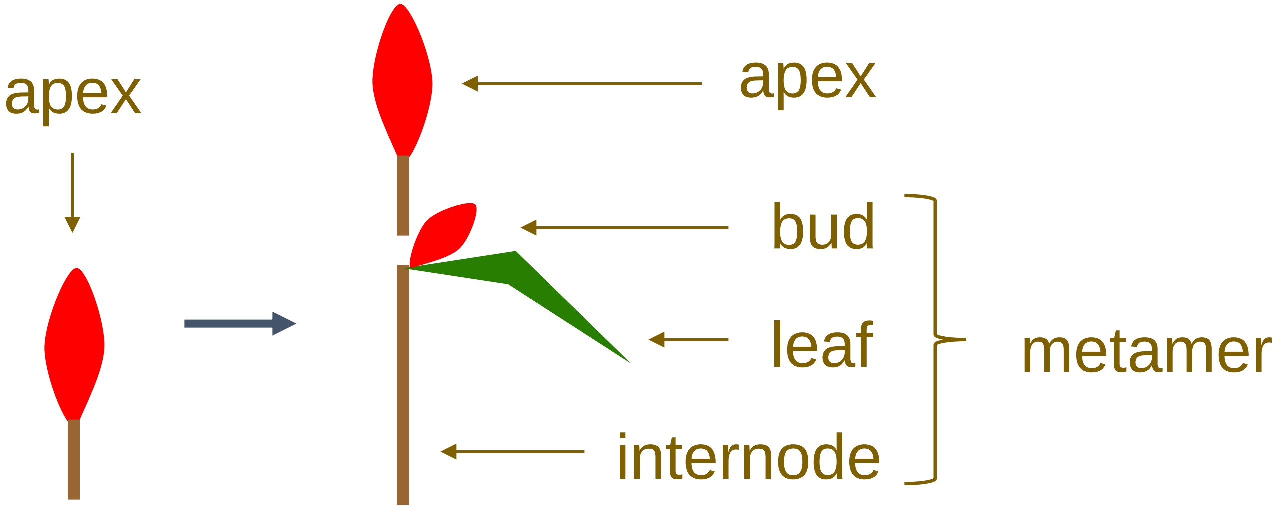


rule

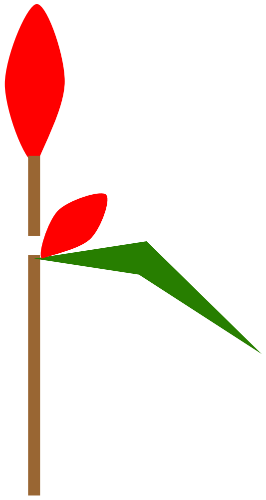


development

The fundamental developmental scheme

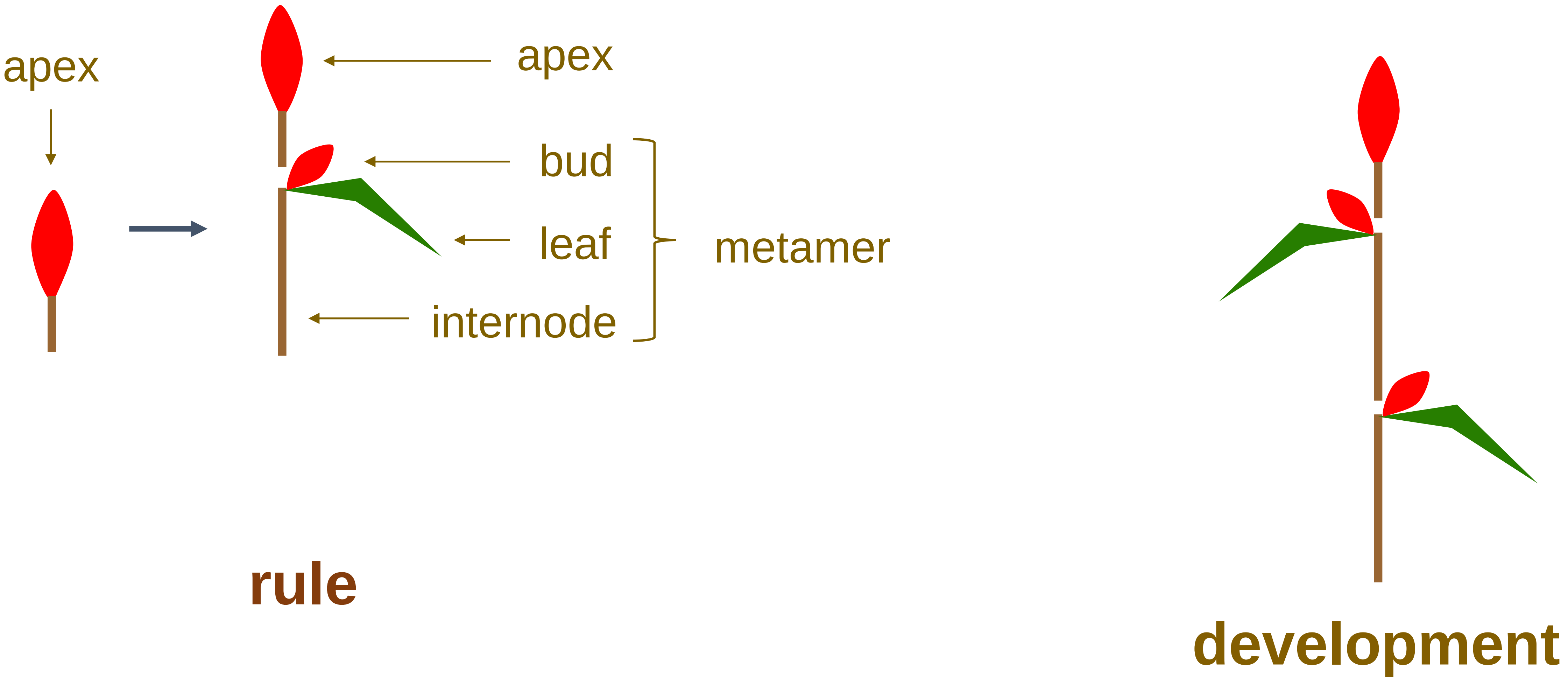


rule

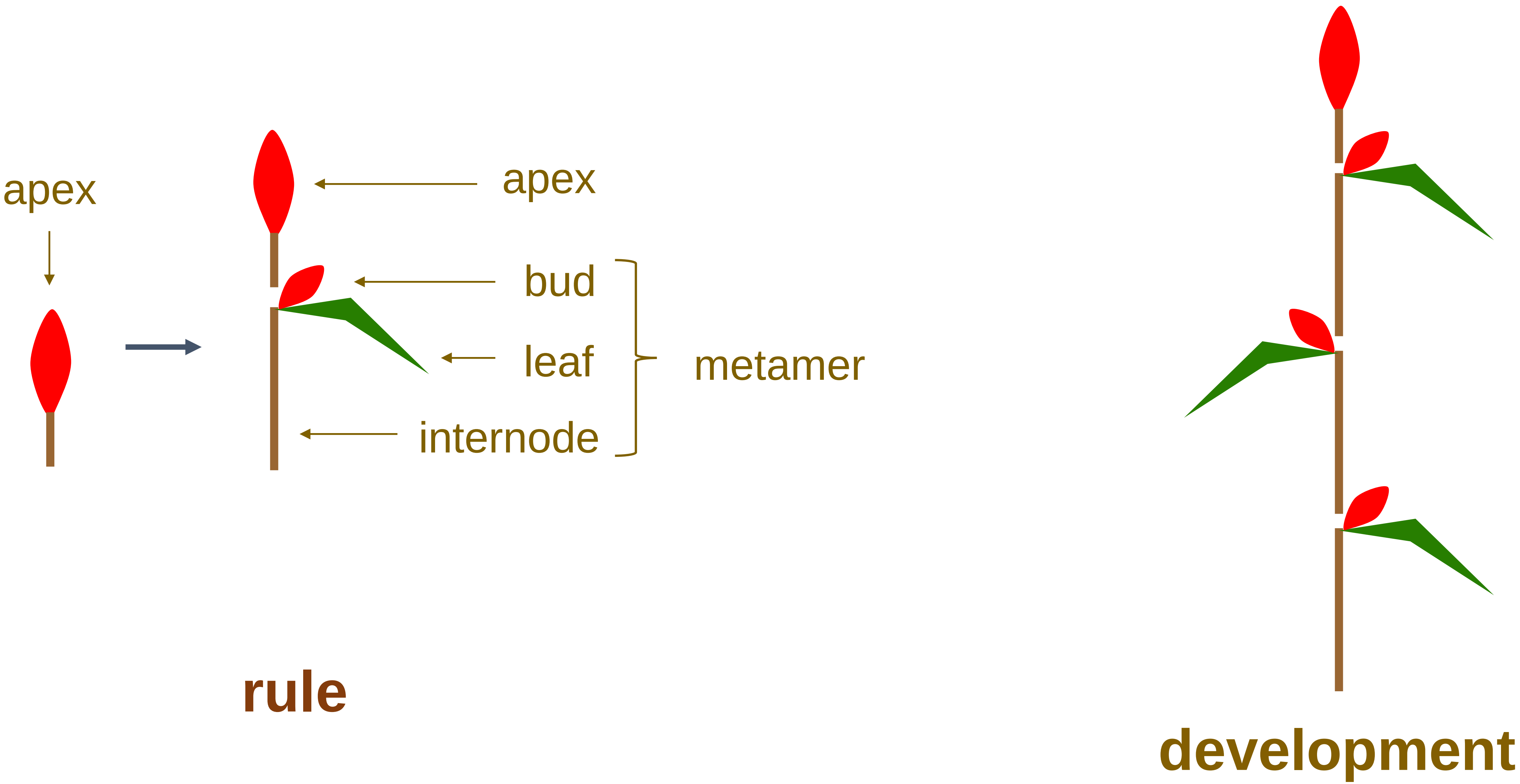


development

The fundamental developmental scheme

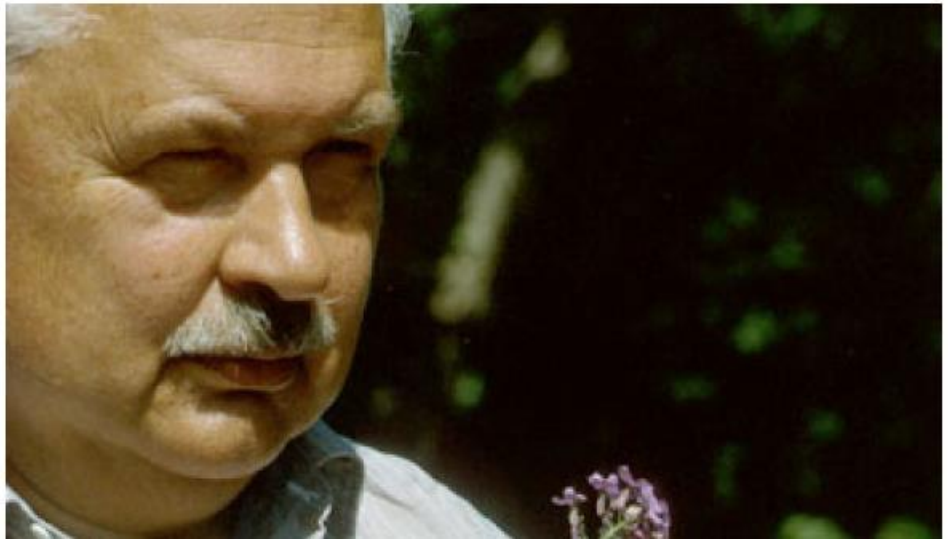


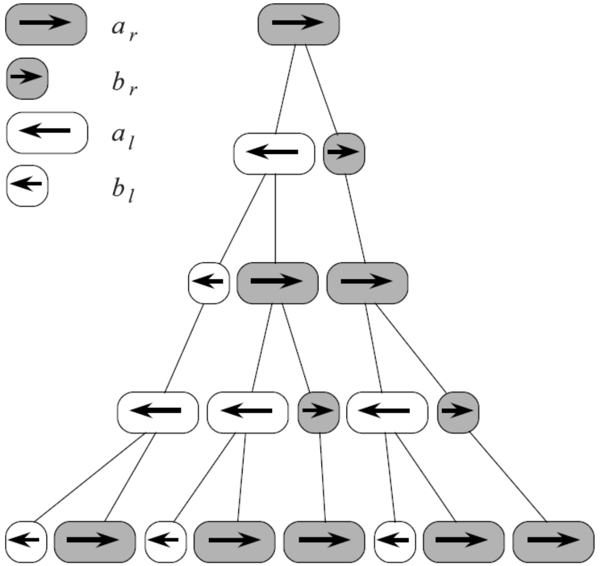
The fundamental developmental scheme

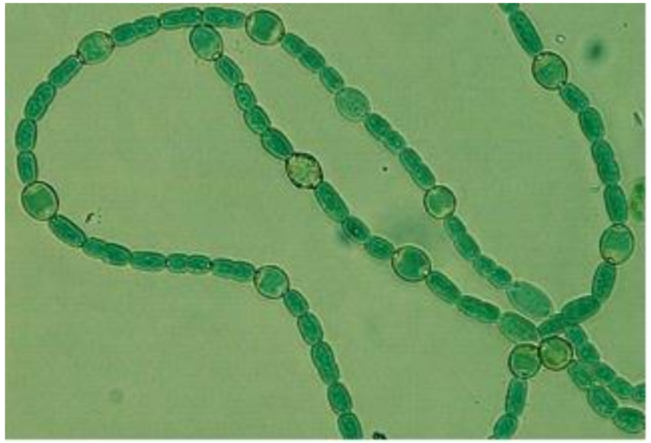


L-Systemy: język do modelowania wzrostu

- Aristid Lindenmayer (1925-1989)
  - Anabaena Catenula
  - 1968 Lindenmayer systems – parallel string rewriting systems







L-Systemy: język do modelowania wzrostu

- Później zastosowano do wzrostu roślin

Przemysław Prusinkiewicz

D0L-Systems

- Formal grammar $G = (V,\ \omega,\ P)$
- V → alphabet of symbols containing elements that can be replaced
- ω → axiom (the initial state of the system)
- P → a set of production rules defining the way symbols can be replaced

- Example:
  - $V = (A, M)$
  - $\omega = A$
  - $P = [(A \rightarrow MA), (M \rightarrow M)]$

D0L-Systems

- Formal grammar $G = (V,\ \omega,\ P)$
- V → alphabet of symbols containing elements that can be replaced
- ω → axiom (the initial state of the system)
- P → a set of production rules defining the way symbols can be replaced

- Example:
  - $V = (A, M)$
  - $\omega = A$
  - $P = [(A \rightarrow MA), (M \rightarrow M)]$

n=0: A

D0L-Systems

- Formal grammar $G = (V,\ \omega,\ P)$
- V → alphabet of symbols containing elements that can be replaced
- ω → axiom (the initial state of the system)
- P → a set of production rules defining the way symbols can be replaced

- Example:
  - $V = (A, M)$
  - $\omega = A$
  - $P = [(A \rightarrow MA), (M \rightarrow M)]$

n=0: A
n=1: MA

D0L-Systems

- Formal grammar $G = (V,\ \omega,\ P)$
- V → alphabet of symbols containing elements that can be replaced
- ω → axiom (the initial state of the system)
- P → a set of production rules defining the way symbols can be replaced

- Example:
  - $V = (A, M)$
  - $\omega = A$
  - $P = [(A \rightarrow MA), (M \rightarrow M)]$

n=0: A
n=1: MA
n=2: MMA

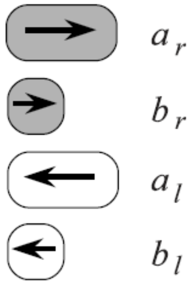
D0L-Systems

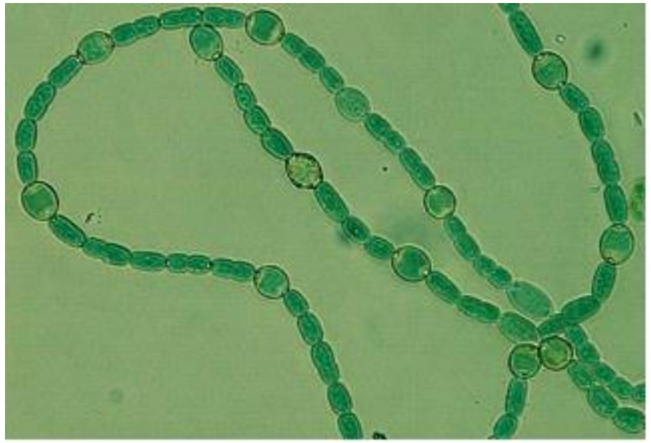
- Formal grammar $G = (V,\ \omega,\ P)$
- V → alphabet of symbols containing elements that can be replaced
- ω → axiom (the initial state of the system)
- P → a set of production rules defining the way symbols can be replaced

- Example:
  - $V = (A, M)$
  - $\omega = A$
  - $P = [(A \rightarrow MA), (M \rightarrow M)]$

n=0: A
n=1: MA
n=2: MMA
n=3: MMMA

Anabaena Catenula: Model 1

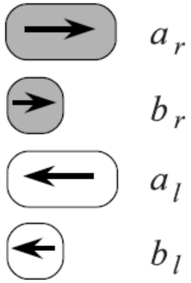


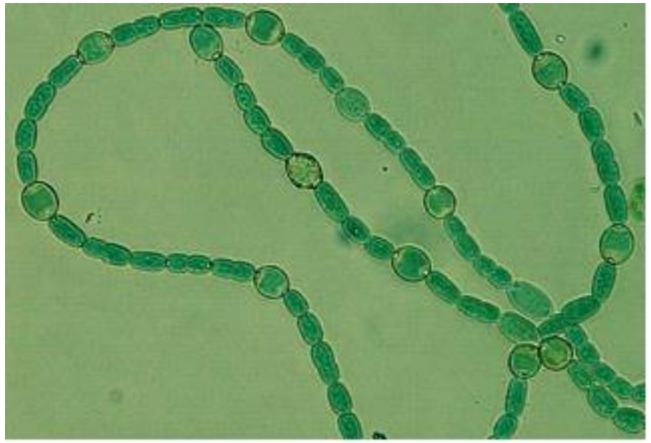


Anabaena Catenula: Model 1

$$p_1 : a_r \rightarrow a_l b_r$$

$$p_2 : a_l \rightarrow b_l a_r$$





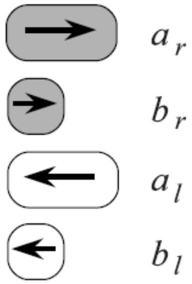
Anabaena Catenula: Model 1

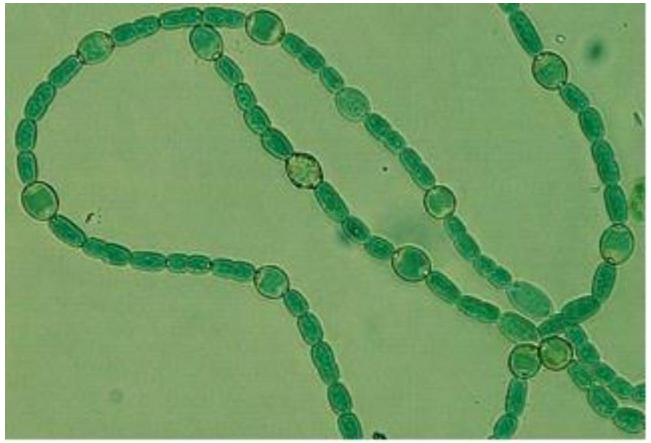
$$p_1 : a_r \rightarrow a_l b_r$$

$$p_2 : a_l \rightarrow b_l a_r$$

$$p_3 : b_r \rightarrow a_r$$

$$p_4 : b_l \rightarrow a_l$$





Anabaena Catenula: Model 1

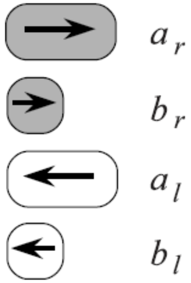
$$\omega : a_r$$

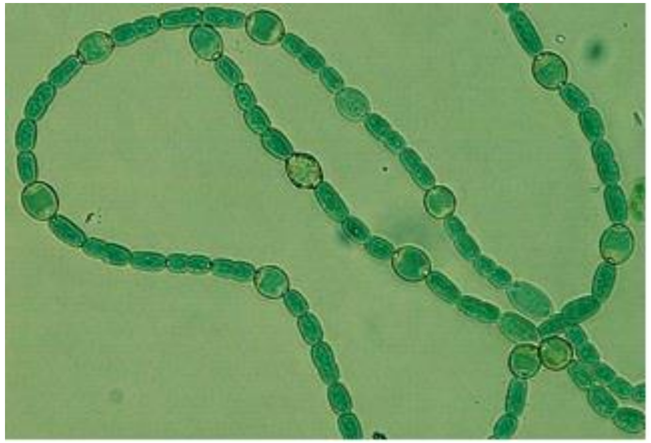
$$p_1 : a_r \rightarrow a_l b_r$$

$$p_2 : a_l \rightarrow b_l a_r$$

$$p_3 : b_r \rightarrow a_r$$

$$p_4 : b_l \rightarrow a_l$$





Anabaena Catenula: Model 1

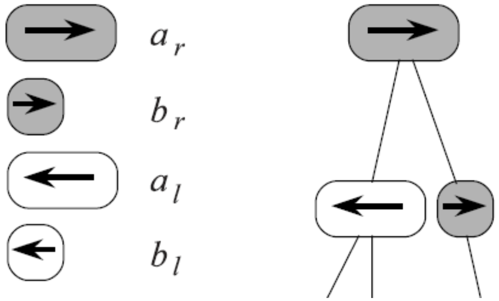
$\omega : a_r$

$p_1 : a_r \rightarrow a_l b_r$

$p_2 : a_l \rightarrow b_l a_r$

$p_3 : b_r \rightarrow a_r$

$p_4 : b_l \rightarrow a_l$



a_r

$a_l b_r$

Anabaena Catenula: Model 1

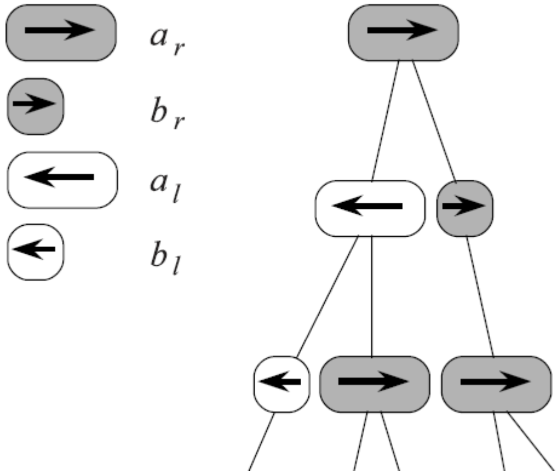
$\omega : a_r$

$p_1 : a_r \rightarrow a_l b_r$

$p_2 : a_l \rightarrow b_l a_r$

$p_3 : b_r \rightarrow a_r$

$p_4 : b_l \rightarrow a_l$



a_r

$a_l b_r$

$b_l a_r a_r$

Anabaena Catenula: Model 1

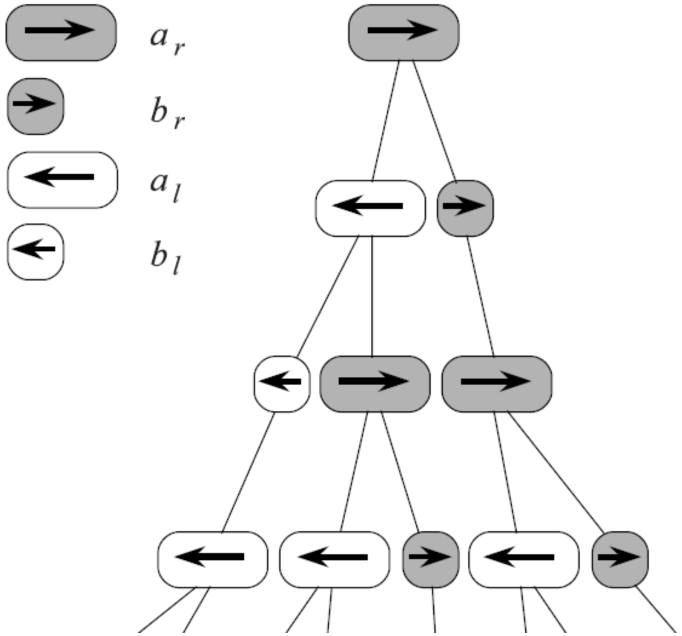
$\omega : a_r$

$p_1 : a_r \rightarrow a_l b_r$

$p_2 : a_l \rightarrow b_l a_r$

$p_3 : b_r \rightarrow a_r$

$p_4 : b_l \rightarrow a_l$



a_r

$a_l b_r$

$b_l a_r a_r$

$a_l a_l b_r a_l b_r$

Anabaena Catenula: Model 1

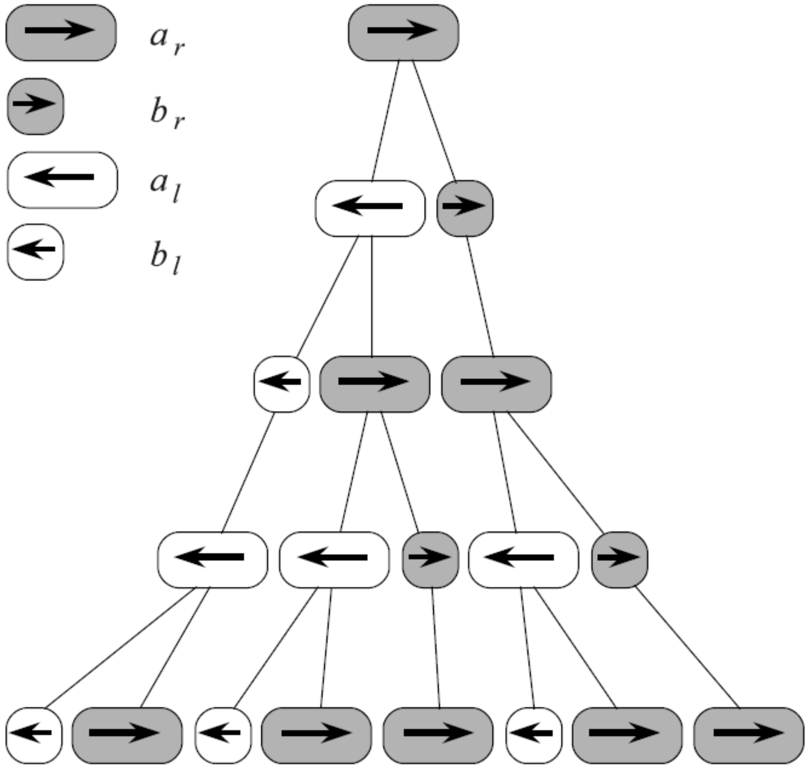
$\omega : a_r$

$p_1 : a_r \rightarrow a_l b_r$

$p_2 : a_l \rightarrow b_l a_r$

$p_3 : b_r \rightarrow a_r$

$p_4 : b_l \rightarrow a_l$



a_r

$a_l b_r$

$b_l a_r a_r$

$a_l a_l b_r a_l b_r$

$b_l a_r b_l a_r a_r b_l a_r a_r$