

> **restart:**

> with(*geom3d*) :

> with(*Student[MultivariateCalculus]*) :

> with(*LinearAlgebra*) :

> with(*linalg*) :

Ejercicio 4.1

Se elabora la ecuacion de la recta l1, se crea el vector v1 y el punto a que componen la recta

> *v1* := *vector*([3, 0, 2]);

$$v1 := \begin{bmatrix} 3 & 0 & 2 \end{bmatrix} \quad (1.1)$$

> *point*(*a*, 2, 1, -2);

$$a \quad (1.2)$$

> *line*(*l1*, [*a*, *v1*]) : # Se crea la ecuacion de la recta l1

> *detail*(*l1*); #se verifica la creacion de la recta l1

Warning, assume that the parameter in the parametric equations is *t*

Warning, assuming that the names of the axes are *x*, *y*, and *z*

name of the object *l1*

form of the object *line3d*

equation of the line [*x*=2+3*t*, *y*=1, *z*=-2+2*t*]

(1.3)

Se elabora la ecuacion de la recta l2, se crea el vector v2 y el punto b que componen la recta

> *v2* := *vector*([1, 2, 4]);

$$v2 := \begin{bmatrix} 1 & 2 & 4 \end{bmatrix} \quad (1.4)$$

> *point*(*b*, 1, 3, 4);

$$b \quad (1.5)$$

> *line*(*l2*, [*b*, *v2*]) : # Se crea la ecuacion de la recta l2

> *detail*(*l2*); #se verifica la creacion de la recta l2

Warning, assume that the parameter in the parametric equations is *t*

Warning, assuming that the names of the axes are *x*, *y*, and *z*

name of the object *l2*

form of the object *line3d*

equation of the line [*x*=1+*t*, *y*=3+2*t*, *z*=4+4*t*]

(1.6)

Ahora con la ecuacion l1 y l2 se calcula la distancia entre ellas.

> **distance**(*l1*, *l2*) ;

$$\frac{5}{19} \sqrt{38} \quad (1.7)$$

> **evalf**(%, 3) ;

$$1.62 \quad (1.8)$$

Ejercicio 4.2

Con el vector V3 se crea la ecuacion de la recta l3

> *v3* := *vector*([1, -1, 2]);

$$v3 := \begin{bmatrix} 1 & -1 & 2 \end{bmatrix} \quad (2.1)$$

```
> point(c, 0, 0, 0);
```

$$c$$

(2.2)

```
> line(l3, [c, v3]) : # Se crea la ecuacion de la recta l3
```

```
> detail(l3);
```

Warning, assume that the parameter in the parametric equations is _t

Warning, assuming that the names of the axes are _x, _y, and _z

name of the object $l3$

form of the object $line3d$

equation of the line $[_x=_t, _y=-_t, _z=2_t]$

Ahora se elabora la ecuacion del plano con los dos vectores y el origen

```
> v4 := vector([1, -2, -3]);
```

$$v4 := \begin{bmatrix} 1 & -2 & -3 \end{bmatrix}$$

(2.4)

```
> v5 := vector([2, 3, 4]);
```

$$v5 := \begin{bmatrix} 2 & 3 & 4 \end{bmatrix}$$

(2.5)

```
> p1 := Plane([0, 0, 0], (v4), (v5)) : GetRepresentation(p1)
```

$$x - 10y + 7z = 0$$

(2.6)

```
> FindAngle(l3, p1); #Funcion que encuentra angulo entre recta y plano, no me funciono
```

Error, (in geom3d:-FindAngle) wrong type of arguments

Utilice la ecuacion de calculo de angulo entre recta y un plano

Se determina las coordenadas del vector normal al plano p1

```
> vector_normal_p1 := crossprod(v4, v5);
```

$$\text{vector_normal_p1} := \begin{bmatrix} 1 & -10 & 7 \end{bmatrix}$$

(2.7)

Se calcula la norma del vector normal del plano p1.

```
> norma_1 := norm(vector_normal_p1, 2);
```

$$\text{norma_1} := 5\sqrt{6}$$

(2.8)

Se calcula la norma del vector director a la recta l3.

```
> norma_2 := norm(v3, 2);
```

$$\text{norma_2} := \sqrt{6}$$

(2.9)

Se realiza el producto escalar entre el vector director de la recta l3 y el vector normal del plano p1. Como el resultado es positivo, no determine el valor absoluto.

```
> num := dotprod(v3, vector_normal_p1);
```

$$\text{num} := 25$$

(2.10)

```
> den := (norma_1) * (norma_2);
```

$$\text{den} := 30$$

(2.11)

```
> argumento := evalf(num/den);
```

$$\text{argumento} := 0.8333333333$$

(2.12)

Se calculo el angulo entre el plano y la recta.

```
> arcsin(argumento);
```

$$0.9851107833$$

(2.13)

Se realiza la conversion a grados del angulo obtenido.

```
> angulo_grados := (0.9851107833 * (180/Pi));
```

$$\text{angulo_grados} := 56.44269023$$

(2.14)