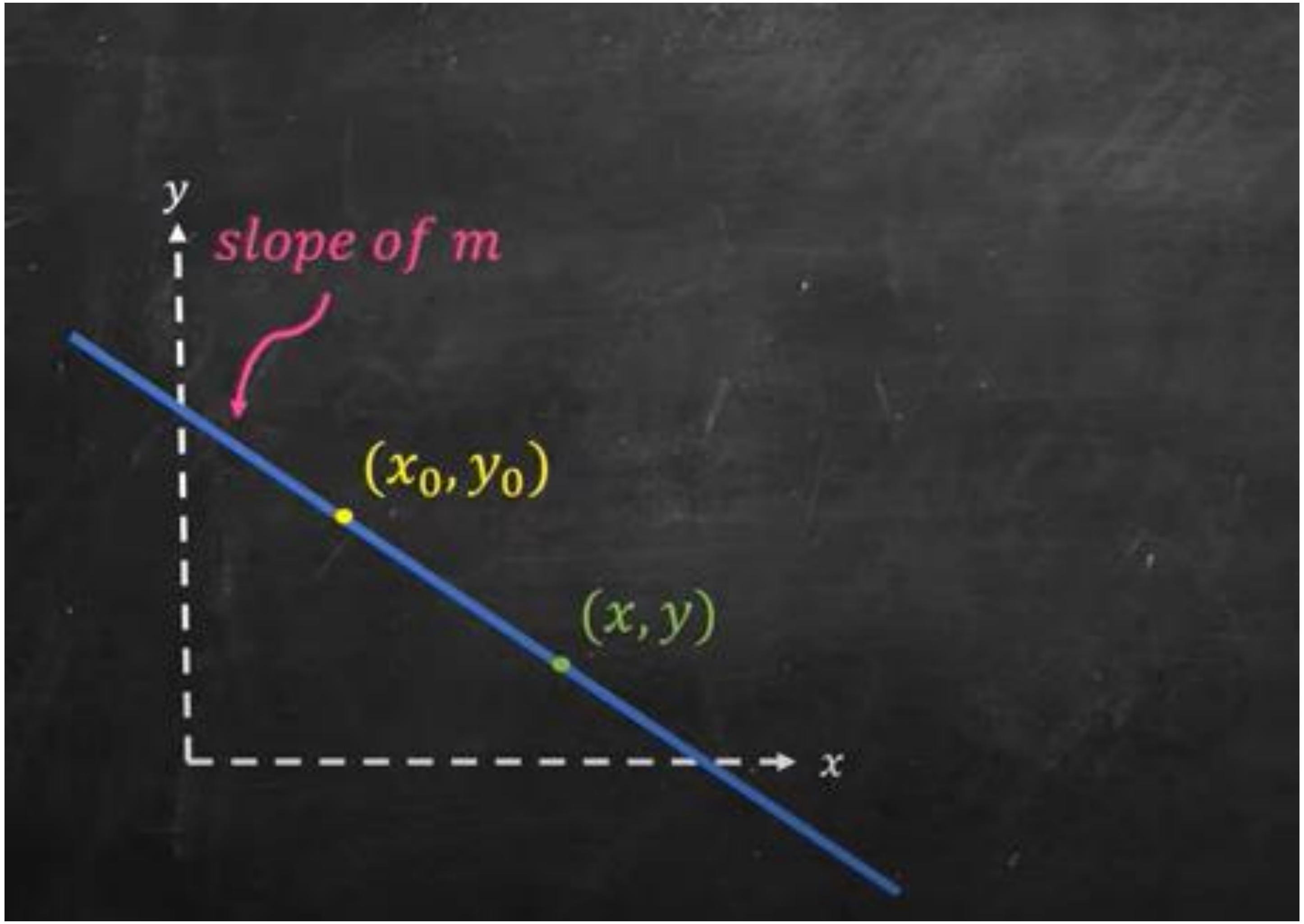
Line in Space (3D)

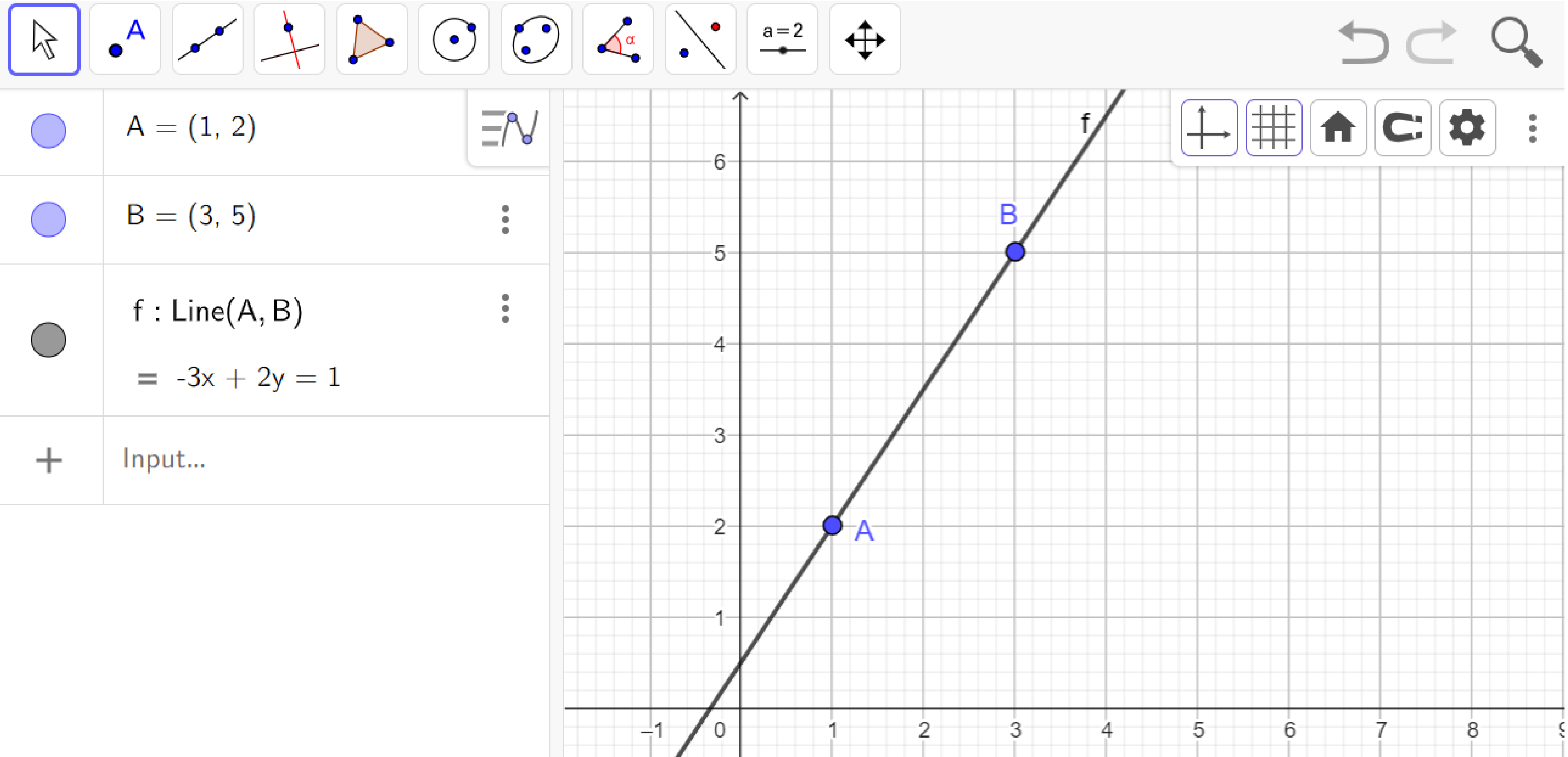
Equation of a line in 2-D

$$y - y_0 = m(x - x_0)$$

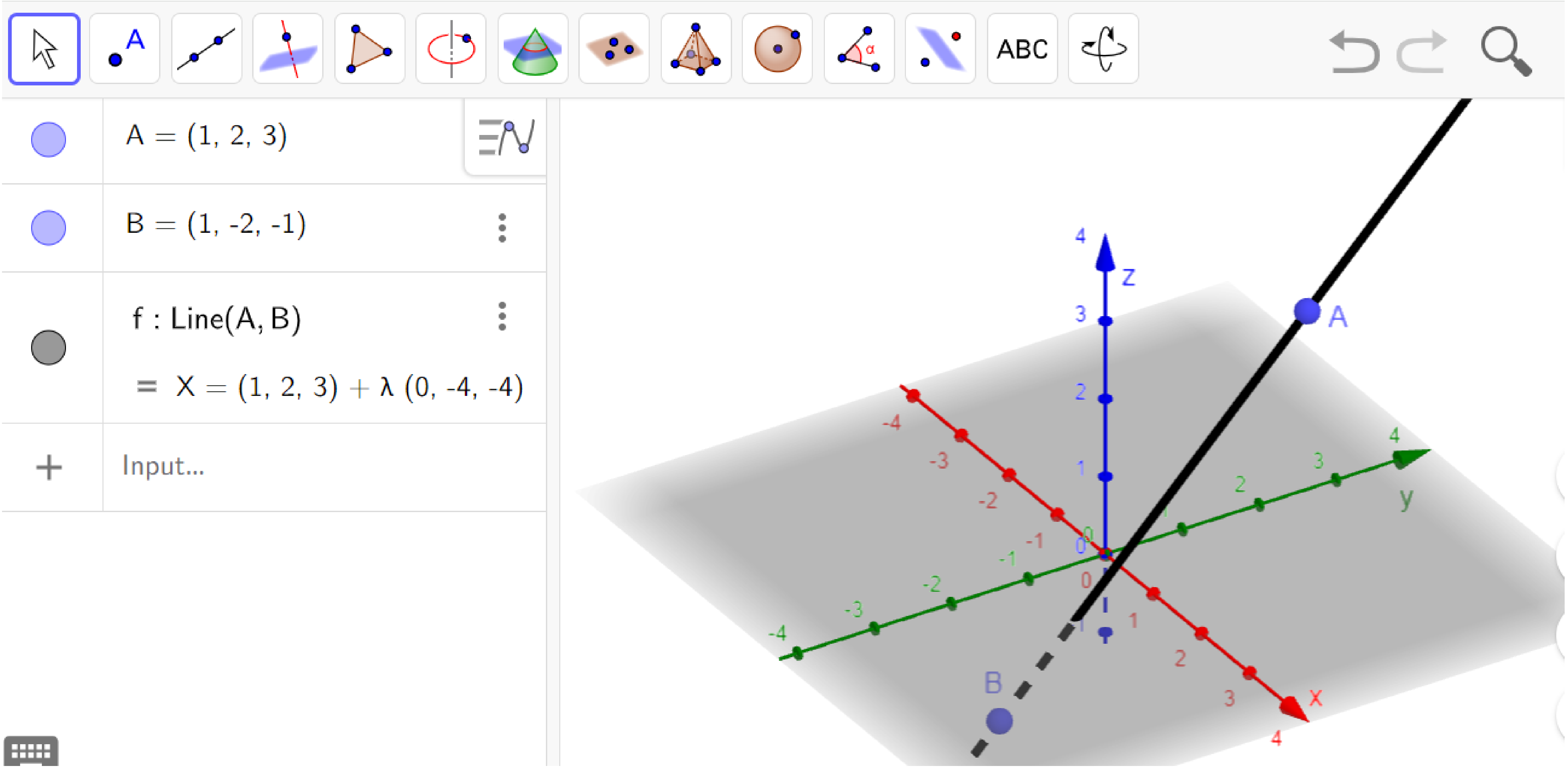


Equation of a line in 2-D

$$ax + by + c = 0$$



Equation of a line in 3-D



Equation of a line in 3-D

In Previous slide we observe that algebraic form of a line in 3-D is

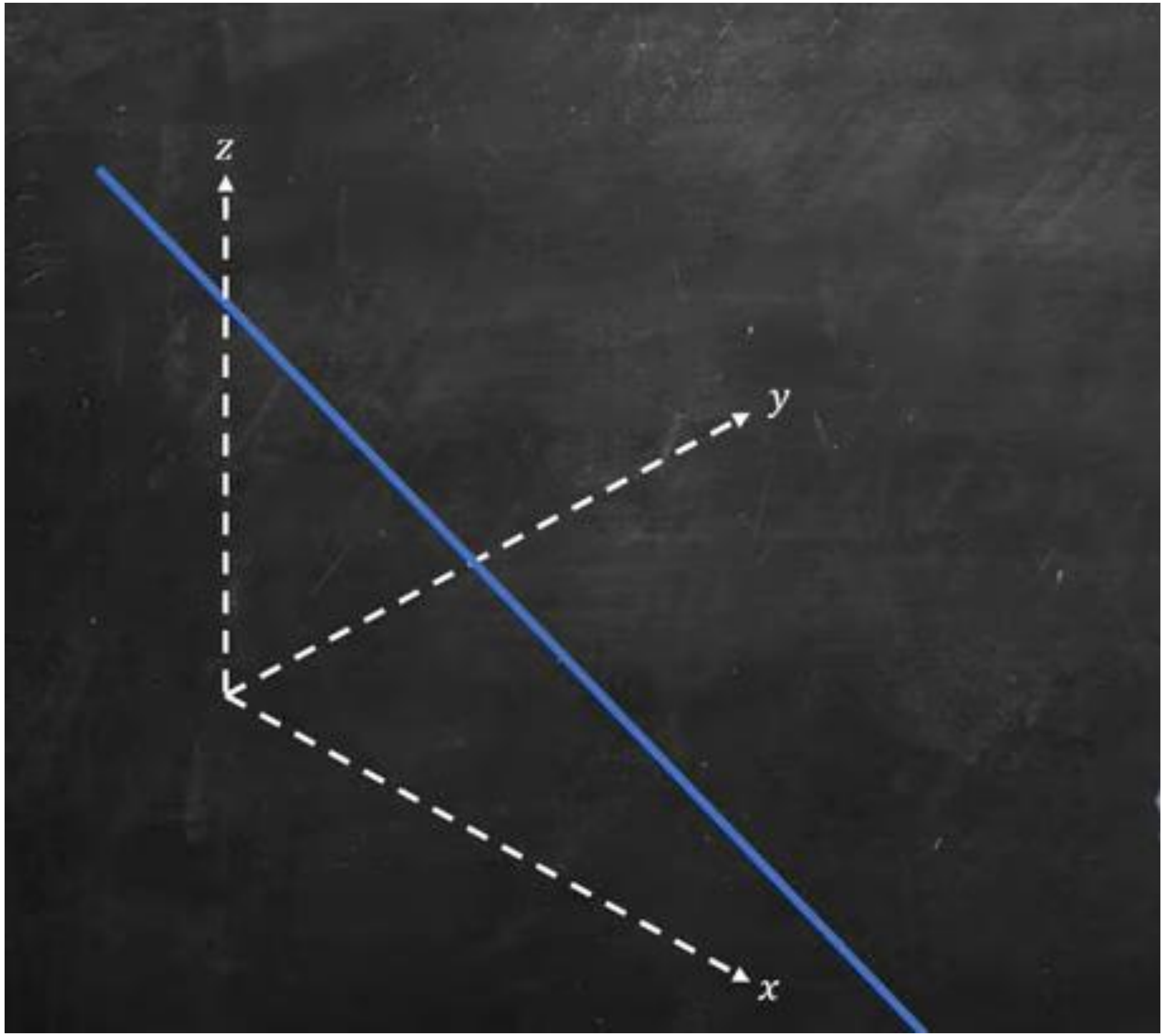
$$X = \langle 1, 2, 3 \rangle + \lambda \langle 0, -4, -4 \rangle$$

Which is generalize form of

$$\vec{\boldsymbol{r}} = \overrightarrow{\boldsymbol{r_0}} + \mathrm{t}\,\vec{\boldsymbol{v}}$$

In next slides, we will show this expression in step by step.

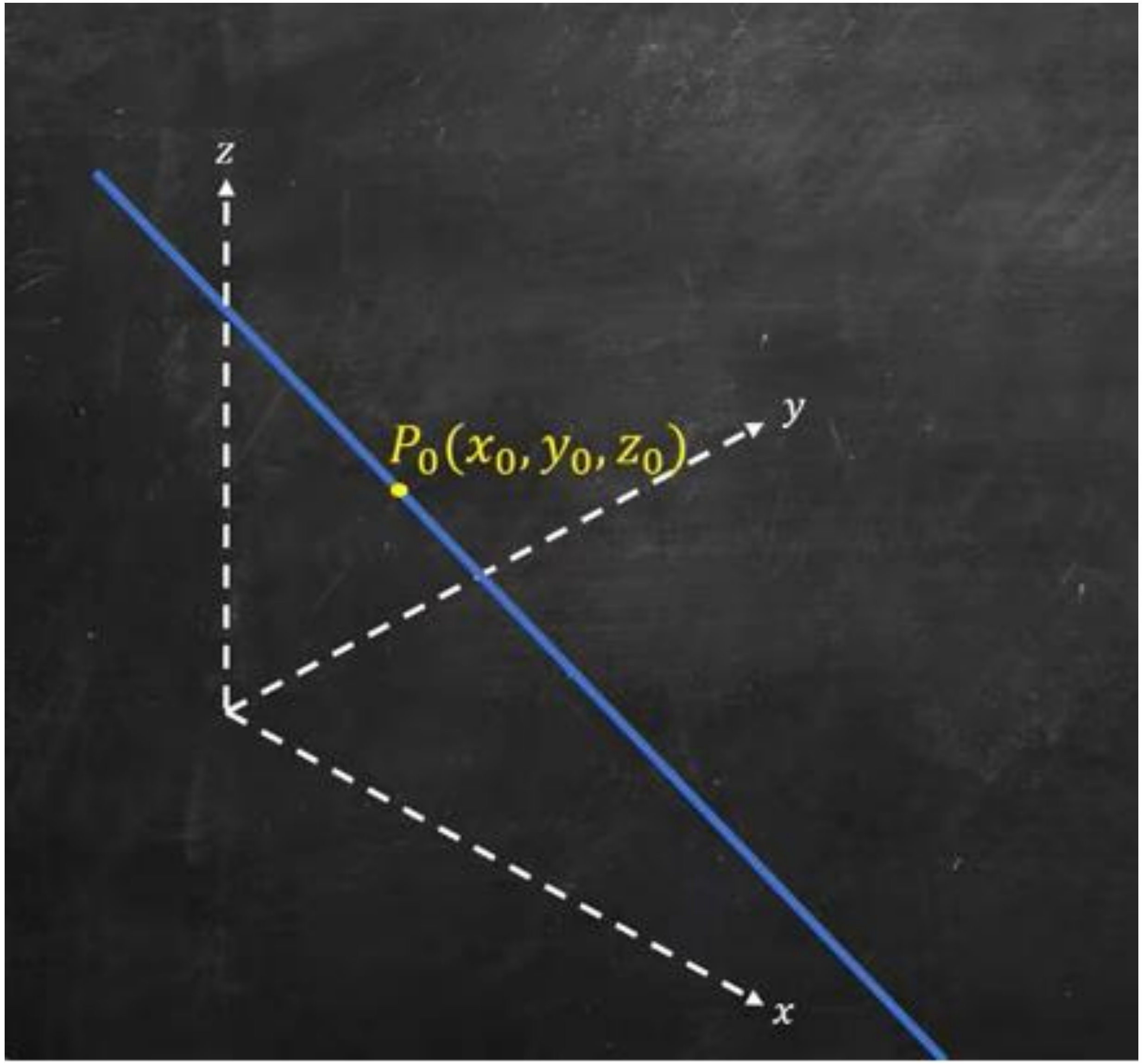
Equation of a line in 3-D



Equation of a line in 3-D

Consider a point on a Line

$P_0(x_0, y_0, z_0)$



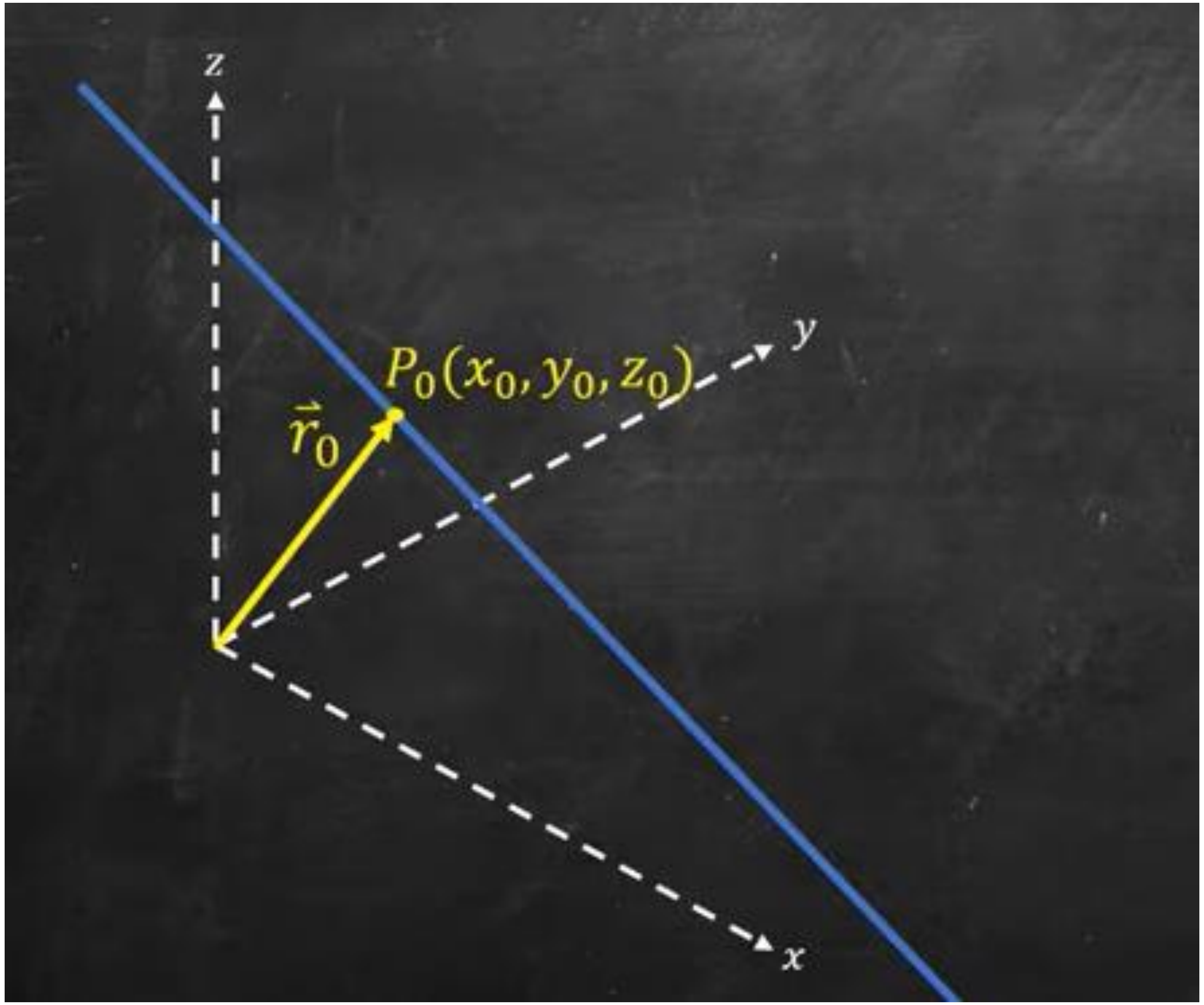
Equation of a line in 3-D

Consider a point on a Line

$$P_0(x_0, y_0, z_0)$$

Whose position vector is

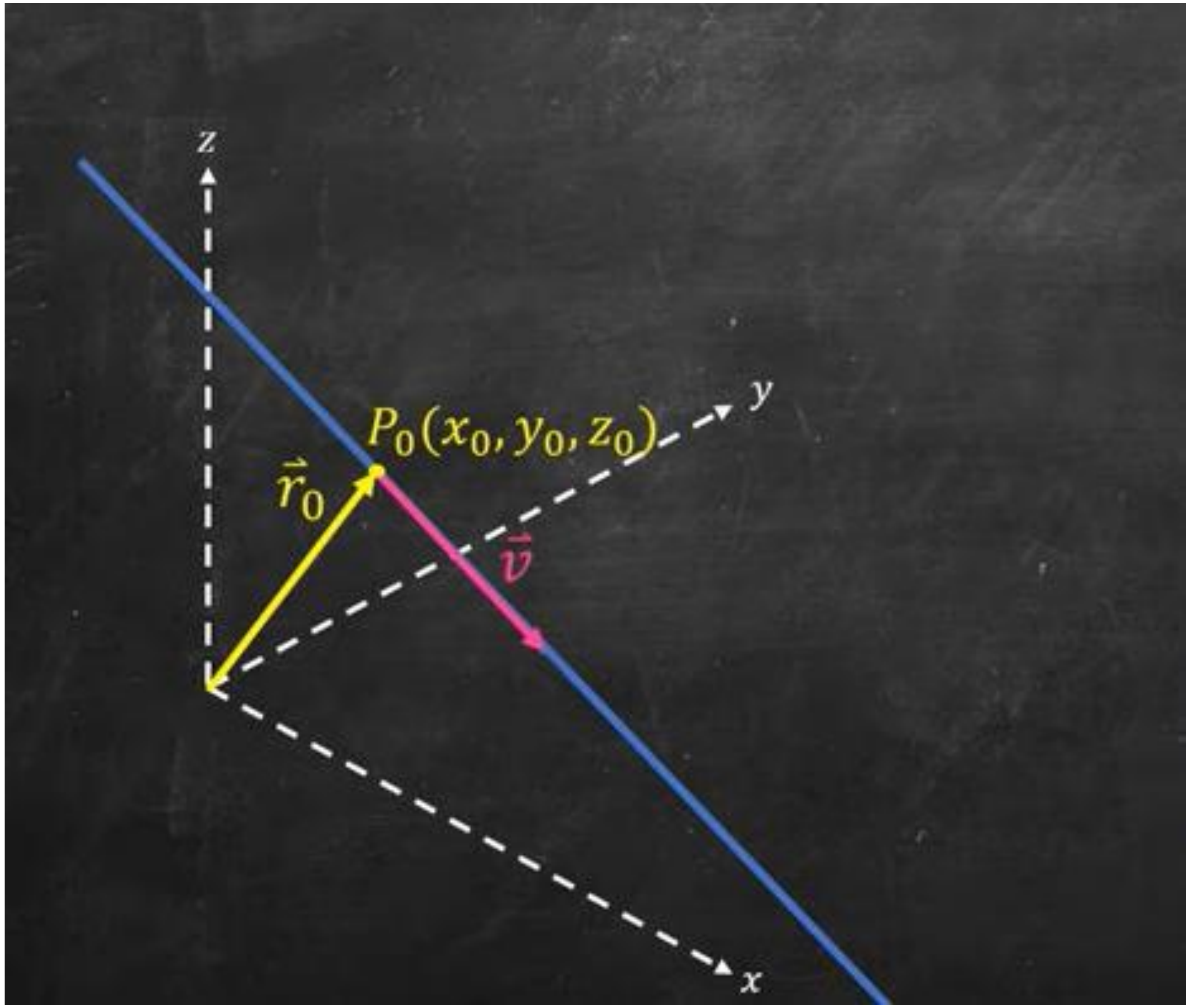
$$\overrightarrow{\boldsymbol{r}_0}$$



Equation of a line in 3-D

Suppose a vector along the line named as

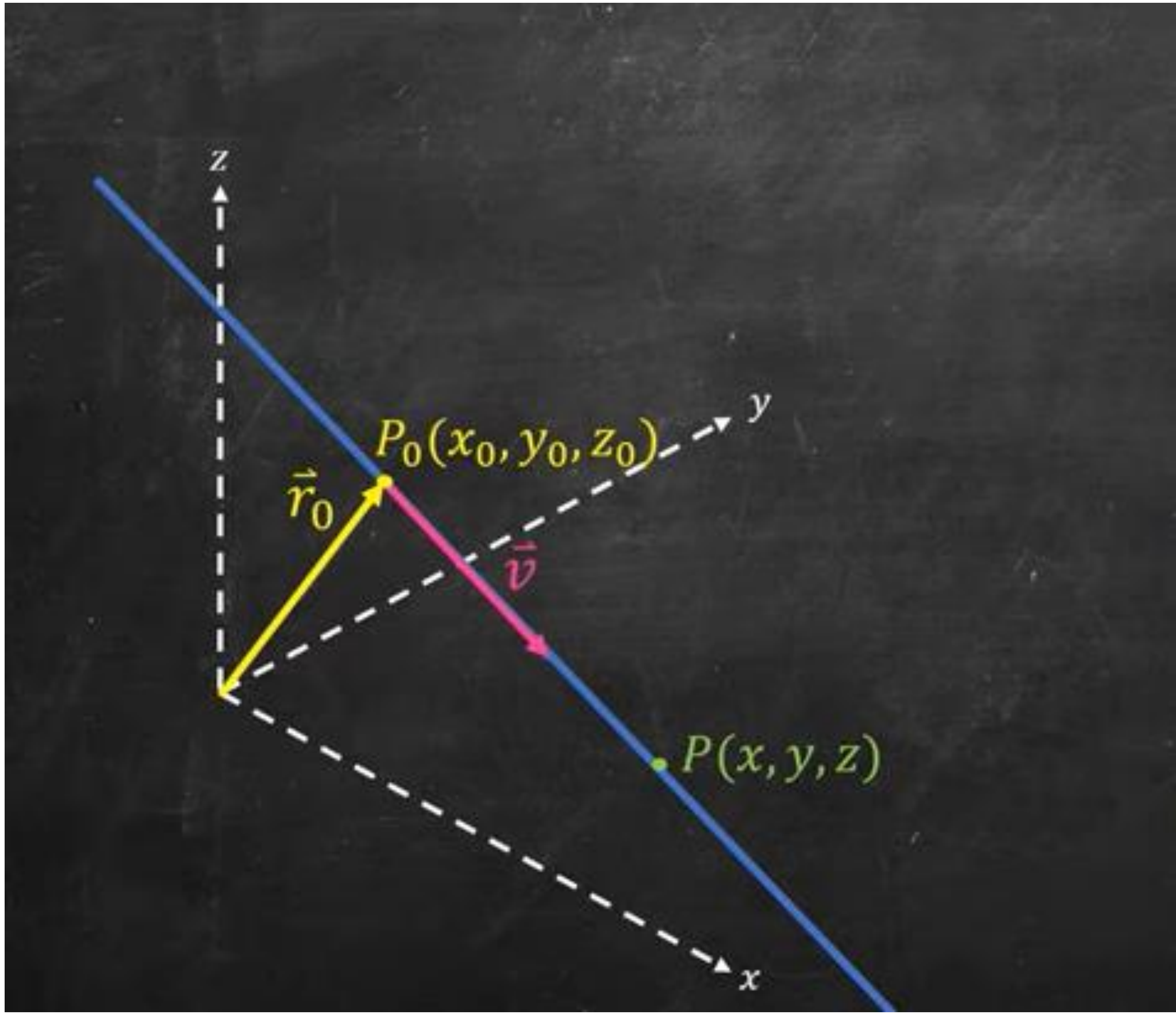
$\vec{v}$



Equation of a line in 3-D

Consider any arbitrary point on the Line

$$P(x, y, z)$$



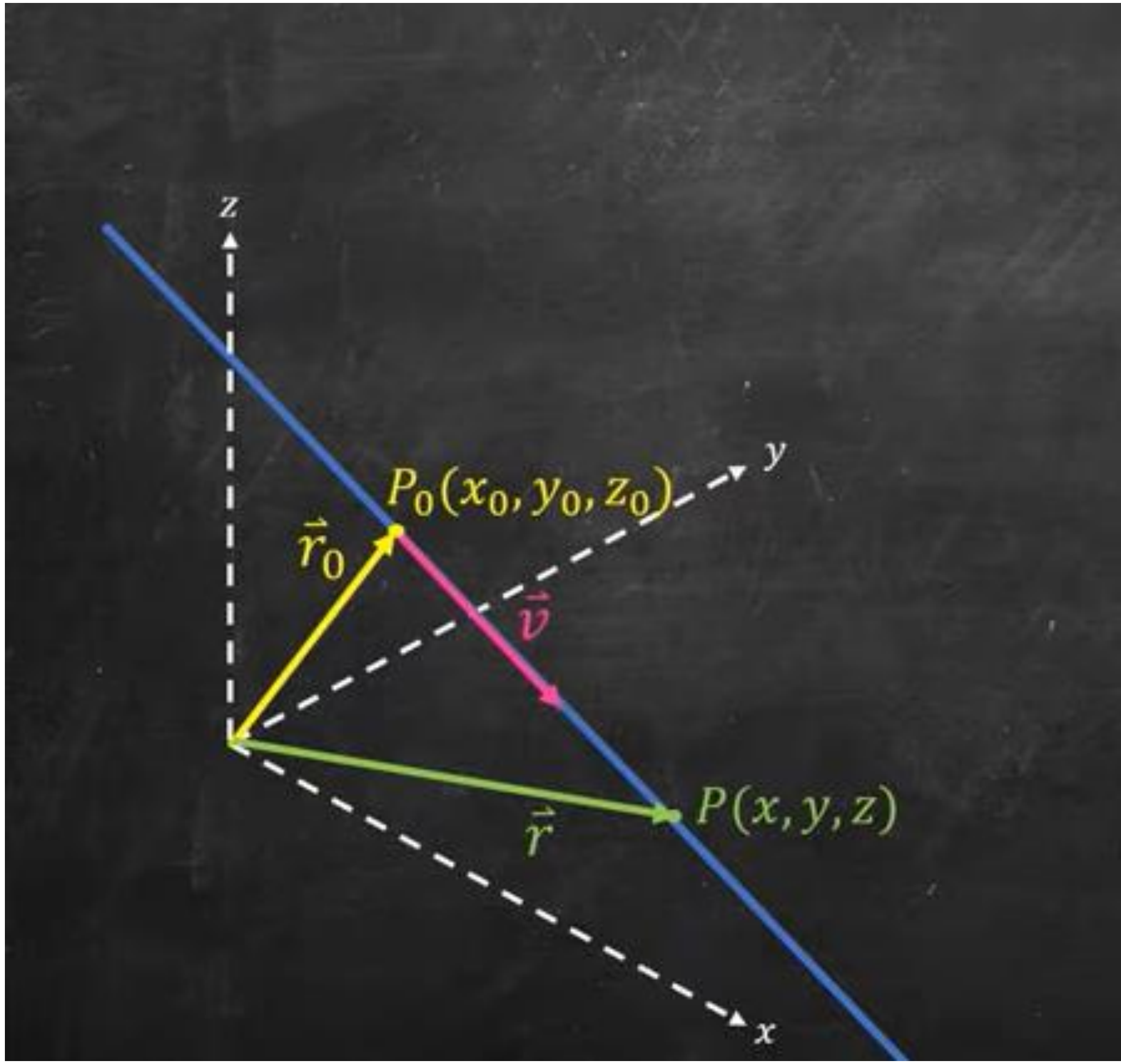
Equation of a line in 3-D

Consider any arbitrary point on the Line

$$P(x, y, z)$$

Whose position vector is

$$\vec{\boldsymbol{r}}$$



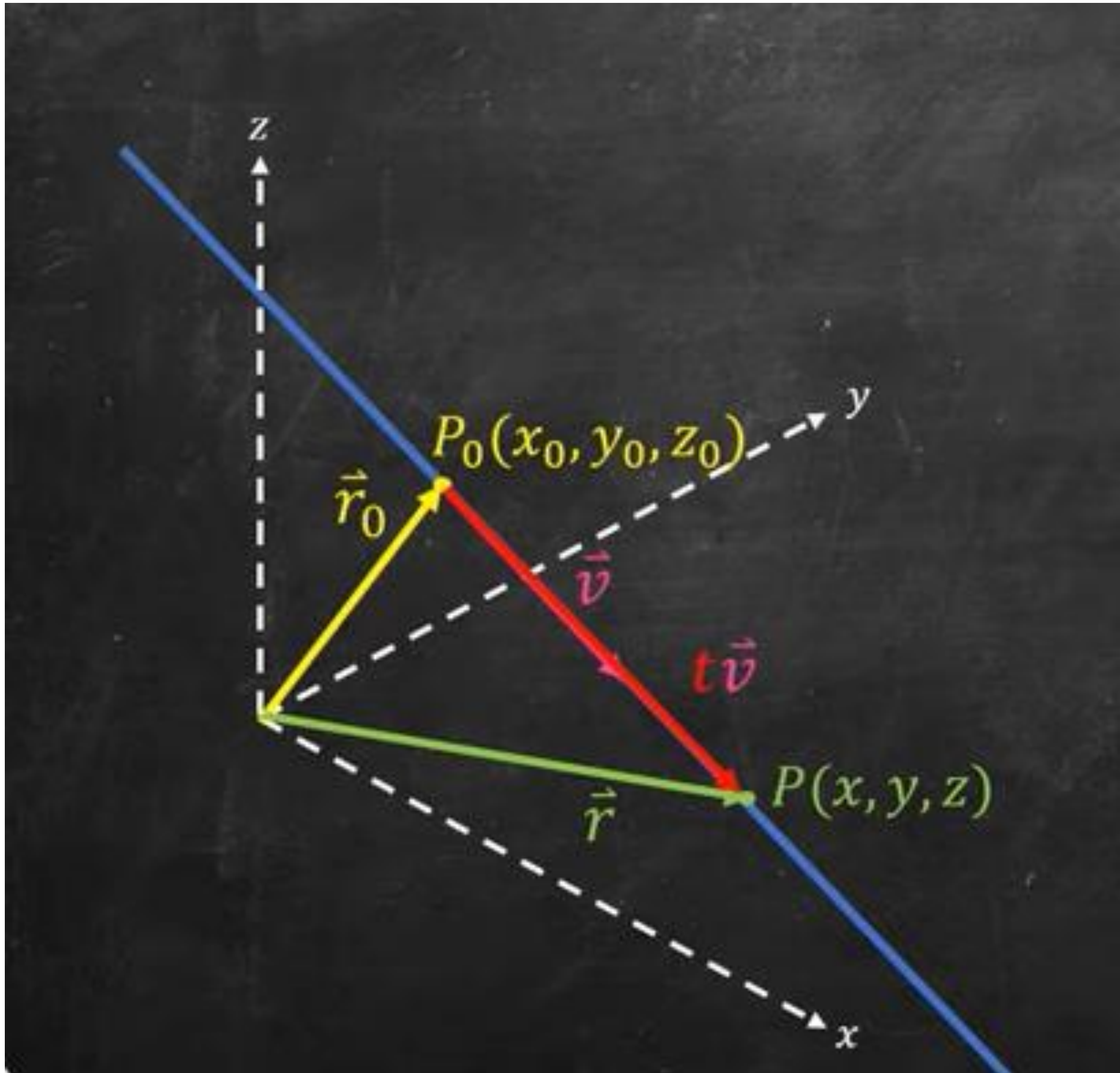
Equation of a line in 3-D

We can make a vector

t $\vec{v}$ from vector $\vec{v}$

up to the tip of the point P

so that we have a triangle.

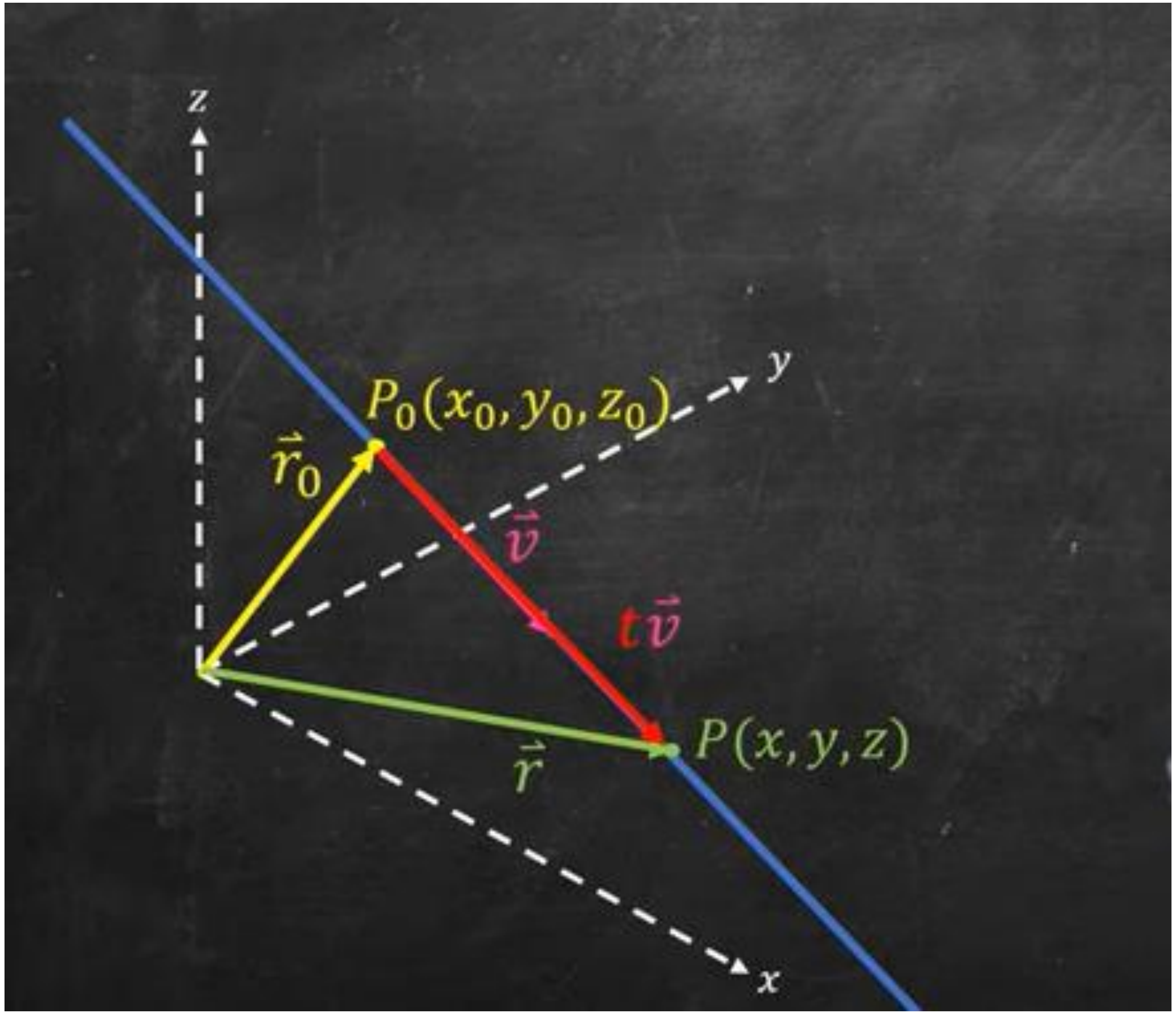


Equation of a line in 3-D

Now, from vector addition we have a vector equation of a line

$$\vec{r} = \overrightarrow{r_0} + t\,\vec{v}$$

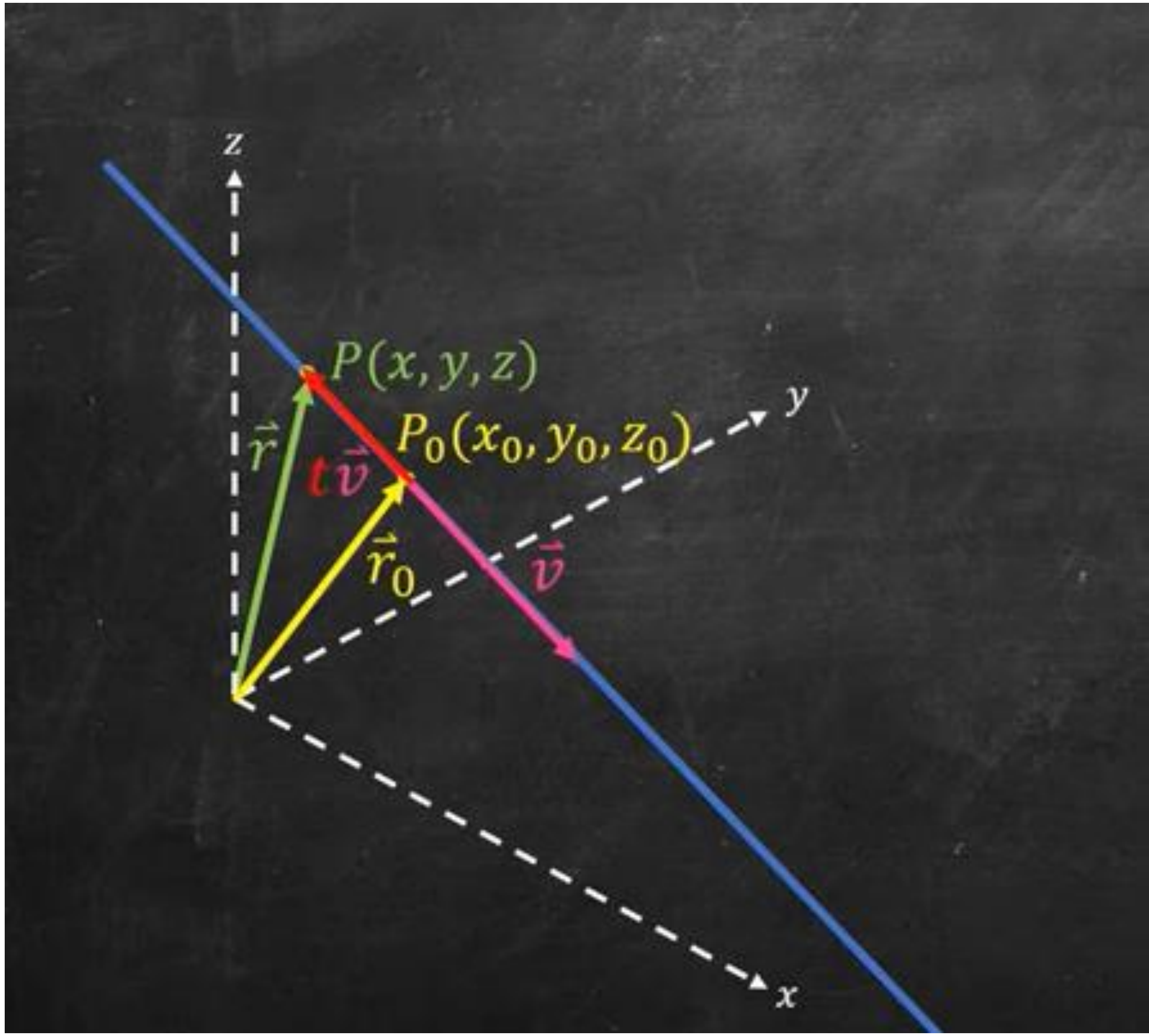
Which is required vector equation of a line in 3-D



Equation of a line in 3-D

It still works if we have a Point P in the opposite direction

$$\vec{r} = \overrightarrow{r_0} + t\,\vec{v}$$



Equation of a line in 3-D

And It will also work if vector $\vec{v}$ is not on the line but parallel to the line

Here

$$\vec{u} = t\vec{v}$$

From Fig

$$\vec{r} = \vec{r_0} + \vec{u}$$

$$\vec{r} = \vec{r_0} + t\,\vec{v}$$

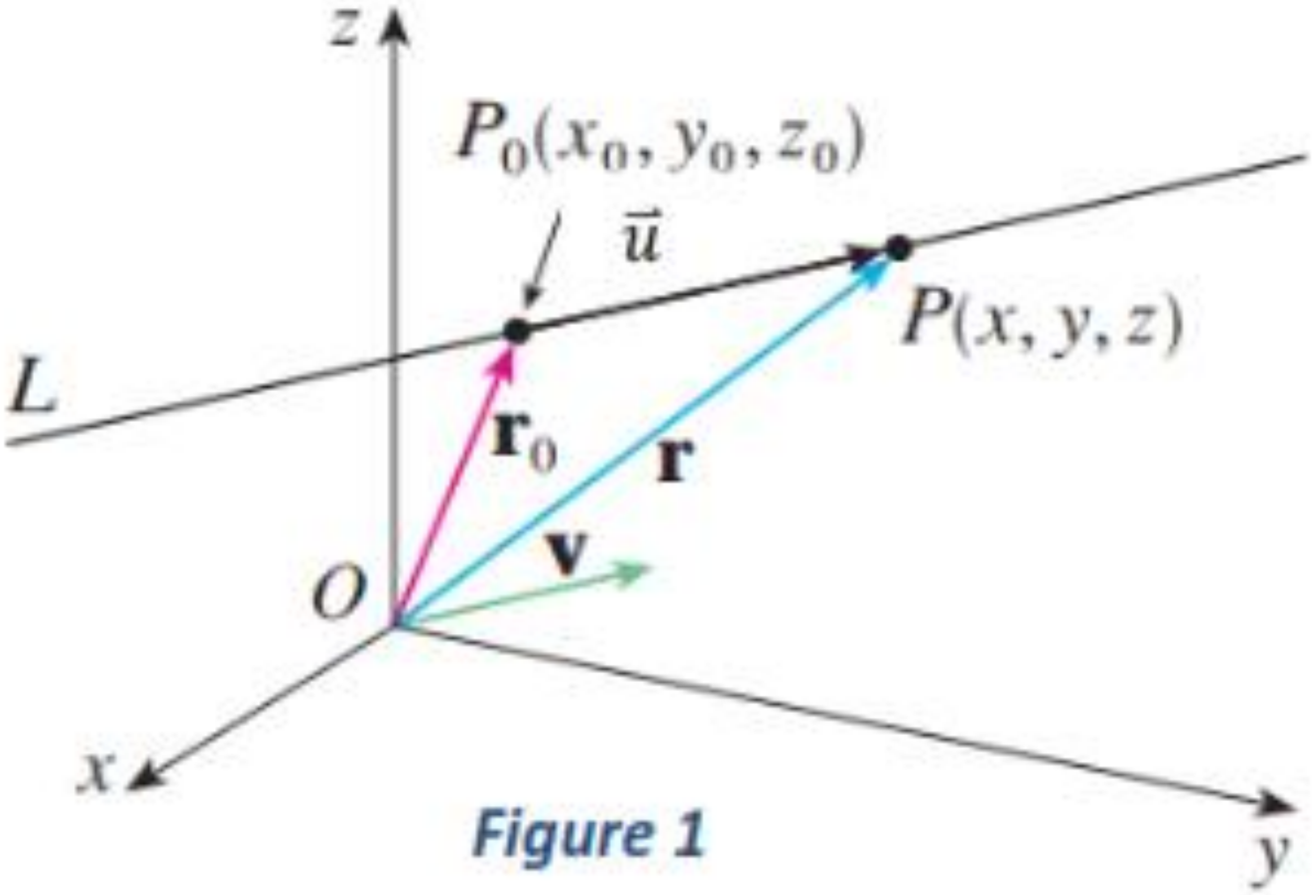


Figure 1

Equation of a line in 3-D

If the vector $\vec{v}$ that gives the direction of the line $\boldsymbol{L}$, is written in component form as

$\vec{v} = < a, b, c >$, then we have $t\,\vec{v} = < t\,a, t\,b, t\,c >$.

We can also write $\vec{r} = < x, y, z >$ and $\overrightarrow{r_0} = < x_0, y_0, z_0 >$,

So, the vector equation $\vec{r} = \overrightarrow{r_0} + t\,\vec{v}$

Becomes

$$< x, y, z > = < x_0 + ta, \quad y_0 + tb, \quad z_0 + tc >$$

Therefore, we have the three scalar equations:

$$\boxed{x = x_0 + ta, \qquad y = y_0 + tb, \qquad z = z_0 + tc \quad \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (2)}$$

Example 1:

Find **parametric equation** & **vector equation** for the line through the point $(-\mathbf{2}, \mathbf{0}, \mathbf{4})$ and parallel to the vector

$$\vec{\boldsymbol{v}} = \mathbf{2}\vec{\boldsymbol{\imath}} + \mathbf{4}\vec{\boldsymbol{\jmath}} - \mathbf{2}\vec{\boldsymbol{k}}$$

Example 2:

Find the **parametric equation** of the line passing through origin and parallel to the vector

$$\vec{v} = 2\hat{\jmath} + \hat{k}$$

Example 3:

Find the **parametric equation** and **vector equation** of the line through $\boldsymbol{P}(-\mathbf{3}, \mathbf{2}, -\mathbf{3})$ and $\boldsymbol{Q}(\mathbf{1}, -\mathbf{1}, \mathbf{4})$.

Example 4:

Find the **parametric equation** of the line passing through the point $(3,-2,1)$ and parallel to the line having parametric equations

$$x = 1 + 2t$$

$$y = 2 - t$$

$$z = 3t$$

Example 5:

Find the parametric equations of the line passing through the point $(2,3,0)$, and perpendicular to the vectors $\vec{u} = \hat{\imath} + 2\hat{\jmath} + 3\hat{k}$ and $\vec{v} = 3\hat{\imath} + 4\hat{\jmath} + 5\hat{k}$.

Example:

Parameterize the line segment joining the points $\boldsymbol{P}\ (-3, 2, -3)$ and $\boldsymbol{Q}\ (1, -1, 4)$.

Question 19: Find the parametric equations of the line segment joining the points $\boldsymbol{P}(-2, 0, 2)$ and $\boldsymbol{Q}(0, 2, 0)$.

Practice Questions

(Textbook: Thomas Calculus 11th Edition) Ex. 12.5: 13-20.

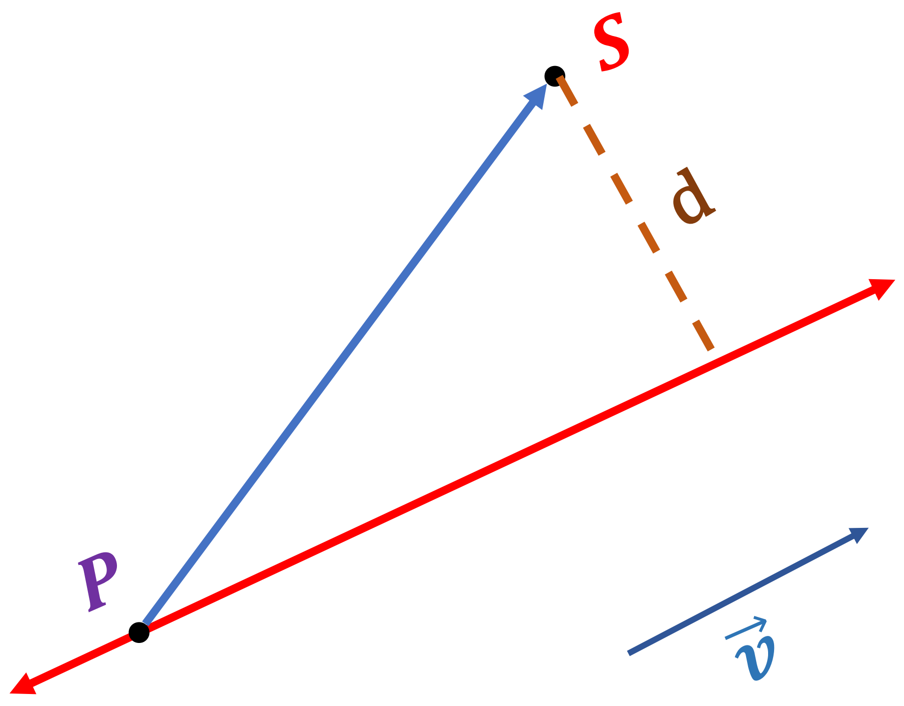
The Distance from a Point to a Line in Space

Distance from a point S

to a Line through Point P

Parallel to $\vec{v}$

$$d = \frac{|\overrightarrow{PS} \times \vec{v}|}{|\vec{v}|}$$



Example

Find the distance from the point $S(1, 1, 5)$ to the line

$$L: \quad x = 1 + t, \quad y = 3 - t, \quad z = 2t.$$

Practice Questions:

Textbook: Thomas Calculus 11th Edition Ex. 12.5: 33-38

In Exercises 33–38, find the distance from the point to the line.

33. $(0, 0, 12)$; $x = 4t, \quad y = -2t, \quad z = 2t$

34. $(0, 0, 0)$; $x = 5 + 3t, \quad y = 5 + 4t, \quad z = -3 - 5t$

35. $(2, 1, 3)$; $x = 2 + 2t, \quad y = 1 + 6t, \quad z = 3$

36. $(2, 1, -1)$; $x = 2t, \quad y = 1 + 2t, \quad z = 2t$

37. $(3, -1, 4)$; $x = 4 - t, \quad y = 3 + 2t, \quad z = -5 + 3t$

38. $(-1, 4, 3)$; $x = 10 + 4t, \quad y = -3, \quad z = 4t$

Python Code (Example)

a) User inputs vectors and computes dot and cross product

b) User inputs points P and Q, and the equation of the line through P and Q is computed

c) Distance between a point and a line

d) User inputs a point and the coefficients of the plane equation, and distance is calculated

```
# Include NumPy, a Python library used for working with arrays.

import numpy as np
```

Part (a) - User inputs vectors and computes dot and cross product

```python
print("Enter the components of vector A (separated by spaces):")
A = np.array(list(map(int, input().split())))
print("Enter the components of vector B (separated by spaces):")
B = np.array(list(map(int, input().split())))
# Dot product
dot_product = np.dot(A, B)
print("\nDot product of A and B:", dot_product)
# Cross product
cross_product = np.cross(A, B)
print("Cross product of A and B:", cross_product)
```

Part (b) - User inputs points P and Q, and the equation of the line through P and Q

```
print("\nEnter the coordinates of point P (separated by spaces):")
P = np.array(list(map(float, input().split())))
print("Enter the coordinates of point Q (separated by spaces):")
Q = np.array(list(map(float, input().split())))
# Direction vector of the line PQ
direction_vector = Q - P
print("\nParametric equations of the line passing through P and Q:")
print(f"x = {P[0]} + {direction_vector[0]}t")
print(f"y = {P[1]} + {direction_vector[1]}t")
print(f"z = {P[2]} + {direction_vector[2]}t")
```

Part (c) - Distance between a point and a line

$$d = \frac{|\overrightarrow{PS} \times \vec{v}|}{|\vec{v}|}$$

```
# User input for the coordinates of a point S
print("Enter the coordinates of the point (separated by spaces):")
point = np.array(list(map(float, input().split())))

# User input for a point on the line (let's call it P)
print("Enter the coordinates of a point on the line (P) (separated by spaces):")
P = np.array(list(map(float, input().split())))

# User input for the direction vector v⃗ of the line
print("Enter the components of the direction vector of the line (separated by spaces):")
direction_vector = np.array(list(map(float, input().split())))
```

```python
# Vector $\overrightarrow{PS}$ (from point P on the line to the given point)
PS = point - P

# Cross product of PS and the direction vector of the line
cross_product = np.cross(PS, direction_vector)

# Distance calculation: |PS x v| / |v|

distance = np.linalg.norm(cross_product) / np.linalg.norm(direction_vector)

print(f"\nThe distance between the point S and the line is: {distance}")
```

Part (d) - User inputs a point and the coefficients of the plane equation, and distance is calculated

```
# Ax, By, Cz and D coefficients

print("\nEnter the coefficients of the plane Ax + By + Cz = D (separated by spaces):")

plane_normal = np.array(list(map(float, input().split()))[:3])

plane_constant = float(input("Enter the constant D: "))


# Coordinates of Point

print("Enter the coordinates of the point (separated by spaces):")

point = np.array(list(map(float, input().split())))
```

Distance formula: $|Ax_1 + By_1 + Cz_1 - D| / \sqrt{A^2 + B^2 + C^2}$

```
numerator = abs(np.dot(plane_normal, point) - plane_constant)

denominator = np.linalg.norm(plane_normal)

distance = numerator / denominator

print("\nDistance from the point to the plane:", distance)
```