

Capítulo 3

Cálculo simbólico: Funciones analíticas

3.1. La toolbox de cálculo simbólico

Hasta ahora hemos manejado datos numéricos. En esta sección estudiaremos algunas utilidades de MATLAB para realizar cálculo simbólico. Las variables utilizadas hasta ahora eran simplemente contenedores de datos. Aquí aprenderemos cómo se define una variable simbólica x y cómo a partir de ella se puede construir una función que puede someterse a diversas operaciones como la derivación o integración. La variable x no contiene ningún dato numérico sino que tiene el carácter abstracto que usualmente se le da en matemáticas.

Podemos darnos una idea de las capacidades de MATLAB para cálculo simbólico mediante la ayuda. Si queremos obtener información sobre una instrucción concreta, por ejemplo sobre `diff`, bien la buscamos usando el menú de ayuda `help windows` o bien escribimos `help sym/diff`.

Otra utilidad interesante es `funtool`. Escribiendo esta instrucción se accede a un panel que permite realizar diversas operaciones. Dispone de una lista de funciones a las que podemos acceder pulsando en `Cycle`. Podemos añadir nuevas funciones mediante `Insert` y eliminarlas de la lista usando `Remove`.

3.2. Creando objetos simbólicos

Función	Salida
<code>sym</code>	Convierte una variable a simbólico. Si la variable no existe, la crea.
<code>syms</code>	Crea variables simbólicas

Pueden crearse objetos simbólicos como sigue:

- » `syms x real %crea la variable simbólica real x.`
- » `f=3*x^2+2*x+1 %crea la variable simbólica f.`

En la instrucción anterior la variable x tiene el sentido de variable independiente habitual en matemáticas. Como veremos posteriormente, f se puede integrar o derivar respecto a x .

```
» syms a b; %crea las variables simbólicas a y b
» c=sym('1/3') %almacena en c la constante simbólica 1/3
» p=sym('pi') %almacena en p la constante simbólica pi
```

Las instrucciones anteriores pueden escribirse como `c=sym(1/3)` o `p=sym(pi)`

El comando `sym` también puede aplicarse a una matriz:

```
» A=[1 2/3;pi sqrt(2)]
» A=sym(A)
```

3.3. Operaciones

Con las variables simbólicas pueden realizarse operaciones elementales del mismo modo que con las variables numéricas utilizando los operadores habituales. Además, hay otros comandos para realizar operaciones simbólicamente. Una lista de éstos puede obtenerse tecleando `help symbolic`. Conviene advertir que hay varios comandos que actúan tanto sobre variables simbólicas como sobre variables numéricas si bien, como es lógico, operan de modo diferente.

Función	Salida
<code>collect</code>	agrupa términos.
<code>factor</code>	factoriza la expresión; si se aplica a un número entero, lo descompone en factores.
<code>expand</code>	multiplica factores.
<code>pretty</code>	presentación más legible.

Ejemplo 15 Agrúpese la expresión $xy + \sin(x)y - y^3x + x^2y^2 + y^3\sin(x)$ en potencias de “ x ” y de “ y ”.

```
» syms x y
» p=x*y+sin(x)*y-y^3*x+x^2*y^2+y^3*sin(x);
» pretty(p)
» q=collect(p)
» pretty(q)
» r=collect(p,y)
» pretty(r)
```

Ejemplo 16 Halle los coeficientes de un polinomio cuyas raíces sean i , $1 - i$ y $3i$.

```

» syms z
» p=(z-i)*(z-1+i)*(z-3i)
» q=expand(p)
» collect(q)
» pretty(ans)

```

Si se quiere volver a la expresión primitiva basta escribir

```

» factor(q) %factoriza el polinomio.

```

3.4. Simplificando resultados

Función	Salida
simplify	intenta simplificar aplicando identidades matemáticas.
simple	intenta simplificar al menor número de caracteres.

Ejemplo 17 *Simplifique la expresión $\sin^2(x) + (1 - \cos(x))^2$.*

```

» p=sin(x)^2+(1-cos(x))^2
» simplify(p)

```

Más poderoso que `simplify` es `simple` y en ocasiones debe aplicarse más de una vez:

Ejemplo 18 *Simplifíquese la expresión $\sqrt[3]{27 + 27x^{-1} + 9x^{-2} + x^{-3}}$.*

```

» p=(27+27/x+9/x^2+1/x^3)^(1/3)
» q=simple(p)
» q=simple(q)
» q=simple(q)

```

3.5. Sustituciones en una expresión simbólica y conversión a numérico

Función	Salida
subs	substituye una expresión
compose(f,g,x,y,z)	compone dos funciones simbólicas, f y g, donde la variable independiente de la composición será z y las independientes de f y de g serán respectivamente x e y. Las variables x, y y z son opcionales
double	obtiene el valor numérico
digits	especifica el número de dígitos
vpa	evalúa una expresión con la precisión deseada

Ejemplo 19 Construya $f = ax^2 + bx + c$ y sustitúyase x por s^2 . Haciendo $a = 1$, $b = 2$ y $c = \sqrt{3}$, obténgase el valor de f para $s = 1$ y $s = 4$.

```
» syms x a b c
» f=a*x^2+b*x+c
» syms s
» g=subs(f,x,s^2) %en f sustituye x por s^2
» h=subs(g,{a,b,c},{1,2,sqrt(3)}) %substitución múltiple.
» k=subs(h,s,[1;4]) %sustituye s por una matriz.
```

Ejemplo 20 Obtenga el valor de $f(x,y) = \sqrt[3]{2x+5y+3}$ en los puntos $(0,0)$, $(0,1)$, $(0,2)$, y $(0,3)$.

```
» syms x y
» f=(2*x+5*y+3)^(1/3)
» v=0:3;
» val=subs(f,{x,y},{0*v,v}) %sustituye (x,y) por (0,v(i))
```

En la parte básica, MatLab utiliza la aritmética de punto flotante y trabaja con 16 dígitos. Por este motivo, si se manejan números de más decimales, lo que sucede con números irracionales, en cada operación se produce un error llamado de redondeo. En cálculo simbólico no se produce este tipo de error pues MatLab no realiza cálculos numéricos, trabaja simbólicamente. Sí puede producirse un error de redondeo cuando se usa la instrucción `double` para convertir un resultado simbólico a numérico.

3.6. Diferenciación e integración

Si f es una expresión simbólica, la instrucción `diff` se usa en una de las cuatro formas siguientes:

<code>diff(f,u,n)</code>	Halla la derivada de orden n (n número entero) respecto a u
<code>diff(f)</code>	Halla la derivada respecto a la variable por defecto
<code>diff(f,u)</code>	Halla la derivada respecto a u
<code>diff(f,n)</code>	Halla la derivada de orden n respecto a la variable por defecto

En las instrucciones segunda y cuarta, si no aparece la variable x , las derivadas se realizan respecto a la variable por defecto, que es “ x ”, o si esta no aparece, la letra del alfabeto más próxima a “ x ” (si hay dos a la misma distancia se elige la de la derecha).

Ejemplo 21 Construya la función $f(x,y) = ax^2 + bxy + xy^3$ y halle:

- la derivada de f respecto a x .

- *la derivada segunda de f respecto a x.*
- *la derivada de f respecto a y.*
- *la derivada segunda de f respecto a y.*

Las instrucciones que tenemos que escribir son:

```
» syms x y a b
» f=a*x^2+b*x*y+x*y^3
» dfx=diff(f)
» d2fx=diff(f,2)
» dfy=diff(f,y)
» d2fy=diff(f,y,2)
```

Conviene señalar que `diff` puede actuar sobre una matriz. También hay un operador `diff` que actúa sobre variables numéricas (obviamente no calcula la derivada).

La instrucción `int` actúa como sigue:

<code>int(f,s)</code>	Calcula una primitiva de f respecto a s.
<code>int(f)</code>	Calcula una primitiva de f respecto a la variable por defecto.
<code>int(f,s,a,b)</code>	Calcula la integral definida respecto a s entre a y b.
<code>int(f,a,b)</code>	Calcula la integral definida respecto a la variable por defecto entre a y b

En las instrucciones segunda y cuarta, si en f no aparece la variable x, calcula las integrales respecto a la variable indicada por **findsym(f,1)**.

Los límites a y b pueden ser datos numéricos o expresiones simbólicas.

Ejemplo 22 *Halle una primitiva de las funciones*

$$f(x) = \frac{1}{\sqrt{x^2+1}} \quad y \quad g(x) = \frac{\ln(x)}{x}$$

```
» syms x
» f=1/sqrt(x^2+1)
» F=int(f)
» G=int(log(x)/x)
» diff(F)
» G=int(log(x)/x)
» G=simplify(ans)
» diff(G)
```

3.7. Solución de ecuaciones

En esta sección vamos a resolver ecuaciones simbólicas mediante la función `solve`. Mediante esta instrucción MATLAB obtiene soluciones de ecuaciones. MATLAB busca soluciones en el campo de los números complejos y cuando no puede obtener soluciones simbólicas intenta obtener soluciones numéricas.

Función	Salida
<code>solve</code>	encuentra soluciones de una ecuación

Ejemplo 23 Calcule todas las raíces del polinomio $z^3 + z^2 - 4z + 6$

La instrucción

```
» syms z
» p=z^3+z^2-4*z+6
» sol=solve(p)
```

nos da las soluciones de la ecuación. La solución es un vector de tres componentes que hemos guardado con el nombre `sol`.

También podíamos haber escrito

```
» sol=solve('z^3+z^2-4*z+6')
```

y también se pueden omitir los apóstrofes, pero en este caso es preciso que `z` esté declarada como variable simbólica

```
» syms z
» sol=solve(z^3+z^2-4*z+6)
```

Ejemplo 24 Halle la solución general de la ecuación $az^4 + bz^2 + c = 0$.

```
» sol=solve('a*z^4+b*z^2+c')
```

En el comando `solve` podemos especificar cual es la variable que deseamos despejar, en el caso de que halla varias. Así, en la ecuación del ejemplo anterior podemos despejar b escribiendo

```
» sol=solve('a*z^4+b*z^2+c','b')
```

También podemos resolver un sistema de ecuaciones.

Ejemplo 25 Halle las soluciones del sistema $-90z + 12w + 45z^2 + 6z^2w - 12zw + z^2w^2 - 2zw^2 + 2w^2 = -90$, $w^2 - 2w = -5$.

MATLAB almacena la solución del sistema en una estructura de datos:

```
» sol=solve('90-90*z+12*w+45*z^2+6*z^2*w-12*z*w+z^2*w^2-2*z*w^2+2*w^2','w^2-2*w+5')
```

Para obtener los valores de la solución escribimos

```
» sol.w,sol.z
```

En el caso de que en el sistema aparezcan más variables que ecuaciones también podemos elegir qué variables deseamos despejar.

Ejemplo 26 En el sistema $ax + by = 0$, $bx - ay + 1 = 0$, despeje las variables a y x en función de las variables y y b .

```
» sol=solve('a*x+b*y','b*x-a*y+1','a','x')
» solucion=[sol.a sol.x]
```

Observamos también que en este caso hay más de una solución.

3.8. Ejercicios

Ejercicio 3.1 Escriba una función $[w]=\text{logalfa}(z, \text{alfa})$ que calcule la imagen por la rama α del logaritmo, $\log_\alpha(z)$, de un complejo $z \in \mathbb{C}$, dando un mensaje de error cuando dicho valor no esté definido. (Utilice la función argalfa de la práctica 1). Compruebe su funcionamiento creando un fichero `ej3_1.m` en el que escribirá las instrucciones de cálculo de $\log_0(-i)$, $\log_\pi(-i)$, $\log_{20\pi}(-i)$ y $\log_{5\pi}(-i)$. En el mismo fichero, dibuje los valores obtenidos con puntos rojos. Calcule con la función el $\log_{\pi/2}(-i)$ ¿existe? ¿porqué?

Ejercicio 3.2 Escriba una función $[c]=\text{cauchyR}(f, x, y)$ que nos diga si $f(x+iy)$ verifica o no las condiciones de Cauchy-Riemann en $\mathbb{C}$. La variable c tomará el valor 1 si se cumplen y 0 en caso contrario. Compruébese su funcionamiento aplicándola a:

1. $f(x+iy) = e^x \cos(y) + e^x \sin(y)i$ en $\mathbb{C}$.
2. $f(x+iy) = 2xy - (x^2 - y^2)i$ en $\mathbb{C}$.
3. $f(x+iy) = x^2 + y^2$ en $\mathbb{C}$.

Guárdense las instrucciones para comprobar el funcionamiento en un fichero `ej3_2.m`

Ejercicio 3.3 Escriba una función `[v]=armoconj(u,x,y)` que, calcule una armónica conjugada de $u(x,y)$, comprobando previamente si es o no armónica. (En caso de no serlo, debe devolver un mensaje de error y parar la ejecución). Compruébese su funcionamiento aplicándola a:

1. $u(x,y) = y^3 - 3x^2y$.
2. $u(x,y) = e^{-y} \sin(x) - x^2 + y^2$.
3. $u(x,y) = \sqrt{x^2 + y^2}$

Utilice la función `[c]=cauchyR(f,x,y)` para comprobar que, en efecto, las funciones obtenidas son armónica conjugada de la dada. Guárdese las instrucciones en un fichero `ej3_3.m`

Ejercicio 3.4 Modifique la función anterior para que tenga dos argumentos de entrada más, `[v]=armo_conj2(u,x,y,z,c)`, que devuelva la armónica conjugada de $u(x,y)$ que en z tome el valor c . Compruébese el funcionamiento con la función $u(x,y) = x^2 - y^2 - x^3 + 3xy^2$ calculando una armónica conjugada suya que verifique que en el punto $-i$ tome el valor $-1 - i$. Idéntico para $f(x,y) = e^{-y} \cos(x) - e^y \cos(-x)$ de forma que la armónica conjugada tome el valor 2 en i . Guárdese las instrucciones para comprobar el funcionamiento en un fichero `ej3_4.m`