13. Vectores

13.1. Operaciones con vectores

Para definir un vector **debemos escribir las coordenadas entre llaves y separadas por comas**. Las coordenadas pueden ser numéricas o simbólicas. Los vectores pueden ser de cualquier dimensión. Si realizamos operaciones entre vectores, estas se realizan **coordenada a coordenada**. En particular la suma, la resta y el producto por escalares son los asociados a la estructura de espacio vectorial.

Ejercicios

- Dado los vectores $u = (3,5,7)$ y $v = (2,-5,9)$ realiza combinaciones lineales con ellos.
- Realiza otras operaciones «no vectoriales».

13.2. Dibujo de vectores

La función **Arrow[{p1,p2}]** crea una flecha que comienza en el punto p1 y termina en el punto p2, pero no la dibuja. Si queremos dibujar los objetos en dos dimensiones, utilizamos en comando **Graphics** y en tres dimensiones el comando **Graphics3D**. Puede ser interesante poner la opción **Axes** en True para poder ver los ejes.

Ejercicios

Representa el vector $u = (3,2)$ y $v = (2,3,1)$.

Nuevas funciones

Arrow, Graphics, Graphics3D.

13.3. Producto escalar y vectorial

La función **Norm[u]** calcula el módulo o norma del vector. La función **Dot[u,v]** calcula el producto escalar (también se puede realiza con un punto) y la función **Cross[u,v]** el producto vectorial. Con **VectorAngle[u,v]** podemos calcular el ángulo entre dos vectores.

Ejercicios

- Dados $u = (2,3,2)$ y $v = (1,-2,1)$ calcula $u \cdot v$.
- Comprueba que el producto escalar de u consigo mismo es el cuadrado de la norma del vector.
- Calcula el ángulo que forman los vectores u y v.
- Calcula el producto vectorial de los vectores.
- Comprueba que producto vectorial es perpendicular a cada factor.

Nuevas funciones

Norm, Dot, Cross, VectorAngle.

13.4. Proyección ortogonal

La proyección de un vector u sobre v es un vector proporcional a v, cuyo módulo es igual a la «sombra» del vector u. La magnitud de dicha sombra se puede calcular con la fórmula $\frac{u.v}{v}$. Todo esto lo puede hacer Mathematica con el comando **Projection[u,v]**.

Ejercicios

- Calcula la proyección p de u sobre v.
- Comprueba que la proyección p y v son paralelos.
- Comprueba que $u - p$ es perpendicular a v.

Nuevas funciones

Projection.

13.5. Ortogonalización de Gram-Schmidt

Con proyecciones y normalizando se puede obtener la ortogonalización de cualquier conjunto de vectores. Dado un conjunto cualquiera de vectores, la ortogonalización consiste en un conjunto de vectores, que formen un conjunto ortonormal y que generen el mismo subespacio que los vectores de partida.

Afortunadamente Mathematica tiene el comando **Orthogonalize[{u,v,...}]**. Los vectores deben ir dentro de una lista.

Ejercicios

Ortogonaliza el conjunto. Comprueba el resultado:

$$u = (2,4,5,7), v = (2,-5,8,9), w = (3,5,1,2)$$

Nuevas funciones

Orthogonalize.

13.6. Comprobaciones con cálculo simbólico

La comprobación de ciertas propiedades se puede realizar con vectores numéricos. Sin embargo el cálculo simbólico nos permite «probar» propiedades, trabajando con letras, en vez de utilizar números. Se pueden realizar las operaciones y comprobar «a ojo» que son iguales. **Para comprobar si dos objetos son iguales se coloca un doble signo igual**. Si la respuesta es **True**, ambos son iguales.

Ejercicios

Realiza las siguientes comprobaciones:

- El producto escalar es conmutativo.
- $u.u = |u|^2$.
- El producto vectorial es anticonmutativo.
- El producto vectorial es perpendicular a cada factor.

13.7. Generación de sucesiones aritméticas

En programación se utilizan muchos los vectores de cualquier dimensión, en un sentido ligeramente distinto al matemático. La función **Range[n]** crea un vector formado por los números enteros del 1 al n. Si añadimos otro número, se forma una secuencia numérica entre ambos números. Incluso admite un tercer argumento, que es el incremento de la secuencia.

La función **Table[a[n],{n,nmin,nmx}]** crea un vector donde todos sus elementos corresponden al término general a_n. La n es la variable (aunque puede ser cualquier otra) y los números indican el primer y el último término.

Ejercicios

- Crea un vector con números del 1 al 9.
- Crea un vector con números del 5 al 13.
- Crea un vector con números del 5 al 6 y con un incremento de 0.1.
- Crea un vector con los cubos de los 7 primeros números.

Nuevas funciones

Range, Table.

13.8. Indexación de vectores

Para acceder a las componentes de un vector **se utiliza el doble corchete** (con el doble punto y coma). También se puede utilizar la función **Part[v,i;;j]**. **Los números negativos en la indexación empieza a contar desde el final.**

Ejercicios

Dado el vector $v = (4, 7, 8, 5, 4, 6, 7, 1)$:

- Mostrar la tercera componente y la segunda empezando por el final.
- Cambiar el valor de la segunda componente por 90.
- Extraer partes del vector.

Nuevas funciones

Part.