

Introducción a Mathematica

jltabara@gmail.com

Índice

1. Números enteros y racionales	9
1.1. Operaciones con números enteros	9
1.2. Potencias de exponente entero	10
1.3. Operaciones combinadas	11
1.4. Operaciones con números racionales	12
2. Números decimales y radicales	13
2.1. Operaciones con números decimales	13
2.2. Aproximaciones decimales y constantes	14
2.3. Potencias fraccionarias y radicales	15
2.4. Operaciones con radicales	16
3. Funciones elementales	17
3.1. Funciones para «redondear»	18
3.2. Funciones trigonométricas	19

3.3. Trigonómicas inversas e hiperbólicas	20
3.4. Exponenciales y logaritmos	21
3.5. Definición de nuevas funciones	22
3.6. Funciones aplicadas a varios valores	23
4. Divisibilidad	24
4.1. División con resto	24
4.2. Números primos	25
4.3. Factorización	26
4.4. Divisores, mcd y mcm	27
5. Polinomios	28
5.1. Operaciones	28
5.2. División con resto	29
5.3. Factorización	30
5.4. Mcd y mcm	31
5.5. Sustitución de variables	32
5.6. Raíces de polinomios	33
5.7. Polinomio interpolador de Lagrange	34
5.8. Números algebraicos y polinomio mínimo	35
6. Fracciones algebraicas	36

6.1. Simplificación	36
6.2. Desarrollar fracciones	37
6.3. Operaciones	38
6.4. Factorización	39
6.5. Descomposición en fracciones simples	40
7. Ecuaciones y sistemas	41
7.1. Los comandos fundamentales	42
7.2. Ecuaciones algebraicas	43
7.3. Sistemas con n° variables = n° ecuaciones	44
7.4. Sistemas con más incógnitas que ecuaciones	45
7.5. Otros tipos de ecuaciones algebraicas	46
7.6. Ecuaciones no algebraicas	47
7.7. Inecuaciones	48
7.8. El método de sustitución	49
8. Números complejos	50
8.1. La unidad imaginaria	50
8.2. Operaciones	51
8.3. Forma polar	52
8.4. Funciones con argumento complejo	53

9. Gráficas de funciones	54
9.1. Gráfica de una o más funciones	54
9.2. Algunas opciones interesantes	55
9.3. El comando Show	56
9.4. Gráficos de funciones implícitas	57
10.Límites	58
10.1.Noción intuitiva de límite	58
10.2.Indeterminaciones	59
10.3.Derivadas como límites	60
10.4.Continuidad	61
10.5.Asíntotas	62
11.Derivadas	63
11.1.Cálculo de derivadas	63
11.2.Derivadas de orden superior	64
11.3.Derivadas simbólicas	65
11.4.Cálculo de tangentes a curvas	66
11.5.Extremos y puntos de inflexión	67
11.6.Series de potencias	68
12.Integrales	69

12.1. Integrales indefinidas	69
12.2. Integrales definidas	70
12.3. Integrales impropias	71
12.4. Integrales de fracciones algebraicas	72
12.5. Integrales dependientes de parámetros	73
12.6. Cálculo de áreas	74
12.7. Sumas de series numéricas	75
13. Vectores	76
13.1. Operaciones con vectores	76
13.2. Dibujo de vectores	77
13.3. Producto escalar y vectorial	78
13.4. Proyección ortogonal	79
13.5. Ortogonalización de Gram-Schmidt	80
13.6. Comprobaciones con cálculo simbólico	81
13.7. Generación de sucesiones aritméticas	82
13.8. Indexación de vectores	83
14. Matrices	84
14.1. Introducción de matrices	84
14.2. Operaciones matriciales	85
14.3. Inversas, determinantes y transpuestas	86

14.4.Rango y reducción por filas	87
14.5.Matrices y vectores	88
14.6.Construcción de matrices especiales	89
14.7.Submatrices	90
14.8.Miscelanea de matrices	91
15.Sistemas de ecuaciones lineales	92
15.1.El comando Solve	92
15.2.El comando RowReduce	93
15.3.Multiplicar por la matriz inversa	94
15.4.El comando LinearSolve[A,B]	95
16.Estadística descriptiva	96
16.1.Los distintos tipos de media	96
16.2.Mediana y moda	97
16.3.Máximo, mínimo y cuantiles	98
16.4.Varianza y desviación típica	99
16.5.Asimetría, kurtosis y momentos	100■
16.6.Covarianza y correlación	101■
17.Gráficos estadísticos	102
17.1.Tablas de frecuencias discretas	102■

17.2. Recuento de clases e histogramas	103■
17.3. Diagrama de cajas y bigotes	104■
17.4. Diagramas de dispersión	105■
18. Factorización de Matrices	106
18.1. Descomposición LU	106■
18.2. Descomposición QR	107■
18.3. Descomposición de Choleski	108■
18.4. Descomposición de Schur	109■
19. Diagonalización de matrices	110
19.1. Los autovectores y autovalores	110■
19.2. Los comandos de diagonalización	111■
19.3. La descomposición en valores singulares	112■
19.4. La forma de Jordan	113■
20. Cálculo vectorial	114
20.1. Dibujo de campos vectoriales	114■
20.2. Gradiente, laplaciano y hessiano	115■
20.3. Divergencia, rotacional y jacobiano	116■
20.4. Identidades vectoriales	117■
20.5. Otros sistemas de coordenadas	118■

21.Transformada de Laplace	119
21.1.Definición	119■
21.2.El comando LaplaceTransform	120■
21.3.La transformada como aplicación lineal	121■
21.4.Algunas propiedades	122■
21.5.La transformada inversa	123■
22.Ecuaciones diferenciales	124
22.1.Solución de una ecuación diferencial	124■
22.2.Ecuaciones de primer orden	125■
22.3.Ecuaciones de mayor orden	126■
22.4.Ecuaciones con valor inicial	127■
22.5.El campo vectorial asociado a la ecuación	128■
22.6.Sistemas de ecuaciones diferenciales	129■
22.7.Resolución numérica	130■
23.Expresiones trigonométricas	131
23.1.Simplificaciones trigonométricas	131■

1. Números enteros y racionales

1.1. Operaciones con números enteros

Si trabajamos con **números enteros**, Mathematica nos facilita los **resultados exactos**, por muy grandes que estos sean. Los operadores para realizar la suma y la resta son los habituales (**+**, **-**). Para multiplicar se utiliza el asterisco (*****) (aunque a veces basta la yuxtaposición, o también un simple espacio en blanco). El operador de división es la barra inclinada (**/**). **En la división, si los números son enteros**, Mathematica entiende que es una fracción y no la transforma en número decimal. Simplemente nos devuelve el **resultado simplificado**. Si queremos el resultado decimal en la división, **escribimos un punto después de alguno de los términos de la división**. Entonces Mathematica entiende que uno de ellos es un número decimal y que lo que queremos es la división con decimales.

Ejercicios

Realiza las siguientes operaciones con números enteros:

a) $5 + 8$

b) $7 - 9$

c) 56×45

d) $48/30$

Nuevas funciones

Plus, Subtract, Minus, Times, Divide.

1.2. Potencias de exponente entero

La potencia se calcula con el circunflejo (\wedge). Si trabajamos con números enteros el resultado es siempre exacto. Si el **exponente es negativo** nos devuelve la correspondiente fracción, aplicando la propiedad $a^{-n} = \frac{1}{a^n}$.

Ejercicios

Realiza las siguientes potencias de enteros:

$$a) 2^3 \quad b) 2^{-6} \quad c) -2^8 \quad d) (-2)^8 \quad e) 2^{525}$$

Nuevas funciones

Power.

1.3. Operaciones combinadas

En las operaciones combinadas Mathematica sigue la jerarquía habitual de las matemáticas: primero realiza los paréntesis, después las potencias, posteriormente las multiplicaciones y divisiones y finalmente las sumas y las restas. **No se pueden usar corchetes aunque si se pueden anidar paréntesis.** Cuando tengamos dos paréntesis seguidos (uno que cierra y otro que abre) si no se escribe ninguna operación entre ellos, el programa realiza la multiplicación, del mismo modo que en la notación matemática. Lo mismo si tenemos un número delante de un paréntesis.

Ejercicios

Realiza las siguientes operaciones combinadas:

$$a)(4+2) \times 3 \quad b)(3+7-6) \times (7-5) \quad c)((3+8) \times 3)^2$$

1.4. Operaciones con números racionales

Mathematica opera de **modo exacto con las fracciones**, a no ser que no aparezcan en ellas números decimales, y nos da **el resultado simplificado**. En cuanto aparece un número decimal, ya nos ofrece el resultado en decimales.

Si en una operación aparecen varias fracciones y un número decimal, el resultado siempre no los facilita como número decimal. De esta forma podemos obtener el resultado decimal de cualquier operación entre fracciones.

Ejercicios

Realiza las siguientes operaciones con fracciones:

$$a) \frac{5}{6} + \frac{3}{4} + 1 \quad b) \frac{4}{5} + \left(\frac{78}{33} - \frac{2}{3} \times \frac{5}{3} \right)^4$$

Nuevas funciones

Rational.

2. Números decimales y radicales

2.1. Operaciones con números decimales

El **separador de decimales es el punto**. Cuando se opera con decimales el resultado se presenta, en principio, con **6 cifras significativas**, aunque internamente Mathematica trabaja con más.

Si el número real es lo «suficientemente» grande, Mathematica nos lo **devuelve en notación científica**. Para introducir un número en notación científica podemos escribirlo como una multiplicación o bien utilizar la notación ($\ast^$).

Si queremos transformar un número decimal en racional, utilizamos el comando **Rationalize[x]**.

Ejercicios

Realiza las siguientes operaciones con decimales y racionaliza alguno de ellos:

$$a) 3,56 + 7,9 \quad b) 3,56/78,9 \quad c) 1,34^5$$

$$d) 4,6^{50} \quad f) 2,4 \times 10^5 - 89$$

Nuevas funciones

Rationalize.

2.2. Aproximaciones decimales y constantes

Si queremos obtener el resultado de una operación con un número determinado de cifras significativas, empleamos la función **N[x,n]**. Es importante **poner corchetes y no paréntesis**, al contrario de la notación matemática. El segundo argumento es opcional e indica el número de cifras significativas del resultado. Si no indicamos este número, en principio Mathematica trabaja con 6 cifras significativas. Si queremos obtener el resultado con las cifras significativas predefinidas por Mathematica, escribimos **/N** tras la operación.

Algunas constantes importantes en Matemáticas, ya vienen incorporadas en el lenguaje: el número π se escribe como **Pi**, el número e como **E** y el número aureo ϕ se denota por **GoldenRatio**.

Ejercicios

- Calcula la división $456789/342$ con distintas cifras significativas.
- Calcula las constantes con distintas aproximaciones.
- Calcula $e^\pi + \pi^2$.

Nuevas funciones

N, Pi, E, GoldenRatio.

2.3. Potencias fraccionarias y radicales

Mathematica es capaz de calcular potencias de exponente negativo y fraccionario. Por lo tanto es capaz de calcular todo tipo de raíces. Para ello debemos tener en cuenta que $\sqrt[n]{x} = x^{1/n}$. Si las raíces no dan números racionales, no transforma en decimal el número, **únicamente lo simplifica**. Si queremos obtener una aproximación decimal podemos emplear **N[x]**.

Además de la notación de potencias, se pueden calcular raíces de otra forma: la **raíz cuadrada** se puede calcular con la función **Sqrt[x]**, donde es importante escribir la primera letra con mayúscula. La **raíz cúbica** se calcula con **CubeRoot[x]** y en general la **raíz de orden n** con la función **Surd[x,n]**, cuyo segundo argumento es el orden de la raíz. **El resultado, en general, aparece en forma de potencias.**

Ejercicios

Calcula de distintos modos los siguientes radicales:

$$a) \sqrt{2} \quad b) \sqrt[3]{2} \quad c) \left(\sqrt[5]{2}\right)^5 \quad d) \sqrt[3]{2^{11}}$$

Nuevas funciones

Sqrt, CubeRoot, Surd.

2.4. Operaciones con radicales

Es sabido que, en general, **los radicales son números irracionales**. Operar con ellos de forma exacta requiere conocer ciertas propiedades generales de los radicales. Mathematica puede manejar de manera simbólica los radicales en todo tipo de operaciones aunque algunas veces necesitaremos utilizar el comando **Simplify[x]** o el comando **Expand[x]**, para «ayudar» al programa. Estas funciones también se pueden emplear en la forma **//Simplify** y **//Expand** después de la operación.

Ejercicios

Realiza las siguientes operaciones con radicales:

- $a) \sqrt{2}\sqrt{3}$ $b) \sqrt[15]{2^5}$ $c) \sqrt{\sqrt[5]{2}}$
- $\sqrt[6]{2^5} \cdot \sqrt{2^7} \cdot \sqrt[3]{2^2}$
- $(3 + 5\sqrt{7})^3$
- $23\sqrt{125} + 3\sqrt{20} - 2\sqrt{45}$
- $\frac{5}{2\sqrt{3}-2}$

Nuevas funciones

Simplify, Expand.

3. Funciones elementales

Mathematica es lo que en inglés se denomina *case-sensitive*, esto es, que diferencia entre mayúsculas y minúsculas. Por ello no podemos escribir la misma palabra, pero cambiando mayúsculas por minúsculas. Además en Mathematica se siguen ciertas convenciones para escribir funciones:

- Las funciones **empiezan por mayúscula**. Si el nombre de la función se construye uniendo dos o más palabras, la inicial de cada palabra también va en mayúscula.
- Después de la función se colocan unos **corchetes** y no unos paréntesis, como es lo habitual en la notación matemática.
- Los distintos argumentos **se separan por comas**.

3.1. Funciones para «redondear»

Existen distintas funciones de redondeo. Unas redondean al entero más cercano (**Round[x]**), otras por defecto (**Floor[x]**) y otras por exceso (**Ceiling[x]**).

Un número decimal tiene una parte entera (**IntegerPart[x]**) y de una parte decimal (**FractionalPart[x]**)

Ejercicios

Haz distintos redondeos de números positivos y negativos.
Calcula su parte entera y decimal.

Nuevas funciones

Round, Floor, Ceiling, IntegerPart, FractionalPart.

3.2. Funciones trigonométricas

Las funciones trigonométricas se calculan por defecto en **rad**ianes. Para realizar el cálculo en **grados**, debemos multiplicar los grados por la constante predefinida **Degree**.

Mathematica siempre nos da el resultado exacto. Como para muchos valores, las funciones trigonométricas producen números con infinitos decimales, el único resultado correcto es dejar la salida igual a la entrada. Ello no quiere decir que Mathematica no sea capaz de realizar el cálculo. Si queremos el resultado con decimales empleamos la función **N[x]**.

Las funciones trigonométricas son: **Sin[x]**, **Cos[x]** y **Tan[x]**, aunque también existen funciones para la secante, la cosecante y la cotangente.

Ejercicios

Realiza los siguientes cálculos trigonométricos:

- $a) \cos\left(\frac{\pi}{3}\right)$ $b) \sin\left(\frac{\pi}{2}\right)$ $c) \tan\left(\frac{2\pi}{3}\right)$ $d) \cos(2)$
- $a) \cos(60^\circ)$ $b) \sin(90^\circ)$ $c) \tan(120^\circ)$ $d) \cos(7^\circ)$
- $3 \cos\left(\frac{\pi}{3}\right) + 7 \sin\left(\frac{3\pi}{4}\right)$

Nuevas funciones

Degree, Sin, Cos, Tan.

3.3. Trigonómicas inversas e hiperbólicas

Si añadimos el prefijo **Arc** al nombre de la función calculamos las funciones inversas y con el sufijo **h** calculamos funciones hiperbólicas. El resultado de las funciones trigonométricas inversas viene dado en radianes. Algunas de estas funciones son: **ArcSin[x]**, **Cosh[x]**, **ArcTanh[x]**,...

Ejercicios

Realiza los siguientes cálculos:

- a) $\arccos\left(\frac{1}{2}\right)$ b) $\arctan(1)$ c) $\arcsin(-1)$
- a) $\arcsin(0,3)$ b) $\arcsin\left(\frac{3}{5}\right)$ c) $\arcsin(2)$
- a) $\sinh(4)$ b) $\cosh(4)$ c) $\tanh(4)$

Nuevas funciones

ArcSin, ArcCos, ArcTan, Sinh, Cosh, Tanh, ArcSinh, ArcCosh, ArcTanh.

3.4. Exponenciales y logaritmos

La exponencial se calcula con la función **Exp[x]** o también elevando el número **E** a la potencia indicada.

El logaritmo **neperiano o natural** se calcula con **Log[x]**. El logaritmo en base 2 con **Log2[x]** y en base 10 con **Log10[x]**. En general para calcular el logaritmo en cualquier base se emplea la función **Log[b,x]**, donde el primer argumento es la base.

Ejercicios

Realiza los siguientes cálculos:

- a) $\exp(1)$ b) $\exp(5,2)$ c) $\exp\left(\frac{5}{3}\right)$
- a) $\ln(e)$ b) $\ln(e^5)$ c) $\ln\left(\frac{77}{3}\right)$
- a) $\log_2(8)$ b) $\log_2(\sqrt{8})$ c) $\log_5(125^4)$

Nuevas funciones

Exp, Log, Log2, Log10.

3.5. Definición de nuevas funciones

Además de las funciones que Mathematica trae predefinidas, nosotros podemos crear nuestras propias funciones. Estas pueden tener varios argumentos. La notación para construirlas se asemeja mucho a la tradicional de las matemáticas, pero con ligeras variaciones:

- A continuación del nombre de la variable debemos colocar un **guión bajo**.
- En vez de un signo igual debemos poner **:=** (aunque en funciones sencillas también nos sirve un único signo igual).

Ejercicios

- Construye la función $f(x) = 1 + x^2$ y calcula algunos valores.
- Construye la función $g(x, y) = 2x + 3y$ y calcula algunos valores.

3.6. Funciones aplicadas a varios valores

Si queremos aplicar una función a varios valores a la vez, debemos colocar los valores en lo que Mathematica denomina una **lista**. Para ello colocamos los valores **entre llaves** y **separados por comas**. Después le pasamos la lista como argumento a la función. La lista puede contener números, expresiones, variables,...

Este procedimiento se denomina **vectorización** en términos informáticos y permite realizar programas mucho más eficientes.

Ejercicios

Crea una lista con varios valores y aplica alguna de las funciones a la lista.

4. Divisibilidad

4.1. División con resto

Para obtener el cociente de una división con resto se utiliza la función **Quotient[x,y]** y para obtener el resto tenemos la función **Mod[x,y]**. Si queremos obtener los dos resultados a la vez, empleamos la función **QuotientRemainder[x,y]**. Esta función nos devuelve una lista, siendo la primera componente el cociente y la segunda el resto.

Ejercicios

Realiza las siguientes divisiones enteras:

$$a) 7/4$$

$$b) 89/33$$

$$c) (-5)/4$$

Nuevas funciones

Quotient, Mod, QuotientRemainder.

4.2. Números primos

Para saber si un número es primo utilizamos la función **PrimeQ[x]**. Si nos devuelve *True*, el número es primo. Si nos devuelve *False* entonces es compuesto.

Para obtener el primo n -ésimo se utiliza la función **Prime[n]**. Dado un número cualquiera, para saber cual es el número siguiente número primo a partir de él, se utiliza la función **NextPrime[x,n]**. Manejando el segundo argumento, podemos calcular también el primo anterior.

Ejercicios

- Comprueba si son primos o compuestos los siguientes números:

$$a) 7 \quad b) 1000 \quad c) 4323 \quad d) 2^{32} + 1$$

- Calcula el primer número primo y el situado en la posición 168.
- Calcula el primer primo que sigue al número 1000. Calcula también el anterior.
- Intenta encontrar, mediante ensayo y error, la cantidad de números primos que son menores que 5000.
- Hacer el ejercicio anterior con la función **PrimePi[x]**.

Nuevas funciones

PrimeQ, Prime, NextPrime, PrimePi.

4.3. Factorización

Para obtener la descomposición en números primos se utiliza la función **FactorInteger[n]**. La salida es una lista de lista. El primer elemento de la sublista es el factor y el segundo el exponente. También puede factorizar números negativos y fracciones. En el caso de la fracciones los factores del denominador aparecen con exponente negativo. Además se eliminan los factores repetidos (se factoriza la fracción irreducible).

Ejercicios

Factoriza los siguientes números:

$$a) 260 \quad b) -260 \quad c) \frac{260}{34} \quad d) 45!$$

Nuevas funciones

FactorInteger, Factorial.

4.4. Divisores, mcd y mcm

Divisors[n] nos devuelve el conjunto de divisores del número en cuestión. Para obtener el máximo común divisor de cualquier cantidad de números se utiliza el comando **GCD[n,m,...]**. Para el mínimo común múltiplo el comando es **LCM[n,m,...]**.

Ejercicios

- Calcula los divisores de 23 y de 60.
- Calcula el *mcd* y el *mcm* de 612 y 5292.
- Calcula lo mismo para 612, 5292 y 48.

Nuevas funciones

Divisors, GCD, LCM.

5. Polinomios

5.1. Operaciones

Los operadores para realizar las operaciones con los polinomios son los habituales. Cuando le pedimos a Mathematica que realice una operación con polinomios, en principio no la realiza. Para desarrollar la expresión empleamos el comando **Expand[pl]**. Se puede trabajar con polinomios en **cualquier número de variables**, aunque casi todo lo aplicaremos a polinomios en una variable.

Ejercicios

Realiza las siguientes operaciones con polinomios:

- $x^3 + (3x^2 + 7)(4x - 1) + (2x - 3)^4$
- $(4x^2 + 5)^2$
- $(x + y)^3$

Nuevas funciones

Expand.

5.2. División con resto

Para realizar la división con resto de dos polinomios tenemos a nuestra disposición distintos comandos, similares a los que hemos visto para números enteros. Todos ellos llevan el prefijo **Polynomial** y además como segundo argumento debemos indicar la variable del polinomio. Para calcular el cociente tenemos **PolynomialQuotient[p,q,x]**, y para el resto, **PolynomialRemainder[p,q,x]**. Para calcular ambos a la vez el comando **PolynomialQuotientRemainder[p,q,x]**.

Ejercicios

Realiza la siguiente división de polinomios, comprobando el resultado:

$$(x^3 + 5x^2 - 3x + 2) : (4x - 3)$$

Nuevas funciones

PolynomialQuotient, PolynomialRemainder, PolynomialQuotientRemainder .

5.3. Factorización

El comando para factorizar polinomios es **Factor[p]**. Si no le añadimos ninguna opción, **factoriza los polinomios en el cuerpo de los racionales** (si las raíces son números irracionales o complejos no factoriza el polinomio). Si ponemos algún coeficiente como número decimal, factoriza, en el cuerpo real, con raíces aproximadas. Para conseguir que factorice en el cuerpo complejo tenemos que poner la opción **GaussianIntegers** en *True*.

Ejercicios

Factoriza los siguientes polinomios y comprueba el resultado:

- $p = x^4 - x^3 - 7x^2 + 13x - 6$
- $q = x^3 - 5x^2 + 8x - 4$
- $x^3 + x^2 - 2x - 2$
- $x^3 + 3x^2 + x + 3$

Nuevas funciones

Factor, GaussianIntegers.

5.4. Mcd y mcm

Para calcular el *mcd* y el *mcm* con polinomios utilizamos **PolynomialGCD**[*p,q,...*] y **PolynomialLCM**[*p,q,...*].

Ejercicios

Calcula el máximo común denominador y el mínimo común múltiplo de los polinomios p y q anteriores.

Nuevas funciones

PolynomialGCD, PolynomialLCM.

5.5. Sustitución de variables

Para sustituir letras en el polinomio, utilizamos `/.`, escribiendo a continuación la sustitución, empleando un guión y el signo `>`, que se transformarán en una flecha. Esto es lo que en Mathematica se denomina una **regla**. Si queremos hacer varias sustituciones tenemos que poner las reglas en una lista.

Otro método consiste en transformar el polinomio en una función.

Ejercicios

Calcula el valor de $p(x)$ cuando x es 5 y cuando x es 1. Sustituye x por $y^2 - 1$.

5.6. Raíces de polinomios

Para obtener las raíces utilizamos **Roots[p==0,x]**. El polinomio debe estar igualado a cero (**se debe emplear un doble igual**), y como segundo argumento tenemos que escribir la variable. La solución, si es posible, es facilitada en forma exacta y con su multiplicidad. El comando encuentra tanto las raíces reales como las complejas.

Muchas veces tanta exactitud es contraproducente. En ese caso se puede utilizar **NRoots[p==0,x]**, que hace lo mismo, pero da la soluciones numéricas. La opción **PrecisionGoal** nos permite fijar el número de cifras de la solución.

Ejercicios

Encuentra las raíces, exactas o aproximadas, de los polinomios:

- $p = x^4 - x^3 - 7x^2 + 13x - 6$
- $x^3 - 5x^2 + 3x + 2$
- $4x^3 + 3x^2 + 2x + 1$

Nuevas funciones

Roots, NRoots, PrecisionGoal.

5.7. Polinomio interpolador de Lagrange

Fijados n puntos del plano, de tal forma que ningun par de ellos tenga la misma coordenada x (no puede estar un punto «encima» de otro), existe un polinomio de grado menor o igual que n que pasa por todos los puntos. Si los puntos estan en «posición general» el polinomio es de grado n . En particular, por dos puntos, pasa una recta, por tres puntos una parábola, ...

El comando para obtener el polinomio es **InterpolatingPolynomial[lista,x]**. La lista debe ser el conjunto de puntos por el que queremos que pase. Como segundo argumento escribimos la letra que queremos para el polinomio.

Ejercicios

Encontrar el polinomio interpolador, comprobando analíticamente que pasa por los puntos:

- Recta que pasa por (3,4) y (6,8).
- Parábola que pasa por (3,6), (2,7) y (4,7).
- Polinomio por los puntos (2,3), (3,4) y (4,5).

Nuevas funciones

InterpolatingPolynomial.

5.8. Números algebraicos y polinomio mínimo

Un número real o complejo a es un número algebraico si existe un polinomio $p(x)$ con coeficientes enteros tal que a es una raíz del polinomio. Si el número es algebraico, existen muchos polinomios que cumplen la condición. Pero existe un único polinomio de grado mínimo que lo cumple: este es el denominado polinomio mínimo del número a . Para conseguir el polinomio mínimo de a escribimos **MinimalPolynomial[a,x]**.

Para saber si un número es algebraico utilizamos el comando **Element[a, Algebraics]**.

Ejercicios

Calcula los polinomios mínimos, comprobando si son algebraicos, los siguientes números. Comprueba que efectivamente los números son raíces del polinomio mínimo:

$$a) 3/2 \quad b) \sqrt{3} \quad c) \sqrt[3]{5} + 8\sqrt{3} \quad d) \pi(?)$$

Nuevas funciones

MinimalPolynomial, Element, Algebraics.

6. Fracciones algebraicas

6.1. Simplificación

Los cocientes de polinomios son las fracciones algebraicas. Si queremos simplificar una fracción podemos **descomponer en factores** tanto el numerador como el denominador y «tachar» los factores repetidos. Otro método para simplificar consiste en dividir tanto el numerador como el denominador **entre el máximo común divisor de ambos polinomios**. Mathematica simplifica las fracciones con el comando **Cancel[p]** o también **Simplify[p]**.

Para acceder al numerador de una fracción tenemos **Numerator[p]** y para el denominador **Denominator[p]**.

Ejercicios

Simplifica las siguientes fracciones algebraicas:

$$a) \frac{x^2 - 5x + 6}{x - 2} \qquad b) \frac{x^2 - 5x + 6}{x^2 - 6x + 8}$$

Nuevas funciones

Cancel, Numerator, Denominator.

6.2. Desarrollar fracciones

El comando **Expand[p]** que nos servía para polinomios sigue funcionando con fracciones algebraicas. Sin embargo es probable que el resultado que nos devuelve no sea del agrado de los ojos de un matemático, a pesar de ser correcto.

En una fracción podemos desarrollar el numerador, el denominador o ambas partes. Los comandos son **ExpandNumerator[p]**, **ExpandDenominator[p]** y **ExpandAll[p]**.

Ejercicios

Desarrolla de distintos modos la fracción algebraica:

$$\frac{3x + (x + 4)^3}{(1 + 2x)^2}$$

Nuevas funciones

ExpandNumerator, ExpandDenominator, ExpandAll.

6.3. Operaciones

Si intentamos realizar operaciones con fracciones algebraicas, Mathematica nos da un resultado correcto, pero no se parece a lo que estamos acostumbrados en matemáticas. Si queremos obtener el resultado como una única fracción empleamos el comando **Together[p]**.

Ejercicios

Realiza la siguiente operación con fracciones algebraicas:

$$\frac{3+x}{2x+5} + \frac{5+x^2}{x^2+1} \cdot \frac{3}{x^2-2x+2}$$

Nuevas funciones

Together.

6.4. Factorización

Con **Factor[p]** también podemos factorizar fracciones algebraicas. Si algún factor está repetido se elimina tanto en el numerador como en el denominador.

Ejercicios

Factoriza las fracciones algebraicas:

$$a) \frac{x^2 - 1}{x^2 - 4}$$

$$b) \frac{x^2 - 1}{x^2 + 2x + 1}$$

6.5. Descomposición en fracciones simples

Una fracción simple tiene en el denominador un polinomio de grado uno elevado a una potencia o bien un polinomio de segundo grado, irreducible, y que puede estar elevado a cualquier potencia. El numerador puede ser una constante o un polinomio de segundo grado si en denominador existe (una potencia) un polinomio de segundo grado irreducible. Es conocido que **cualquier fracción algebraica se puede escribir como la suma de un polinomio y una serie de fracciones simples**. Esta descomposición es única. El comando que nos facilita la descomposición en fracciones simples es **Apart[p]**.

Ejercicios

Descomponer en fracciones simples:

$$a) \frac{7x+3}{x^2+3x-4} \qquad b) \frac{2x^2-x+4}{x^3+4x}$$

Nuevas funciones

Apart.

7. Ecuaciones y sistemas

- **En Mathematica el signo igual de las ecuaciones se escribe con dos signos iguales.**
- **Los sistemas de ecuaciones se escriben entre llaves y separados por comas.**

7.1. Los comandos fundamentales

El principal comando para resolver ecuaciones de modo exacto es **Solve[p==q,x]**. Si los números no tienen decimales, el programa intenta devolver el resultado exacto. **Si algún número es decimal, nos da la solución en formato decimal.**

Para transformar las soluciones en una lista utilizamos las reglas de sustitución.

A veces la solución exacta no nos es de mucha utilidad, pues puede contener gran cantidad de radicales. Si queremos la solución decimal utilizamos el comando **NSolve[p==q,x]**. Con la opción **WorkingPrecision** podemos fijar el número de cifras significativas de las soluciones (también lo podemos poner como tercer argumento).

El comando **Reduce[p==q,x]** se utiliza del mismo modo, pero a veces puede ser más útil que el comando **Solve**.

Ejercicios

Resuelve las siguientes ecuaciones de primer grado:

- $3x + 6 = 0$
- $\frac{3-x}{3} + \frac{4x-1}{7} = 4$
- $ax + b = 0$

Nuevas funciones

Solve, NSolve, Reduce, WorkingPrecision.

7.2. Ecuaciones algebraicas

Una ecuación es algebraica, si se puede reducir a **resolver un polinomio igualado a cero**. Desde el Renacimiento se conocen fórmulas exactas que nos permiten resolver **cualquier ecuación de grado igual o menor a cuatro**. Pero desde los trabajos de Abel, Ruffini y Galois sabemos que, en general, a partir del quinto grado no existen fórmulas semejantes para su resolución. Debido a ello, Mathematica no puede resolver de modo exacto dichas ecuaciones.

Mathematica calcula soluciones reales y complejas. Si la ecuación tiene alguna solución múltiple, aparecen las soluciones repetidas

Ejercicios

Resuelve las siguientes ecuaciones:

- $x^2 - 5x + 6 = 0$ (dos soluciones reales)
- $x^2 + 2x + 1 = 0$ (una solución doble)
- $x^2 - 4x + 5 = 0$ (soluciones complejas conjugadas)
- $ax^2 + bx + c = 0$
- $x^5 - 6x^4 + 10x^3 - 10x^2 + 9x - 4 = 0$
- $x^5 + x^4 + x^3 + x^2 + x + 4 = 0$

7.3. Sistemas con n° variables = n° ecuaciones

Los sistemas más elementales tienen el **mismo número de ecuaciones que de incógnitas**. Mathematica puede resolver sistemas lineales y saber si son **compatibles** o **incompatibles**. Si el sistema es **indeterminado** también nos da el conjunto de soluciones. Además puede resolver sistemas no lineales de cualquier dimensión. También podemos utilizar **NSolve** para obtener las soluciones aproximadas.

Si el número de variables es igual al número de ecuaciones, podemos omitir el segundo argumento y no escribir las variables.

Ejercicios

Resuelve los sistemas, de modo exacto y aproximado:

$$\blacksquare \begin{cases} 3x + 7y = 9 \\ -5x + 9y = 5 \end{cases} \quad (\text{una única solución})$$

$$\blacksquare \begin{cases} x + y = 0 \\ x + y = 1 \end{cases} \quad (\text{sin solución})$$

$$\blacksquare \begin{cases} x + y = 9 \\ 2x + 2y = 18 \end{cases} \quad (\text{infinitas soluciones})$$

$$\blacksquare \begin{cases} x^2 - 46x = 8 \\ -6x + 7y = -3 \end{cases} \quad (\text{sistema no lineal})$$

7.4. Sistemas con más incógnitas que ecuaciones

En este caso, como tenemos más letras que ecuaciones, debemos indicarle a Mathematica que letras son las que queremos resolver. Si no le decimos nada, Mathematica toma esa decisión por nosotros.

Ejercicios

Resuelve para distintas incógnitas

$$\begin{cases} 6x + 9y - 3z = 8 \\ 3x - 8y + 9z = 7 \end{cases}$$

7.5. Otros tipos de ecuaciones algebraicas

Algunas ecuaciones, en principio, no parecen ecuaciones algebraicas, pues en ellas aparecen radicales, fracciones algebraicas, ... Sin embargo, en algunos casos, se pueden transformar en ecuaciones algebraicas «casi» equivalentes. Lo de casi equivalentes significa que con esos procedimientos pueden aparecer soluciones de la ecuación algebraica que no son soluciones de la ecuación de partida. Las mas simples son las ecuaciones con radicales cuadráticos. Si dejamos sola una raíz y elevamos al cuadrado, vamos eliminando uno a uno todos los radicales. También entran en este bombo las ecuaciones con fracciones algebraicas: mediante operaciones algebraicas se reduce a una única fracción algebraica y después se «resuelve el numerador».

Ejercicios

Resuelve:

$$\blacksquare 1 + \sqrt{3x+4} = x+1$$

$$\blacksquare \frac{6}{x^2-1} - \frac{2}{x-1} = 2 - \frac{x+4}{x-1}$$

7.6. Ecuaciones no algebraicas

La mayor parte de las ecuaciones no algebraicas (transcendentes) **no se pueden resolver por métodos exactos**. Para resolver dichas ecuaciones se emplean métodos numéricos. En general, los métodos numéricos no encuentran todas las soluciones, sino que tenemos que darle una «semilla» y el método encontrará una solución próxima a la semilla.

El comando para resolver sistemas por métodos numéricos es **FindRoot[p==q,{x,s}]**. Además de la ecuación y la variable, debemos proporcionarle una semilla. Tanto la variable como la semilla van en el segundo argumento en forma de lista. Con la opción de **WorkingPrecision** tendremos mayor precisión. Este comando se puede utilizar también para sistemas, pero ahora debemos fijar una semilla para cada variable.

Ejercicios

- Resuelve las ecuaciones con distintas semillas.

$$a) x^2 - 1 = 0 \quad b) \cos(x^2) - 6x = \tan(x)$$

- Resuelve el sistema con semillas $x = 1$, $y = 2$.

$$\begin{cases} y = \exp(x) \\ x + y = 2 \end{cases}$$

Nuevas funciones

FindRoot.

7.7. Inecuaciones

El comando **Reduce[p,x]** se puede utilizar para ecuaciones en lugar del comando **Solve**. Sin embargo el comando **Solve** no se puede utilizar para inecuaciones, que se resuelven con **Reduce**. El resultado de la inecuación aparece escrito como una expresión booleana, que debemos traducir al lenguaje de los intervalos.

Le expresión booleana está escrita en el lenguaje de Mathematica. Con el comando **TraditionalForm** podemos escribirla en un lenguaje más matemático.

Ejercicios

Resuelve las inecuaciones:

- $4x - 5 > 9$
- $x^2 - 5x + 6 \geq 0$
- $\frac{x^2 - 5x + 6}{x^2 - 6x + 8} \geq 0$

Nuevas funciones

TraditionalForm.

7.8. El método de sustitución

Uno de los primeros métodos que se estudian para resolver sistemas es el método de sustitución. Si tenemos dos ecuaciones, despejamos una letra en una de las ecuaciones y sustituimos en la otra. En Mathematica podemos hacer esto con el comando **Eliminate[p==q,x]**. Le facilitamos el sistema y la letra que queremos eliminar.

Cada vez que eliminamos una letra se elimina una ecuación. Podemos eliminar varias letras a la vez, situando la lista de ellas como segundo argumento.

Ejercicios

Eliminar cada una de las letras en el sistema:

$$\begin{cases} 3x + 7y = 9 \\ -5x + 9y = 5 \end{cases}$$

Nuevas funciones

Eliminate.

8. Números complejos

8.1. La unidad imaginaria

La unidad imaginaria se denota por **I** (mayúscula). Para introducir un número complejo lo hacemos en la forma binómica **a+b*I** o bien empleando la función **Complex[a,b]**. Muchas veces podemos suprimir el asterisco que separa al número de la unidad imaginaria. En la salida, la unidad imaginaria se escribe en minúscula y con una tipografía especial.

Ejercicios

- Calcula la raíz cuadrada de -1.
- Resuelve la ecuación $x^2 + 1 = 0$.
- Calcula distintas potencias de la unidad imaginaria.

Nuevas funciones

I , Complex.

8.2. Operaciones

Las operaciones algebraicas se realizan con los operadores de siempre.

Para calcular el módulo de un número complejo empleamos **Abs[z]**, para obtener el argumento (en radianes) **Arg[z]**, para el conjugado **Conjugate[z]**, y para las partes real e imaginaria **Re[z]** e **Im[z]**.

Ejercicios

- Dados los números complejos $z = 3 + 5i$ y $w = 3 - 2i$, calcula su suma, su resta, multiplicación, división, inverso y potencias.
- Calcula el módulo, el argumento, el conjugado, la parte real e imaginaria de w .

Nuevas funciones

Abs, Arg, Conjugate, Re, Im.

8.3. Forma polar

Si un número complejo z tiene módulo m y argumento θ (en radianes), entonces, **según la fórmula de Euler**, el número se puede escribir como $z = m \cdot \exp(i\theta)$. También se puede escribir como $z = m \cos(\theta) + im \sin(\theta)$.

Si tenemos un número complejo en forma polar y lo queremos en forma binómica, utilizamos la función **ComplexExpand**.

Ejercicios

Utilizando la fórmula de Euler, construye el número complejo 5_{30° y calcula el conjugado, la parte real, la imaginaria y el módulo.

Nuevas funciones

ComplexExpand.

8.4. Funciones con argumento complejo

Un número entero z tiene siempre n raíces de orden n . Para encontrar todas las raíces de dicho orden debemos resolver la ecuación $x^n - z = 0$. Si nosotros elevamos el número z a la potencia $1/n$ **únicamente obtenemos una de las posibles raíces**.

Muchas de las funciones elementales se pueden aplicar sobre argumentos complejos. En general dichas funciones son multivaluadas (en realidad dan lugar a superficies de Riemann) y Mathematica generalmente nos devuelve un único valor.

Ejercicios

- Calcula distintas raíces del número $3 + 5i$.
- Calcular la raíz cúbica de -8 .
- Realiza los siguientes cálculos, dando el resultado en forma binómica:

$$a) e^{\pi i} \quad b) \log(4 + 5i) \quad c) \arcsin(3) \quad d) \exp(ix)$$

9. Gráficas de funciones

9.1. Gráfica de una o más funciones

Para dibujar funciones usamos `Plot[f[x],[x,min,max]]`. Debemos escribir una función con una única variable, y en una lista debemos dar la variable, el valor mínimo y el valor máximo del eje horizontal. El tamaño del eje vertical se adapta a la función que estemos dibujando.

Para dibujar **más de una gráfica en los mismos ejes**, debemos utilizar la misma variable para las dos funciones y escribirla como primer argumento en una lista.

Las gráficas se pueden **redimensionar y guardar en multitud de formatos**, utilizando el botón derecho del ratón.

Ejercicios

Dibujar la siguientes funciones individualmente, variando el dominio. Dibujar ambas funciones en los mismos ejes:

$$f(x) = \sin(x)$$

$$g(x) = x^2 - 3x$$

Nuevas funciones

Plot.

9.2. Algunas opciones interesantes

El comando **Plot** tiene una gran cantidad de opciones (se pueden ver todas con el comando **Options[f]**) que nos permiten variar muchos detalles del gráfico. Entre las más interesantes están:

- **AspectRatio**. Varía la proporción entre la altura y la anchura. Por defecto dibuja gráficas con la relación aurea. Si ponemos **Automatic** las escalas de ambos ejes se igualan.
- **PlotRange**. En una lista escribimos el valor mínimo y máximo del eje vertical.
- **PlotStyle**. Puede modificar muchos aspectos de la gráfica, en particular su color. Debemos emplear nombres de colores en inglés.

Ejercicios

Dibujar la función x^2 en el intervalo $(-3, 3)$ y variar las opciones anteriores.

Nuevas funciones

Options, AspectRatio, PlotRange, PlotStyle.

9.3. El comando **Show**

El comando **Show[a]** permite mostrar varios objetos gráficos en unos mismos ejes. En particular esto nos puede servir para dibujar varias gráficas de funciones, cada una en un dominio determinado. Ellos nos permite dibujar funciones definidas a trozos de una manera muy sencilla. Necesitamos la opción **PlotRange** en **All** para que se vean todas las gráficas.

Mejor que esto es definir directamente una función definida a trozos, con el comando **Piecewise[lista]**. Al utilizar **Piecewise** si algún punto del dominio queda sin definición, se le asigna el valor cero.

Ejercicios

Dibujar la función definida a trozos, utilizando el comando **Show** y también con el comando **Piecewise**:

$$f(x) = \begin{cases} x^2 & \text{si } -2 < x \leq 0 \\ \sin(x) & \text{si } 0 < x < 2 \end{cases}$$

Nuevas funciones

Show, Piecewise.

9.4. Gráficos de funciones implícitas

Para dibujar funciones implícitas empleamos

ContourPlot[p==q,{x,xmin,xmax},{y,ymin,ymax}]

El primer argumento es una ecuación en las variables x e y y el segundo y tercer argumento indican el «trozo» de plano en el que dibujamos la función.

Puede ser necesario tener las opciones **Axes->True** y también **AspectRatio->Automatic** para que no se deforme la gráfica.

Ejercicios

Realiza los gráficos de las siguientes funciones implícitas:

- $2x + y = 3$ (una recta)
- $x^2 + y^2 = 1$ (una circunferencia)
- $x^2 + 3y^2 = 1$ (una elipse)
- $y^2 = x^3 - x$ (curva elíptica)

Nuevas funciones

ContourPlot, Axes.

10. Límites

10.1. Noción intuitiva de límite

Para calcular límites se utiliza la función **Limit[f,x → x₀]**. Esta función puede admitir 3 argumentos, siendo el último opcional. El primer argumento debe ser la función a la que le queremos calcular el límite. En el segundo escribimos la variable y una flecha (guión y signo >) dirigida al punto. En el tercero podemos utilizar **Direction**: si apunta al 1 entonces el límite se calcula por la izquierda y si apunta a -1 lo calcula por la derecha. Debemos tener en cuenta que Mathematica siempre calcula, por defecto, límites por la derecha, así que algún límite puede no existir, por no coincidir con el límite por la izquierda. El símbolo ∞ se escribe **Infinity**.

Ejercicios

- Consideremos la función $f(x) = \frac{\sin(x)}{x}$. Calcula valores de la función para elementos próximos a cero, dibuja la función en un entorno del cero y calcula el límite.
- Haz lo mismo con la función $f(x) = \frac{|x|}{x}$.

Nuevas funciones

Limit, Direction, Infinity.

10.2. Indeterminaciones

Mathematica calcula límites en cualquier punto. En particular es capaz de resolver muchos tipos de indeterminaciones.

Ejercicios

Resuelve las siguientes indeterminaciones:

$$\lim_{x \rightarrow 2} \left(\frac{x^2 - 5x + 6}{x^2 - 6x + 8} \right) \rightsquigarrow (\text{tipo } \infty/\infty)$$

$$\lim_{n \rightarrow \infty} \left(1 + \frac{x}{n} \right)^n \rightsquigarrow (\text{tipo } 1^\infty)$$

$$\lim_{x \rightarrow 16} \left(\frac{x - 16}{\sqrt{x} - 4} \right) \rightsquigarrow (\text{tipo } 0/0)$$

10.3. Derivadas como límites

Las derivadas se pueden calcular con límites. Para calcular la derivada de una función f en un punto a debemos efectuar el límite:

$$\lim_{h \rightarrow 0} \frac{f(a+h) - f(a)}{h}$$

Ejercicios

Calcula, utilizando límites, la derivada del coseno.

10.4. Continuidad

Para que una función sea continua, al menos, **deben existir los límites laterales y coincidir**. Si el valor de la función coincide con estos valores, la función es continua. Si el valor de la función no coincide con los límites entonces, redefiniendo la función, podemos hacerla continua. Se dice en este caso que la discontinuidad es evitable.

Ejercicios

Comprueba la continuidad en $x = 0$ de las funciones:

$$f(x) = \frac{1}{x}$$

$$g(x) = (2 + x)^{\frac{1}{x}}$$

10.5. Asíntotas

Dada una función $f(x)$, decimos que $y = a$ es una **asíntota horizontal** (por la derecha) si $\lim_{x \rightarrow \infty} f(x) = a$. Una recta $x = a$ es una **asíntota vertical** si alguno de los límites laterales $\lim_{x \rightarrow a} f(x)$ es infinito. Finalmente, una recta $y = mx + n$ es una **asíntota oblicua** si $\lim_{x \rightarrow \infty} (f(x) - (mx + n)) = 0$. En todos los casos, la gráfica de la función y la de la asíntota se acercan entre si.

Ejercicios

Calcula las asíntotas y dibuja la función.

$$\frac{2x^2 - 6x + 3}{x - 4}$$

11. Derivadas

11.1. Cálculo de derivadas

Para derivar se utiliza el comando **D[f,x]**. Debemos escribir la función y como segundo argumento la variable. **Si la función tiene varias variables calcula las derivadas parciales.**

También se puede definir una nueva función y **utilizar la comilla simple** para calcular la derivada. Este método sirve únicamente para funciones con una sola variable.

Es importante que la variable que empleemos no tenga ningún valor asignado. Si lo tiene lo eliminamos con **Clear[var]**.

Ejercicios

Calcula la derivada de las funciones:

$$a) 3x^2 + 8x \quad b) 4y^5 - 6y^3 \quad c) \cos(y \cdot x^2) \quad d) x^n$$

Grafica alguna de las derivadas anteriores, utilizando el comando **Evaluate[expr]** si es necesario.

Nuevas funciones

D, Evaluate, Clear.

11.2. Derivadas de orden superior

Para realizar derivadas de orden superior debemos escribir una lista, donde especificamos la variable y el orden de derivación. Se puede derivar de este modo sobre varias variables. También podemos escribir repetidamente la variable sobre la que queremos hacer la derivada de orden superior a 1.

Con la función **HoldForm[expr]** podemos ver en notación matemática lo que hemos tecleado, pero no le ejecuta.

Ejercicios

Calcula las siguientes derivadas:

$$a) \frac{\partial^5}{\partial x^5} \sin(6x) \quad b) \frac{\partial^5}{\partial x^2 \partial y^3} \sin(4x + 6y)$$

Nuevas funciones

HoldForm.

11.3. Derivadas simbólicas

Mathematica puede calcular derivadas simbólicas. Si trabajamos con funciones que no tienen ninguna expresión asignada y operamos con ellas, Mathematica nos calcula la derivada aplicando las reglas de derivación.

Ejercicios

Calcular la derivada de un producto y de una división de funciones.

11.4. Cálculo de tangentes a curvas

La primera aplicación de las derivadas, y su origen histórico, está en el cálculo de tangentes a curvas. **Para calcular la tangente a una curva en un punto, debemos conocer la derivada de la función en dicho punto.** Esto nos da la pendiente de la recta y como conocemos el punto, podemos establecer la ecuación de la recta.

Ejercicios

Calcula la tangente a $f(x) = x^2 + 3x$ en el punto $x = 2$ y dibuja ambas gráficas.

11.5. Extremos y puntos de inflexión

Otra de las aplicaciones de la derivada consiste en el cálculo de extremos, así como de los puntos de inflexión de curvas. Para ello debemos realizar la primera o la segunda derivada, igualar a cero y resolver la ecuación. Los extremos y los puntos de inflexión se encuentran entre las soluciones.

Ejercicios

Encuentra los posibles extremos y los puntos de inflexión de:

$$x^3 - 6x + 7$$

Dibuja conjuntamente la función y su derivada, para comprobar los extremos.

11.6. Series de potencias

Toda función que sea suficientemente derivable en el entorno de un punto **admite un desarrollo en serie de Taylor**. Si la función es analítica, dicho polinomio converge (en un sentido apropiado) a la función, pero en todos los casos la gráfica de la función y del polinomio se «aproximan» en el entorno del punto.

Si queremos obtener el polinomio de Taylor empleamos la función **Series[f,{x,x₀,n}]** . En una lista debemos dar la variable, el punto y el orden del desarrollo.

Para eliminar el resto se utiliza la función **Normal[serie]**. Ya con el resto eliminado se puede definir una función.

Ejercicios

- Calcula desarrollos de distintos ordenes de las funciones:

$$a) \exp(x) \quad b) \cos(x^2)$$

- Dibuja una función y su polinomio de Taylor en un entorno conveniente.

Nuevas funciones

Series, Normal

12. Integrales

12.1. Integrales indefinidas

Mathematica puede calcular tanto integrales definidas como indefinidas en modo exacto con el comando **Integrate[f,x]**. Para calcular la primitiva, solamente tenemos que escribir la función y como segundo argumento la variable.

Es conocido en matemáticas que no todas las integrales se pueden expresar como una expresión con funciones elementales. Por ello Mathematica puede no saber hacer algunas integrales o dar algunos resultados «sorprendentes» en algunas integrales.

Ejercicios

Calcula las siguientes integrales indefinidas, derivando para comprobar algún resultado:

▪ a) $\int x^5$ b) $\int 4y^3 - 5y$ c) $\int x \cos(x)$

▪ a) $\int \exp(-x^2)$ b) $\int \frac{\sin(x)}{x}$

Nuevas funciones

Integrate.

12.2. Integrales definidas

Para las integrales definidas usamos **Integrate[f,{x,a,b}]**.

A las integrales definidas les pasa lo mismo que a las definidas. En principio, si tenemos una primitiva, basta sustituir los extremos para calcular la integral definida. **El problema es que muchas funciones no tienen primitiva, expresable en forma cerrada.** Pero en este caso se puede recurrir a la **integración numérica**, empleando el comando **NIntegrate[f,{x,a,b}]**. Para fijar el número de cifras significativas tenemos la opción **WorkingPrecision**.

Ejercicios

Calcula las integrales definidas:

■ a) $\int_0^4 4x^3 - 4x$ b) $\int_0^{\pi/2} \cos(x)$

■ a) $\int_0^4 \frac{\sin(x)}{x}$ $\int_0^{\pi} \sin(\sin(x))$

Nuevas funciones

NIntegrate.

12.3. Integrales impropias

Las integrales impropias se caracterizan por alguno de estos dos «problemas».

- Alguno de los límites es infinito.
- La función no está acotada en el intervalo de integración.

Mathematica también pueda calcular integrales impropias. Si la integral no converge nos informa de ello en la solución.

Ejercicios

Calcula las integrales impropias:

$$a) \int_1^{\infty} \frac{1}{x} \quad b) \int_0^6 \frac{1}{\sqrt{x}} \quad c) \int_{-\infty}^{\infty} e^{-x^2} \quad d) \int_{-2}^2 \frac{1}{x^2}$$

12.4. Integrales de fracciones algebraicas

Las integrales de fracciones algebraicas se pueden realizar descomponiendo en fracciones simples con el comando `Apart[frac]` e integrando cada término. Naturalmente Mathematica también puede realizar la integral de manera directa.

Ejercicios

Calcula la integral de la fracción algebraica. Descompón en fracciones simples e integra:

$$\int \frac{5x + 3}{x^2 + 2x - 3}$$

12.5. Integrales dependientes de parámetros

Si una función tiene varias variables e integramos con respecto a una de ellas, se dice que estamos haciendo **una integral con parámetros**. No se debe confundir este concepto con la integración en varias variables.

Ejercicios

Calcula la integral:

$$\int A \cdot \cos(\omega t) dt$$

12.6. Cálculo de áreas

La primera aplicación de las integrales es el **cálculo de áreas**. Si la función es siempre positiva, la integral definida es el área comprendida entre la gráfica de la función y el eje x . **Si la función tiene partes positivas y negativas, tenemos que hacer varias integrales**, siendo los límites de integración los puntos de corte con el eje x . Siguiendo ideas similares se puede calcular el área entre dos curvas.

Ejercicios

- Calcular el área del seno en el intervalo $(0, \pi)$.
- Calcular el área entre las curvas $y = 4x$ e $y = x^2$.

12.7. Sumas de series numéricas

Sabemos que las integrales definidas son en realidad sumas a las que se les hace posteriormente un límite. La suma de series es ciertamente similar: se realiza una suma parcial y posteriormente se toma un límite. En realidad, si tomamos una medida adecuada sobre el conjunto de los naturales, **una serie no es más que un ejemplo de integral de Lebesgue**.

Dada una sucesión a_n podemos sumar algunos de sus términos o también sumar un número infinito de términos (la suma de la serie). El comando que empleamos es **Sum[a,{i,imin,imax}]**. En el último término podemos poner ∞ y sumar la serie entera. Si la serie es divergente, entonces Mathematica nos informa de ello.

Ejercicios

- Realizar las siguientes sumas:

$$a) \sum_1^{100} n \quad b) \sum_1^n i \quad c) \sum_1^n i^2 \quad d) \sum_1^{\infty} \frac{1}{n^2}$$

- Estudiar las sumas parciales y la convergencia de la serie armónica.

Nuevas funciones

Sum.

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 realizar 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 $\frac{u \cdot v}{v}$. 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.
- $u \cdot u = |u|^2$.
- 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.

14. Matrices

14.1. Introducción de matrices

Para introducir una matriz escribimos unas llaves y dentro, separados por comas, escribimos cada uno de los vectores fila. Como las filas son vectores deben ir entre llaves también. En realidad **una matriz es una lista de listas**. Mathematica siempre nos presenta la salida de una matriz en la forma anterior. Si queremos ver los resultados con la presentación habitual debemos utilizar **MatrixForm[A]**. En el caso de los vectores estos se representan como vectores columna.

Ejercicios

Escribe las matrices:

$$A = \begin{pmatrix} 5 & 7 & 8 \\ -3 & 4 & 2 \\ 4 & 7 & 9 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

Nuevas funciones

MatrixForm.

14.2. Operaciones matriciales

Los operadores habituales realizan las **operaciones componentes a componente**. En el caso de la suma y la resta corresponden a la estructura de espacio vectorial del conjunto de matrices de un determinado orden. Sin embargo debemos tener en cuenta que **el asterisco no proporciona el producto matricial, ni el circunflejo la potencia matricial**. Para el **producto matricial se emplea el punto** (o también la función **Dot[m,n]**) y para la potencia **MatrixPower[A,n]**.

Ejercicios

Con las matrices anteriores realiza las operaciones:

- $a) A + B$ $b) A - B$ $c) A * B$ $d) A.B$
- $a) B.A$ $b) 3A + 5B$ $c) A^5$ $d) A^{-1}$

Nuevas funciones

Dot, MatrixPower.

14.3. Inversas, determinantes y transpuestas

La inversa de una matriz se puede calcular elevando a -1 la matriz o más cómodamente con el comando **Inverse[A]**. El determinante con **Det[A]**, la traza con **Tr[A]** y la transpuesta con **Transpose[A]**.

Ejercicios

- Calcula los comandos anteriores sobre las matrices A y B .
- Comprueba la propiedad $u \cdot (v \times w) = \det(u, v, w)$.

Nuevas funciones

Inverse, Det, Tr, Transpose.

14.4. Rango y reducción por filas

El rango se calcula con **MatrixRank[A]**. Este es el número de filas linealmente independientes. Podemos también reducir por filas la matriz con la orden **RowReduce[A]** y contar el número de filas no nulas.

Ejercicios

Calcula el rango y la reducción por filas de las matrices anteriores.

Nuevas funciones

MatrixRank, RowReduce.

14.5. Matrices y vectores

Si tenemos un vector, con las dimensiones adecuadas, **se puede multiplicar por la derecha o por la izquierda por una matriz**. El caso de multiplicar por ambos lados se obtiene, desde el punto de vista algebraico una forma cuadrática, y desde el punto de vista geométrico el producto escalar asociado a la matriz en cuestión.

Ejercicios

Dada la matriz A y el vector $v = (3, 6, -1)$ calcula:

$$a) A \cdot v$$

$$b) v \cdot A$$

$$c) v \cdot A \cdot v$$

14.6. Construcción de matrices especiales

Mathematica tiene algunas funciones que nos permiten construir algunas matrices especiales:

- **IdentityMatrix[n]** \Rightarrow matriz identidad de orden n .
- **DiagonalMatrix[v]** \Rightarrow matriz cuadrada con el vector v en la diagonal.
- **ConstantArray[a,{m,n}]** \Rightarrow matriz $m \times n$ constante.
- **RandomInteger[a,{m,n}]** \Rightarrow matriz $m \times n$ formada por números aleatorios entre 1 y a .

Ejercicios

- Construye la matriz identidad de orden 4.
- Una matriz cuadrada con la diagonal (4,7,9).
- Utilizar el segundo argumento de **DiagonalMatrix**.
- La matriz nula de tamaño 3×4 .
- Crea una matriz aleatoria.

Nuevas funciones

IdentityMatrix, DiagonalMatrix, ConstantArray, RandomInteger.

14.7. Submatrices

Introduciremos con ejemplos, la manera de extraer submatrices de una matriz. Para ello debemos emplear el doble corchete o las funciones **Take[m]** o **Drop[m]**:

- **A[[2,3]]** extrae el elemento a_{23} de la matriz.
- **A[[4]]** extrae la cuarta fila de la matriz.
- **A[[All, 2]]** nos devuelve la segunda columna.
- **A[[{1,2,4},All]]** extrae las filas 1, 2 y 4.
- **Take[A,{2,6},{3,5}]** submatriz con las filas de la 2 a la 6 y con las columnas de la 3 a la 10
- **Drop[A,{2,6},{3,5}]** submatriz que se obtiene al borrar las filas de la 2 a la 6 y borrar las columnas de la 3 a 10.

Ejercicios

Construye una matriz aleatoria y extrae diversas submatrices con los comandos anteriores.

Nuevas funciones

Take, Drop.

14.8. Miscelanea de matrices

He aquí una recopilación de funciones aplicables a matrices:

- **\$Post:=If[MatrixQ[#], MatrixForm[#],#]&** \Rightarrow todas las matrices se muestran en formato matemático.
- **Dimensions[m]** \Rightarrow tamaño de la matriz.
- **HermitianMatrixQ[m]** \Rightarrow True si $A = A^*$.
- **OrthogonalMatrixQ[m]** \Rightarrow True si $A \cdot A^t = \text{Id}$.
- **PositiveDefiniteMatrixQ[m]** \Rightarrow True si es definida positiva.
- **SymmetricMatrixQ[m]** \Rightarrow si $A = A^t$.
- **Diagonal[m]** \Rightarrow extrae la diagonal de m .

15. Sistemas de ecuaciones lineales

15.1. El comando **Solve**

Ya hemos visto como el comando **Solve** puede resolver todo tipo de sistemas, en particular, puede resolver sistemas de ecuaciones lineales. Las ecuaciones, formando una lista, son el primer argumento. Una lista de las variables es el segundo argumento. También podemos emplear el comando **NSolve** si queremos aproximaciones numéricas.

Ejercicios

- Resuelve los sistemas con el comando **Solve**.

$$SCD \quad \left\{ \begin{array}{rclcl} x & +y & +4z & = & 25 \\ 2x & +y & & = & 7 \\ -3x & & +z & = & -1 \end{array} \right.$$

$$SCI \quad \left\{ \begin{array}{rclcl} x & +y & +4z & = & 25 \\ 2x & +y & & = & 7 \\ x & +y & +4z & = & 25 \end{array} \right.$$

$$SI \quad \left\{ \begin{array}{rclcl} x & +y & +4z & = & 25 \\ 2x & +y & & = & 7 \\ x & +y & +4z & = & 90 \end{array} \right.$$

15.2. El comando **RowReduce**

Si escribimos la matriz ampliada del sistema y la reducimos por filas, nos queda **un sistema equivalente, donde las soluciones se pueden ver «a ojo»**. También se puede ver de manera cómoda si el sistema es compatible indeterminado (si tiene una fila llena de ceros) o si el sistema es incompatible (en una fila todos son ceros menos el último número).

Ejercicios

- Resuelve los mismos sistemas con **RowReduce**

$$\left\{ \begin{array}{rrcr} x & +y & +4z & = & 25 \\ 2x & +y & & = & 7 \\ -3x & & +z & = & -1 \end{array} \right. \quad \left\{ \begin{array}{rrcr} x & +y & +4z & = & 25 \\ 2x & +y & & = & 7 \\ x & +y & +4z & = & 25 \end{array} \right.$$

$$\left\{ \begin{array}{rrcr} x & +y & +4z & = & 25 \\ 2x & +y & & = & 7 \\ x & +y & +4z & = & 90 \end{array} \right.$$

- Resuelve un sistema generado al azar.

Nuevas funciones

RowReduce.

15.3. Multiplicar por la matriz inversa

Si tenemos un sistema con el **mismo número de ecuaciones que de incógnitas**, lo escribimos en forma matricial $A \cdot X = B$, donde la matriz A es cuadrada. Si la matriz A es invertible, multiplicando por la izquierda por la inversa de A , obtenemos la solución del sistema. En esencia este es el conocido **método de Cramer**. Solamente se puede aplicar si la matriz del sistema es cuadrada y no singular, por lo que el sistema tiene una única solución.

Ejercicios

Resolver el sistema con la matriz inversa:

$$\begin{cases} x + y + 4z = 25 \\ 2x + y = 7 \\ -3x + z = -1 \end{cases}$$

15.4. El comando **LinearSolve[A,B]**

Si A es la matriz del sistema y B el vector de términos independientes, con **LinearSolve[A,B]** obtenemos **una solución del sistema**, en el caso de que el sistema tenga solución. Si el sistema tiene infinitas soluciones, además de una solución particular, que es lo que nos aporta el comando **LinearSolve**, necesitamos resolver la ecuación homogénea. La ecuación homogénea se resuelve con el comando **NullSpace[m]**. Si el comando **NullSpace** nos devuelve un conjunto de vectores, la solución general es la suma de la particular y la solución de la homogénea.

Ejercicios

Resolver los mismos sistemas con los nuevos comandos:

$$\begin{cases} x & +y & +4z & = & 25 \\ 2x & +y & & = & 7 \\ -3x & & +z & = & -1 \end{cases} \quad \begin{cases} x & +y & +4z & = & 25 \\ 2x & +y & & = & 7 \\ x & +y & +4z & = & 25 \end{cases}$$

$$\begin{cases} x & +y & +4z & = & 25 \\ 2x & +y & & = & 7 \\ x & +y & +4z & = & 90 \end{cases}$$

Nuevas funciones

LinearSolve, NullSpace.

16. Estadística descriptiva

16.1. Los distintos tipos de media

Dado un conjunto de datos, en forma de vector, pretendemos dar **un número que resuma dicha información**. Con **Mean[v]** tenemos la **media aritmética**. El comando **Total[v]** nos permite sumar todos los elementos y **Lenght[v]** nos dice cuantos tiene.

Además de la media aritmética, existen muchas otras: La **media armónica (HarmonicMean[v])**, la **media geométrica (GeometricMean[v])**, la **media contraarmónica (ContraharmonicMean[v])** y la **media cuadrática (RootMeanSquare[v])**. Sus formulas son:

$$h) n : \left(\sum \frac{1}{x_i} \right) \quad g) \left(\prod x_i \right)^{1/n} \quad c) \frac{\sum x_i^2}{\sum x_i} \quad r) \sqrt{\frac{1}{n} \sum x_i^2}$$

Ejercicios

Calcula distintas medias con los datos

21,22,21,23,22,21,23,23,21,21,21,24,22

Nuevas funciones

Mean, Total, Lenght, HarmonicMean, GeometricMean, ContraharmonicMean, RootMeanSquare.

16.2. Mediana y moda

La **mediana** de un conjunto de datos es el dato que se encuentra en el punto medio, una vez ordenados los datos. Si el número de datos es par, entonces es la media de los dos datos centrales. Con **Sort[v]** podemos ordenar los datos del vector y extraer los valores con el doble corchete. También podemos emplear **Median[v]** que nos proporciona directamente la mediana.

El dato que más se repite (el más común) es la **moda**. En Mathematica tenemos el comando **Commonest[v]** para calcular la moda.

Ejercicios

Calcular la mediana de los datos anteriores. Ordenarlos y ver que coincide con el valor situado en el punto medio. Calcular la moda.

Nuevas funciones

Sort, Median, Commonest.

16.3. Máximo, mínimo y cuantiles

El **máximo** de los datos se obtiene con **Max[v]** y el **mínimo** con **Min[v]**. Restando ambos podemos obtener el **rango**. Los tres **cuantiles** con **Quartiles[v]**. El **rango intercuartílico** (la resta del tercer y primer cuartil) con **InterquartileRange[v]**. Finalmente los **percentiles** se calculan con **Quantile[v,p]**, donde además de los datos, debemos facilitar el percentil, como un número entre 0 y 1.

Ejercicios

Calcula el máximo, el mínimo, el rango, los cuantiles, el rango intercuartílico y algún percentil de los datos anteriores.

Nuevas funciones

Max, Min, Quartiles, InterquartileRange, Quantile.

16.4. Varianza y desviación típica

La **varianza** (**Variance[v]**) de un conjunto de datos es:

$$\frac{\sum (x_i - \bar{x})^2}{n - 1}$$

La varianza que calcula Mathematica es la que esta dividida en el número de datos **menos 1**, que muchas veces se encuentra en la literatura con el nombre de cuasivarianza.

La función (**StandardDeviation[v]**) nos da **desviación estándar** y se calcula aplicando la raíz cuadrada a la varianza.

La **desviación media** (**MeanDeviation[v]**) se calcula:

$$\frac{1}{n} \sum |x_i - \bar{x}|$$

Ejercicios

Calcula los parámetros al conjunto de datos.

Nuevas funciones

Variance, StandardDeviation, MeanDeviation.

16.5. Asimetría, kurtosis y momentos

La asimetría y la kurtosis son una medida de la posible asimetría de los datos y del posible apuntamiento de la distribución. Los comandos son **Skewness[v]** y **Kurtosis[v]**.

Para calcular el momento de orden r tenemos el comando **Moment[v,r]** y para el momento central **CentralMoment[v,r]**.

Ejercicios

Calcula la asimetría, la kurtosis y algún momento de los datos.

Nuevas funciones

Skewness, Kurtosis, Moment, CentralMoment.

16.6. Covarianza y correlación

Dados dos vectores de datos u y v de la misma dimensión, el comando **Correlation[u,v]** calcula la correlación entre ambas variables y **Covariance[u,v]** calcula la covarianza.

Ejercicios

Construir dos vectores aleatorios de la misma dimensión y calcular la correlación y la covarianza.

Nuevas funciones

Correlation, Covariance.

17. Gráficos estadísticos

17.1. Tablas de frecuencias discretas

Si tenemos una distribución discreta, la función **Tally[v]** nos permite hacer una tabla de frecuencias. El resultado es una matriz, cuya primera columna son los datos y la segunda columna son las frecuencias absolutas. El comando **Sort[v]** nos ordena la tabla. Con los dobles corchetes podemos extraer la segunda columna. Para dibujar el diagrama de barras utilizamos **BarChart[v]** y para el diagrama de sectores **PieChart[v]** y sus versiones tridimensionales.

Ejercicios

Genera un vector de 100 números aleatorios entre el 1 y el 6. Haz la tabla de frecuencias, ordénala y representa el diagrama de barras y el de sectores asociado.

Nuevas funciones

Tally, BarChart, PieChart, BarChart3D, PieChart3D.

17.2. Recuento de clases e histogramas

Si tenemos una distribución de datos continua, debemos dividir los datos en clases y contar la frecuencia de cada clase. Ello se consigue con **BinCounts[list,{a,b,inc}]**.

Con la orden **Histogram[list]** se crea un histograma de nuestros datos. El programa decide cuantas clases hacer.

El comando **SmoothHistogram[list]** «suaviza» el histograma.

Ejercicios

- Genera 100 datos aleatorios entre 0 y 6. Realiza un recuento con distinto número de clases y realiza el histograma, también con distinto número de clases.
- Utilizando **RandomVariate** genera 1000 números de acuerdo a una distribución normal tipificada y dibuja el histograma.

Nuevas funciones

BinCounts, Histogram, RandomVariate, NormalDistribution, SmoothHistogram.

17.3. Diagrama de cajas y bigotes

Dado un conjunto de datos, podemos dibujar el diagrama de cajas y bigotes con el comando **BoxWhiskerChart[v]**. Si en vez de un vector, ponemos dos, nos dibuja dos diagramas. Los vectores deben ir en forma de lista.

Ejercicios

Genera 100 números aleatorios y dibuja el diagrama de cajas y bigotes. Genera 500 números aleatorios y dibuja ambos diagramas a la vez.

Nuevas funciones

BoxWhiskerChart.

17.4. Diagramas de dispersión

Dados dos conjuntos de datos del mismo tamaño n , lo podemos entender como un conjunto de n puntos en el plano. El comando para dibujar puntos en el plano es **ListPlot[a]** donde el argumento es una lista formada por los puntos que queremos dibujar. Esta lista en realidad es una matriz de dos columnas.

Ejercicios

- Representar los puntos (2, 4), (3, 1) y (2, 2).
- Generar un conjunto de 100 puntos reales aleatorios en el intervalo $[0, 1]$ y representar el diagrama de dispersión.

Nuevas funciones

ListPlot.

18. Factorización de Matrices

18.1. Descomposición LU

No siempre es posible hallar esta factorización. Permutando adecuadamente las filas, si es posible.

El comando es **LUDecomposition[M]**. La salida está formada por tres objetos. **El primero es una matriz donde se encuentran mezcladas las matrices L y U .** El segundo es un vector de permutaciones. El tercer resultado solamente tiene interés cuando los cálculos se realizan en coma flotante. Debido a estas peculiaridades del comando, podemos pedirle la descomposición a **WolframAlpha**, escribiendo:

LU decomposition of [Matriz]

Ejercicios

Realizar la descomposición LU de la matriz:

$$\begin{pmatrix} 1 & 2 & 1 \\ 4 & -1 & -3 \\ -2 & 1 & 1 \end{pmatrix}$$

Nuevas funciones

LUDecomposition.

18.2. Descomposición QR

La descomposición QR de una matriz A consiste en encontrar dos matrices Q y R , tales que:

- $A = Q \cdot R$
- Q es una matriz ortogonal ($Q \cdot Q^t = \text{Id}$)
- R es triangular superior.

El comando **QRDecomposition[M]** nos proporciona dos matrices Q y R que cumplen:

$$\text{Transpose}[Q] \cdot R = A$$

Para saber si una matriz es ortogonal podemos realizar el producto de la matriz por su transpuesta o también utilizar el comando **OrthogonalMatrixQ[M]**.

Ejercicios

Realizar la descomposición QR de la matriz:

$$\begin{pmatrix} 1 & 2 & 1 \\ 4 & -1 & -3 \\ -2 & 1 & 1 \end{pmatrix}$$

Nuevas funciones

QRDecomposition, OrthogonalMatrixQ.

18.3. Descomposición de Choleski

Dada una matriz simétrica y definida positiva, se puede encontrar una base ortonormal, donde la matriz se reduce a la identidad. A nivel matricial, ello implica la existencia de una matriz L que cumple:

$$L \cdot L^t = A$$

Dicha matriz L se puede elegir que sea triangular inferior. Esto es conocido como descomposición de Cholesky, que se puede obtener con **CholeskyDecomposition[M]**. El resultado de este comando es una matriz L **triangular superior** que cumple:

$$\text{Transpose}[L] \cdot L = A$$

Para saber si una matriz es definida positiva utilizamos **PositiveDefiniteMatrixQ[M]**

Ejercicios

Realizar la descomposición de Choleski de la matriz:

$$\begin{pmatrix} 4 & 12 & -16 \\ 12 & 37 & -43 \\ -16 & -43 & 98 \end{pmatrix}$$

Nuevas funciones

CholeskyDecomposition, **PositiveDefiniteMatrixQ**.

18.4. Descomposición de Schur

Dado un espacio vectorial complejo, y una matriz con coeficientes complejos A , es posible (debido a que los complejos son algebraicamente cerrados) encontrar una matriz P tal que:

$$P^* \cdot A \cdot P$$

sea triangular.

Para obtener la transpuesta conjugada empleamos **ConjugateTranspose[M]**. Para saber si una matriz es unitaria podemos multiplicar la matriz por su compleja conjugada o podemos emplear el comando **UnitaryMatrixQ[M]**.

El comando **SchurDecomposition[M]** nos proporciona una matriz unitaria U y una matriz triangular T que cumplen:

$$U \cdot T \cdot \text{ConjugateTranspose}[U] = A$$

Ejercicios

Realiza la descomposición de Schur de la matriz:

$$\begin{pmatrix} -149 & -50 & -154 \\ 537 & 180 & 546 \\ -27 & -9 & -25 \end{pmatrix}$$

Nuevas funciones

ConjugateTranspose, UnitaryMatrixQ, SchurDecomposition.

19. Diagonalización de matrices

19.1. Los autovectores y autovalores

Dada una matriz A cuadrada, consideramos la matriz $A - \lambda \text{Id}$. Al calcular el determinante se obtiene un polinomio en λ . Las soluciones de dicho polinomio son los **autovalores** de A . Si λ es un autovalor, la matriz $A - \lambda \text{Id}$ es singular y tiene un núcleo no nulo. Los vectores del núcleo son los **autovectores** asociados al autovalor λ . Si conseguimos tantos autovectores linealmente independientes como dimensión tenga la matriz, la matriz es diagonalizable.

Ejercicios

- Encuentra los autovalores y autovectores de:

$$A = \begin{pmatrix} 1 & 0 & 2 \\ 3 & 3 & -3 \\ 1 & 0 & 2 \end{pmatrix}$$

- Comprueba que los autovectores efectivamente lo son.

19.2. Los comandos de diagonalización

Para saber si una matriz es diagonalizable tenemos el comando **DiagonalizableMatrixQ[M]**. Una vez que sabemos que es diagonalizable con **CharacteristicPolynomial[M,x]** tenemos el polinomio característico. Las raíces de este polinomio son los valores propios, que podemos conocer con **Eigenvalues[M]**. Los vectores propios asociados se obtienen con **Eigenvectors[M]**. Si llamamos **P** a la matriz transpuesta que proporciona **EigenVectors** entonces se cumple:

$$\text{Inverse}[P] \cdot M \cdot P = D$$

siendo **D** la matriz diagonal formada por los valores propios. Con **Eigensystem[M]** tenemos los valores propios y los vectores propios.

Ejercicios

Diagonaliza la matriz:

$$A = \begin{pmatrix} 1 & 0 & 2 \\ 3 & 3 & -3 \\ 1 & 0 & 2 \end{pmatrix}$$

Nuevas funciones

DiagonalizableMatrixQ, **CharacteristicPolynomial**,
Eigenvalues, **Eigenvectors**, **Eigensystem**.

19.3. La descomposición en valores singulares

Dada una matriz rectangular A , entonces $A \cdot A^t$ es una matriz cuadrada, simétrica y definida positiva y por lo tanto tiene todos los autovalores positivos. La raíz cuadrada de dichos autovalores son **los valores singulares de A** . Se pueden obtener con `SingularValueList[A]`. El comando `SingularValueDecomposition[A]` devuelve tres matrices (U, V, W) , donde v es diagonal y tiene los valores singulares en la diagonal y además:

$$A = U \cdot V \cdot W^t$$

La matrix A debe estar escrita con números decimales, pues en caso contrario devuelve todo en función de números algebraicos.

Ejercicios

Encuentra los valores singulares y la descomposición en valores singulares de la matriz:

$$A = \begin{pmatrix} 3. & 5 & 6 & 5 \\ 2 & 7 & 4 & 5 \\ 2 & 5 & 9 & 1 \end{pmatrix}$$

Nuevas funciones

SingularValueList, SingularValueDecomposition.

19.4. La forma de Jordan

Si la matriz no es diagonalizable, lo más que podemos hacer con la matriz es reducirla a la **forma normal de Jordan** (también se llama **forma canónica** de la matriz). **Si la matriz es diagonalizable la forma normal de Jordan es la matriz diagonal.**

El comando **JordanDecomposition[A]** realiza la descomposición. El resultado de este comando son dos matrices. La primera es la matriz P de cambio de base y el segundo J es la forma de Jordan de la matriz. Todas las matrices que hemos calculado están ligadas por la expresión.

$$A = P \cdot J \cdot P^{-1}$$

Ejercicios

Realiza la descomposición de Jordan de las matrices:

$$A = \begin{pmatrix} 1 & 0 & 2 \\ 3 & 3 & -3 \\ 1 & 0 & 2 \end{pmatrix} \quad B = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 4 & -1 \\ 0 & 1 & 2 \end{pmatrix}$$

Nuevas funciones

JordanDecomposition.

20. Cálculo vectorial

20.1. Dibujo de campos vectoriales

Una función $V : \mathbb{R}^n \rightarrow \mathbb{R}^n$ se puede entender como un conjunto de n funciones de n variables. Sin embargo también podemos pensarlo de otro modo: dado un punto $a \in \mathbb{R}^n$, $f(a)$ es un elemento de \mathbb{R}^n y por lo tanto es un vector. Si dibujamos dicho vector $f(a)$, «naciendo» en el punto a , tenemos un campo vectorial: en cada punto de \mathbb{R}^n tenemos un vector dibujado. Para dibujar estos campos vectoriales se emplea el comando

VectorPlot[campo, {x,xmin,xmax},{y,ymin,ymax}]

y el análogo en tres dimensiones **VectorPlot3D**

Ejercicios

Dibuja el campo vectorial bidimensional $V = yi - xj$ y el campo tridimensional $V = (2xy, 4x, 5zy)$.

Nuevas funciones

VectorPlot, VectorPlot3D.

20.2. Gradiente, laplaciano y hessiano

Dada una función $f : \mathbb{R}^n \rightarrow \mathbb{R}$, diversas combinaciones de derivadas parciales tienen sentido geométrico. El gradiente de una de estas funciones se calcula con `Grad[f,{x1,...,xn}]` y el laplaciano con `Laplacian[f,{x1,...,xn}]`. Si no indicamos nada en contra, **Mathematica supone que estamos utilizando coordenadas cartesianas**.

Como el gradiente y la divergencia no son más que combinaciones de derivadas parciales, se pueden obtener con el operador **D**. El gradiente se calcula con (ojo a la doble llave):

`D[f[x,y,z],{{x,y,z}}]`

El **hessiano** es la matriz de las derivadas segundas y no tiene comando específico en Mathematica. Lo podemos calcular con:

`D[f[x,y,z],{{x,y,z},2}]`

Ejercicios

Calcula el gradiente, el laplaciano y el hessiano de una función general y de:

$$f = x^2 + y^2 + z^2$$

$$g = xyz^6 + 3(x+y)z^2$$

Nuevas funciones

Grad, Laplacian.

20.3. Divergencia, rotacional y jacobiano

Los campos vectoriales en \mathbb{R}^n , una vez fijadas unas coordenadas, se pueden entender como funciones vectoriales de varias variables. Para definir una función vectorial creamos una lista (entre llaves) con cada una de las funciones coordenadas. La divergencia se calcula con **Div** $\{\{f_1, \dots, f_n\}, \{x_1, \dots, x_n\}\}$ y el rotacional con **Curl** $\{\{f_1, \dots, f_n\}, \{x_1, \dots, x_n\}\}$

El **jacobiano** no tiene comando asignado en Mathematica, pero lo calculamos con:

$$\mathbf{D}[\mathbf{V}[\mathbf{x}, \mathbf{y}, \mathbf{z}], \{\{\mathbf{x}, \mathbf{y}, \mathbf{z}\}\}]$$

Ejercicios

Calcula la divergencia, el rotacional y el jacobiano del campo:

$$V = \{3(x+y)z, 5xyz^3, 8x^2 + 3z - 7y\}$$

Nuevas funciones

Div, Curl.

20.4. Identidades vectoriales

En el cálculo vectorial existen muchas identidades que nos permiten realizar los cálculos de manera más sencilla. Con Mathematica podemos comprobar con ejemplos la veracidad de dichas fórmulas. Además de eso, y gracias al cálculo simbólico, se pueden «demostrar» estas identidades.

Ejercicios

Comprueba las siguientes identidades vectoriales:

- $\text{rot}(\text{grad}(f)) = 0$
- $\text{div}(\text{grad}(f)) = \text{lap}(f)$
- $\text{div}(\text{rot}(V)) = 0$

20.5. Otros sistemas de coordenadas

Además de en coordenadas cartesianas, Mathematica realiza cálculos en **distintos sistemas de coordenadas**. Las más típicas son las esféricas y las cilíndricas (o las polares en dimensión 2). Para decirle a Mathematica que realice los cálculos en estas coordenadas debemos especificar **“Spherical”** y **“Cylindrical”**, **“Polar”** como un argumento extra (es necesario escribir las comillas). Las variables pueden tener cualquier nombre, pero si queremos seguir la notación matemáticas podemos utilizar la tecla “Esc”.

Ejercicios

Calcula en coordenadas esféricas el gradiente de:

$$r \cos(\theta) \sin(\phi)$$

Nuevas funciones

Spherical, Cylindrical, Polar.

21. Transformada de Laplace

21.1. Definición

La transformada de Laplace convierte una función $f(t)$ en la variable t , en una función $F(s)$ en la variable s :

$$F(s) = \mathcal{L}(f(t))(s) = \int_0^{\infty} f(t)e^{-st} dt$$

Al estar definida mediante una integral impropia se imponen ciertas condiciones para asegurar la convergencia:

- **f debe estar definida para todos los valores positivos.**
Los valores de f en la parte negativa del eje no se consideran en la integral.
- **Normalmente debe ocurrir que $s > a$ para que la integral sea convergente,** puesto que para número negativos $\exp(-st)$ es una función con un gran crecimiento y se tienen serias posibilidades de no convergencia.

Ejercicios

Calcula la transformada de Laplace del seno utilizando la definición:

$$\mathcal{L}(f(t))(s) = \int_0^{\infty} f(t)e^{-st} dt$$

21.2. El comando **LaplaceTransform**.

Se utiliza **LaplaceTransform[f,t,s]** para efectuar la transformada, donde f es la función en la primera variable, t es la variable de la función sin transformar y s es el variable de la función transformada.

En el manejo habitual de la transformada de Laplace se parte de una serie de transformadas de funciones elementales que suelen venir tabuladas.

Ejercicios

- Calcula las siguientes transformadas de Laplace:

$$a) \mathcal{L}(\sin(\omega t)) \quad b) \mathcal{L}(\cos(\omega t)) \quad c) \mathcal{L}(e^{\omega t})$$

- Calcula la transformada de la delta de Dirac.

Nuevas funciones

LaplaceTransform, DiracDelta.

21.3. La transformada como aplicación lineal

La transformada de Laplace es una aplicación lineal. Para conocer la transformada de un polinomio basta con conocer las transformadas de las potencias.

Ejercicios

Calcula transformadas de constantes, potencias y de polinomios.

21.4. Algunas propiedades

Las propiedades de la transformada de Laplace son la que hacen tan útil a este mecanismo. Como muestra enumeramos algunas:

- $\mathcal{L}(f + g) = F(s) + G(s)$
- $\mathcal{L}(f') = s \cdot F(s) - f(0)$
- $\mathcal{L}\left(\int_0^t f(u)du\right) = \frac{1}{s}F(s)$
- $\mathcal{L}(e^{at} \cdot f) = F(s - a)$

Ejercicios

- Comprueba las propiedades anteriores.
- Calcula, paso a paso, la transformada de Laplace de:

$$t^3 e^{-5t} \cos(2t)$$

21.5. La transformada inversa

Bajo ciertas condiciones, la transformada de Laplace se puede invertir y dada una función $F(s)$ se puede encontrar la función $f(t)$. Para ello se utiliza **InverseLaplaceTransform[F,s,t]**

Ejercicios

Calcula la siguiente transformada inversa de Laplace:

$$\mathcal{L}^{-1}\left(\frac{1}{(s+1)^2(s+2)}\right)$$

Nuevas funciones

InverseLaplaceTransform.

22. Ecuaciones diferenciales

22.1. Solución de una ecuación diferencial

Una ecuación diferencial es una igualdad donde aparece, al menos, una derivada de una función. En general la función desconocida se denota por la letra y . Sus derivadas, o bien por y', y'', \dots o bien por $\frac{dy}{dx}, \frac{d^2y}{dx^2}, \dots$, donde la x es la variable de la que depende la función. **Una solución de una ecuación diferencial es cualquier función que al sustituirla en la ecuación nos proporcione una identidad.** En general las ecuaciones diferenciales tienen infinitas soluciones.

Ejercicios

Comprobar que las siguientes funciones son solución de las ecuaciones diferenciales escritas a su derecha.

$$\exp(x) \hookrightarrow y' = y$$

$$7\exp(x) \hookrightarrow y' = y$$

$$\cos(x) \hookrightarrow y'' = -y$$

$$\sin(x) \hookrightarrow y'' = -y$$

$$a \sin(x) + b \cos(x) \hookrightarrow y'' = -y$$

22.2. Ecuaciones de primer orden

El comando para obtener la solución general es:

DSolve[ecuación, y,x]

- Es importante escribir la dependencia de la variable en todas las apariciones de la función y sus derivadas.
- Para extraer la solución se pueden utilizar los corchetes dobles.
- El resultado es una «función pura».
- Las infinitas soluciones dependen de una constante, que se denota **C[1]**.

Ejercicios

Resuelve y comprueba la solución:

$$a) \frac{dy}{dx} = y \qquad b) y' = 2x + 3y$$

Nuevas funciones

DSolve

22.3. Ecuaciones de mayor orden

En las ecuaciones de orden mayor que uno, aparecen tantas constantes como orden tenga la ecuación. Las derivadas sucesivas se van indicando con más apóstrofes.

La mayor parte de las ecuaciones diferenciales no las puede resolver Mathematica de modo exacto y en muchas de ellas aparecen las llamadas **funciones especiales**.

Ejercicios

Resuelve y comprueba alguna solución:

$$a) y'' = y$$

$$b) x^2 \cdot y'' + x \cdot y' + (x^2 - 9) \cdot y = 0$$

22.4. Ecuaciones con valor inicial

Dada una ecuación diferencial de primer grado, si fijamos el valor en un punto de la solución, estamos eligiendo una de las infinitas soluciones de la ecuación general. **El teorema de unicidad de ecuaciones diferenciales, nos dice que, bajo supuestos muy generales, dicha solución es única.** Para dar la condición inicial la escribimos junto con la ecuación diferencial en forma de lista.

`DSolve[{ecuación, condición}, y,x]`

Naturalmente si la ecuación es de grado mayor debemos dar tantas condiciones como orden tenga la ecuación.

Ejercicios

- Resuelve y dibuja $y' = y$ con $y(0) = 3$.
- Resuelve y dibuja $y'' = y$ con las condiciones $y(3) = 4$ e $y'(0) = 2$.

22.5. El campo vectorial asociado a la ecuación

Dada una ecuación diferencial $y' = f(x, y)$ el campo vectorial $\{1, f(x, y)\}$ es el campo asociado la ecuación diferencial. Para dibujar el campo utilizamos:

VectorPlot[campo,{x,xmin,xmax},{y,ymin,ymax}]

Las curvas que son soluciones de la ecuación se adaptan al campo en el siguiente sentido: el vector velocidad de la curva en el punto es también el vector del campo vectorial.

Ejercicios

Dada la ecuación diferencial $y' = y$, dibuja el campo vectorial y una solución de la ecuación diferencial, todo en la misma gráfica.

Nuevas funciones

VectorPlot.

22.6. Sistemas de ecuaciones diferenciales

En un sistema, existen varias funciones desconocidas, todas dependiendo de la misma variable. También existen varias ecuaciones y se pueden establecer condiciones iniciales para varias funciones. La solución general dependerá de varios parámetros y en cuanto vayamos añadiendo condiciones iniciales, se irán reduciendo las constantes de integración.

Ejercicios

- Resuelve el siguiente sistema de ecuaciones diferenciales:

$$\begin{cases} \frac{dx}{dt} = t \\ \frac{dy}{dt} = t^2 \end{cases}$$

- Añadir la condiciones iniciales $x(0) = 1$ e $y(2) = 3$.

22.7. Resolución numérica

Normalmente una ecuación diferencial no admite una solución explícita en términos de funciones, ya sean elementales o especiales. En este caso debemos recurrir a la integración numérica.. El comando para hacer esta aproximación es:

NDSolve[ecuación, y,{x,xmin,xmax}]

Es imprescindible dar las condiciones iniciales, de modo que la solución sea única. La solución será una función definida únicamente en el dominio $(xmin, xmax)$. Para extraer la solución volvemos a emplear el doble corchete. Con la solución no podemos hacer muchas cosas, puesto que no tiene una fórmula explícita. Podemos, eso si, **calcular valores de la función en puntos del dominio y graficar la función, también en puntos del dominio.**

Ejercicios

Resuelve de modo aproximado la ecuación $y' + y = 0$ con la condición inicial $y(0) = 1$, en el intervalo $(0, 2\pi)$. Grafica la solución.

Nuevas funciones

NDSolve.

23. Expresiones trigonométricas

23.1. Simplificaciones trigonométricas

El comando **Simplify** puede simplificar algunas expresiones trigonométricas. Debemos decir que este problema no puede ser bien resuelto por Mathematica pues no existe un concepto matemático de simplificación en el conjunto de expresiones algebraicas. De entre todas las expresiones trigonométricas equivalentes, no podemos decir cual es la mas simple, al contrario de lo que sucede por ejemplo con las fracciones. En este caso la más sencilla (o irreducible) es la que tiene números mas pequeños.

Ejercicios

Simplifica las siguientes expresiones:

- $\sin^2(x) + \cos^2(x)$

- $\frac{\sin(x)}{\cos(x)}$

- $\frac{\sin^2(x)}{1 - \cos^2(x)}$

- $\frac{\sin^2(x)}{1 - \sin^2(x)}$