HERRAMIENTAS INFORMÁTICAS PARA MATEMÁTICAS
(Apuntes elaborados por el equipo docente para el curso 2020-21)

Tema 3

CÁLCULOS MATEMÁTICOS BÁSICOS

En este tema se propone algunos elementos necesarios para la solución de muchos problemas matemáticos básicos. Algunos de estos cálculos, como los límites de funciones, o la diferenciación y la integración, son necesarios en pasos para la resolución de problemas matemáticos más complejos. La factorización o las simplificaciones trigonométricas pueden ser herramientas de gran utilidad para el análisis de soluciones.
El tema estará dividido en los siguientes apartados:

- Límites, diferenciación e integración.
- Factorización, simplificaciones y desarrollos en serie.
- Simplificaciones trigonométricas.

3.1 LÍMITES, DIFERENCIACIÓN E INTEGRACIÓN

3.1.1 Límites

La diferencia esencial entre el cálculo del límite de funciones empleando Maxima y Scilab es el empleo de cálculo simbólico para realizar las operaciones. Se puede observar que Maxima tiene una función dedicada al cálculo de límites, dando como resultado el valor exacto y Scilab nos permitirá realizar cálculos numéricos, dando un valor aproximado, desarrollando los conceptos de la definición de límite, δ-ε.

Sin llegar a realizar una definición detallada y formal de límite de una función. El límite de una función *f(x)* cuando *x* tiende a un valor *c* es *L*,

$$\lim_{x \to c} f(x) = L$$

se puede aproximar al valor de *f(x)*, *L*, tanto como se desee, ya que siempre se podrá encontrar un valor de *x* cercano a c que lo permita.

$$\forall \varepsilon > 0 \; \exists \; \delta > 0 : \forall x (0 < \left| x - p \right| < \delta \Rightarrow \left| f(x) - L \right| < \varepsilon$$

Para ilustrar el problema de los límites, se desarrollan algunos ejemplos típicos.

Ejemplo 3.1 *Calcular el siguiente límite,*

$$\lim_{x \to \frac{\pi}{2}} \frac{2\sin(x)}{3\cos(x)}$$

Para solucionar este problema con Scilab, se puede crear la siguiente rutina,

```
-->format(20)

-->x=%pi/2;

-->limit1=2*sin(x)/(3*cos(x))
 limit1  =

    10887492902130246.
```

Código 3.1 Cálculo de límite en Scilab.

En este caso se estaría cometiendo un error, ya que el problema presenta distintos valores si se realiza el límite de la función por la derecha o por la izquierda de $\frac{\pi}{2}$. Es recomendable, sobre todo si no se conoce la función, que en Scilab se calcule el límite a la derecha y a la izquierda. Una rutina más adecuada se muestra a continuación,

```
-->format(20)

-->x=%pi/2;

-->delta=1D-15;

-->limit1_m=2*sin(x-delta)/(3*cos(x-delta))
 limit1_m  =

    569092674652823.125

-->limit1_M=2*sin(x+delta)/(3*cos(x+delta))
 limit1_M  =

  - 635531541328984.375
```

Código 3.2 Cálculo de límite en Scilab.

El resultado obtenido muestra un valor que se puede considerar infinito numéricamente. Obteniendo el resultado infinito positivo por la izquierda e infinito negativo por la derecha. Según la definición de límite se puede obtener valores más cercanos al límite de la función si se aproxima al valor de la variable, haciendo δ más pequeño.

Se realiza esta misma operación con Maxima.

```
(%i11) limit(2*sin(x)/(3*cos(x)),x,%pi/2,minus);
(%o11) ∞

(%i12) limit(2*sin(x)/(3*cos(x)),x,%pi/2,plus);
(%o12) -∞

(%i13) limit(2*sin(x)/(3*cos(x)),x,%pi/2);
(%o13) infinity
```

Código 3.3 Cálculo de límite con Maxima.

Se puede ver que para el cálculo de límites con Maxima, se emplea la función `limit()`.
La sintaxis de esta función es la siguiente:

```
limit(función, variable, límite, sentido de aproximación)
```

En el ejemplo se observa que el resultado es distinto si se añade el parámetro `minus` o `plus`, realizando la aproximación al límite por la izquierda o por la derecha respectivamente, ya que la función tangente es discontinua. Si no se emplea este argumento, Maxima muestra como salida el valor de infinito complejo, `infinity`.

También se puede considerar el límite de una función tendente a infinito. Se ilustra este tipo de límites con un ejemplo.

Ejemplo 3.2 *Calcular el siguiente límite,*

$$\lim_{x\to\infty}\frac{x^2+1}{2x^2+x+5}$$

El planteamiento numérico del problema con Scilab sería el siguiente,

```
-->format(20)

-->min_l=log(1D5)
 min_l  =

    11.5129254649702286

-->max_l=log(1D20)
 max_l  =

    46.0517018598809145

-->x_l=linspace(min_l,max_l,20);

-->fx_2=(exp(x_l).^2+1)./(2*exp(x_l).^2+exp(x_l)+5);

-->plot(x_l,fx_2)
```

Código 3.4 Calculo de límite y representación gráfica en Scilab.

La utilidad de visualizar numéricamente el límite de la función es útil. La gráfica resultante se muestra a continuación. Se puede observar que el límite de la función cuando se aumenta el valor de *x*, tiende al valor 0.5.

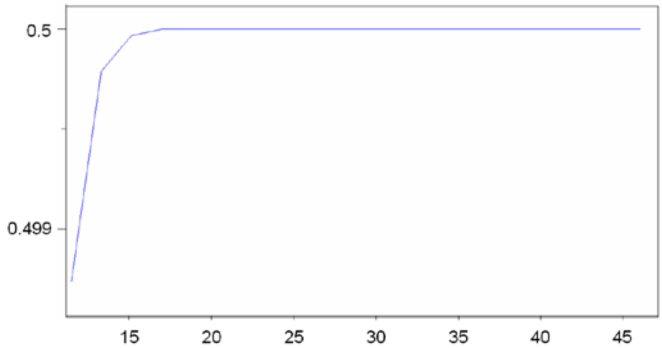


Figura 3.1 Valor del límite de la función frente al valor límite de la variable.

Se procede a calcular el límite de esta función empleando Maxima.

```
(%i1) limit((x^2+1)/(2*x^2+x+5),x,inf);
(%o1) 1/2
```

Código 3.5 Cálculo de límite infinito en Maxima.

El resultado obtenido del cálculo del límite es exacto, el valor de la variable se declara como `inf`, infinito positivo, pero también se puede declarar infinito negativo mediante `minf`.

```
(%i5) limit(x^3,x,minf);
(%o5) -∞
(%i6) limit(x^3,x,inf);
(%o6) ∞
```

Código 3.6 Cálculo de límites infinitos en Maxima.

Maxima ofrece también una herramienta útil, como la detección de límites acotados indeterminados.
Se muestra a continuación un ejemplo de este tipo de límites.

Ejemplo 3.3 *Calcular del límite,*

$$\lim_{x\to 0} \sin\left(\frac{1}{x}\right)$$

La evaluación de esta interesante función produce en Scilab los siguientes resultados,

```
-->x=0;

-->delta=1D-15;

-->sin(1/(x-delta))
 ans  =

  - 0.91555855566346722

-->sin(1/(x+delta))
 ans  =

    0.91555855566346722

-->delta=1D-16;

-->sin(1/(x-delta))
 ans  =

  - 0.77968800660697879

-->sin(1/(x+delta))
 ans  =

    0.77968800660697879
```

```
-->delta=1D-17;

-->sin(1/(x-delta))
 ans  =

    0.46453010483537271

-->sin(1/(x+delta))
 ans  =

  - 0.46453010483537271

-->delta=1D-18;

-->sin(1/(x-delta))
 ans  =

  - 0.60267362482109843

-->sin(1/(x+delta))
 ans  =

    0.60267362482109843
```

Código 3.7 Aproximación numérica del límite en Scilab.

A la vista de los resultados se podría decir que la función es simétrica respecto al valor 0, pero aparentemente no muestra convergencia a un valor determinado.

Se genera una representación gráfica para comprender mejor la función,

```
-->lim_inf=log(1D-20)
 lim_inf  =

  - 46.0517018598809145

-->lim_sup=log(1D-2)
 lim_sup  =

  - 4.60517018598809091

-->val=lim_inf:0.1:lim_sup;

-->for i = 1 : 415
-->fx(i)=sin(1/exp(val(i)));
-->end

-->plot(val,fx)
```

Código 3.8 Representación gráfica en Scilab de una función con límite indeterminado.

El resultado es el siguiente,

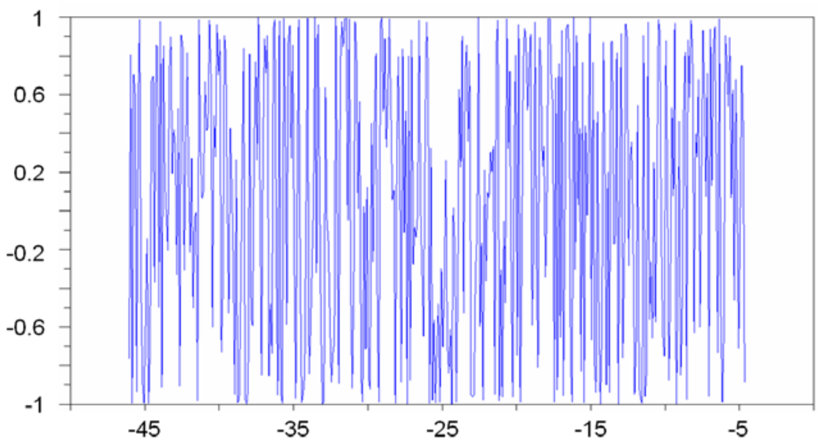


Figura 3.2 Función con límite indeterminado en Scilab.

En la gráfica se puede observar que la función tiene un límite indeterminado, estando acotada entre -1 y 1. Los valores negativos de la función tienen un comportamiento simétrico.

En Maxima, el resultado de calcular el límite de la función es `ind`, esta salida informa al usuario que el límite es indeterminado y acotado, pero no informa cual es el rango en el que está acotado.

```
(%i14) limit (sin(1/x), x, 0);
(%o14) ind
```

Código 3.9 Función con límite indeterminado en Maxima.

3.1.2 Diferenciación

En este apartado se muestra como se realiza el cálculo de las derivadas. Se emplean dos enfoques distintos, obtención de la derivada de manera simbólica y obtención de su aproximación numérica.

Se muestra en primer lugar la potencia de Maxima para el cálculo de la derivada de una función con una sola variable independiente.

Se propone como ejemplo sencillo la función,

$$f(x) = x^2 \sin(x)$$

Se calcula $\dfrac{df(x)}{dx}$,

```
(%i7) diff (x^2*sin(x),x);
(%o7) 2 x sin(x)+x^2 cos(x)
```

Código 3.10 Diferenciación en Maxima.

Se puede observar en el ejemplo como se emplea la función `diff()`, donde en este caso, la sintaxis tiene la forma,

```
diff(función, variable independiente)
```

Se puede observar un ejemplo similar, pero empleando 2 variables independientes, $f(x,y)=x^2\sin(yx)$

Se calcula $\dfrac{df(x,y)}{dx}$, $\dfrac{df(x,y)}{dy}$

```
(%i8) diff (x^2*sin(y*x),x);
(%o8) 2 x sin(x y)+x^2 y cos(x y)

(%i9) diff (x^2*sin(y*x),y);
(%o9) x^3 cos(x y)
```

Código 3.11 Diferenciación en Maxima.

La función `diff()`, también permite realizar la derivada *n* de la función deseada. En este caso, se pretende calcular la derivada décima de la función,

$$f(x)=e^{ax}$$

```
(%i12) diff (exp(a*x),x,10);
(%o12) a^10 %e^(a x)
```

Código 3.12 Diferenciación en Maxima.

Maxima permite definir la dependencia de funciones a variables y realizar sus derivadas sin tener que declararlas. A continuación se muestra un ejemplo.

Sea $f(u,v)$, $g(u,v)$, calcular la derivada de la función,

$$h(u,v)=\frac{f(u,v)}{g(u,v)}$$

```
(%i13) depends ([f,g], [u,v]);
(%o13) [f(u,v),g(u,v)]

(%i14) diff (f/g,u);
```

$$\text{(\%o14)}\quad \frac{\frac{d}{d\,u} f}{g}-\frac{f\left(\frac{d}{d\,u} g\right)}{g^2}$$

Código 3.13 Diferenciación en Maxima.

Se puede observar en el ejemplo que se ha empleado la función `depends()` para definir la dependencia de unas variables con otras variables.

Se mostrará ahora como resolver el cálculo de la derivada de manera numérica con Scilab. Una definición de derivada de una función para un valor determinado de la variable independiente,

$$\frac{df(x)}{dx} = \lim_{h\to 0} \frac{f(x+h) - f(x)}{(x+h) - x} = \lim_{h\to 0} \frac{f(x+h) - f(x)}{h}$$

Desde el punto de vista numérico, se puede conseguir un valor numérico preciso de la derivada en un punto de una función si se puede calcular el valor de la función lo suficientemente cerca del punto de evaluación. Se propone la siguiente función como ejemplo,

Ejemplo 3.4 *Calcular numéricamente la derivada de la función entre [0,3],*

$$f(x) = \sin(x^2)$$

Se propone en siguiente código de Scilab para calcular la aproximación numérica de la derivada como se ha mostrada previamente.

```
x=linspace(0,3,50);
h=10D-2;
x_h=x+h;
fx=sin(x.^2);
fx_h=sin(x_h.^2);
d_fx=(fx_h-fx)/h;
plot(x,d_fx)


real_d_fx=2*x.*cos(x.^2);
plot(x,real_d_fx);
```

Código 3.14 Diferenciación numérica con Scilab*.

En este caso, se destaca el error producido al no emplear un valor de *h*, paso, suficientemente pequeño. Este error se puede aproximar tanto a *0* como se desee.

(*) NOTA: Como en este caso el código empleado es más largo que en algunos de los problemas anteriores, se ha decido realizar la secuencia del programa en Scinotes, que permite guardar un programa y ejecutarlo de una sola vez. Los colores que emplea Scinotes se deben a que cuenta con un analizador sintáctico. En apartados posteriores se insistirá en el uso esta herramienta.

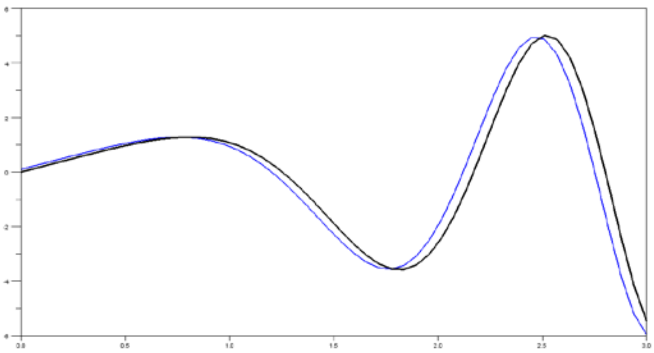


Figura 3.3 Solución gráfica exacta de la derivada y su aproximación numérica en Scilab.

La línea negra corresponde a la solución exacta y la azul a la aproximada numéricamente.

Scilab ofrece el uso de una función para simplificar el proceso de la diferenciación numérica. La función `diff()` calcula la diferencia entre dos valores consecutivos en la función. La función `diff()` permite calcular la diferencia de manera iterativa, pudiendo obtener la derivada numérica enésima. A continuación se muestra el ejemplo anterior empleando esta función. En este caso solo se emplea la evaluación de los puntos de la función definidos por *x*, siendo el paso la distancia entre dichos puntos (*3/49*). Para calcular la segunda derivada se emplearía la sentencia, `dfx=diff(fx,2)/h`, y así sucesivamente.

```
x=linspace(0,3,50);
h=3/49;
fx=sin(x.^2);
dfx=diff(fx)/h;
plot(x(1,1:49),dfx);

real_d_fx=2*x.*cos(x.^2);
plot(x,real_d_fx);
```

Código 3.15 Diferenciación numérica con Scilab.

Las gráficas obtenidas son similares a las obtenidas con el código anterior.

3.1.3 Integración

Las herramientas matemáticas que se emplean en este curso permiten realizar cálculos simbólicos y numéricos de integración.

En primer lugar se muestran algunas de las capacidades de cálculo simbólico de Maxima. Una de las funciones más potentes de Maxima es `integrate()`. Considera dos tipos de integrales, las integrales indefinidas y las integrales definidas. La sintaxis de la función es la siguiente,

`integrate(funcion,variable)`, para las integrales indefinidas.

`integrate(funcion, variable, limite inferior, limite superior)`, para las integrales definidas.

Se muestra a continuación algunos ejemplos.

<u>Integral indefinida con Maxima</u>
A continuación se muestra algunos ejemplos para ilustrar el cálculo simbólico de las integrales indefinidas con Maxima.

Ejemplo 3.5 *Calcular las siguientes integrales indefinidas,*

$$f(x) = \int b^x dx$$
$$f(x) = \int \frac{1}{\sqrt{1-x^2}} dx$$
$$f(x) = \int 1 + \tan^2(x) dx$$

```
(%i1) integrate (b^x, x);
         x
        b
(%o1) ------
      log(b)

(%i6) integrate (1/(1-x^2)^0.5, x);
(%o6) asin(x)

(%i5) integrate (1+tan(x)^2, x);
(%o5) tan(x)
```

Código 3.16 Integración indefinida en Maxima.

Maxima permite resolver integrales de forma exponencial, trigonométrica y trigonométrica inversa.

<u>Integral definida con Maxima</u>
También resuelve integrales definidas.

```
(%i24) integrate (x^0.5, x,0,1);
rat: replaced 1.5 by 3/2 = 1.5
rat: replaced 0.66666666666667 by 2/3 = 0.66666666666667
       2
(%o24) -
       3

(%i23) integrate (%pi*(r^2-x^2), x,-r,r);
            3
       4 π r
(%o23) ------
         3
```

Código 3.17 Integración definida en Maxima.

Se puede observar como se pueden resolver numéricamente o simbólicamente los ejemplos propuestos. En el último ejemplo se puede ver el cálculo de la integral de revolución de una esfera parametrizando los límites de la integral.

<u>Integral definida con Scilab</u>
El cálculo numérico de las integrales definidas se pueden realizar en Scilab mediante la función `intg()`. La sintaxis de la función `intg()`es la siguiente,

```
intg(limite inferior, limite superior, función)
```

Esta función, admite además otros parámetros para definir el umbral de error de cálculo.

Otra función de similares características a la mostrada anteriormente es la función `intergrate()`. Admite los mismos parámetros que la función `intg()`. A continuación se muestra un sencillo ejemplo de su uso. En el ejemplo se puede observar como se declara como parámetro el nombre de la variable a integrar.

```
-->x_s=1;

-->x_i=-1;

-->Ix=integrate('exp(x^2)','x',x_i,x_s)
 Ix  =

    2.9253035
```

Código 3.18 Integración definida en Scilab con `integrate()`.

Ejemplo 3.6 *Calcular el volumen de una esfera de radio unidad,*

Se propone el siguiente código de Scilab para solucionar el problema,

```
-->function y=f(x), y=%pi*(1^2-x^2),endfunction

-->sup=intg(-1,1,f)
 sup  =

    4.1887902

-->4*%pi/3
 ans  =

    4.1887902
```

Código 3.19 Integración definida en Scilab.

Las funciones mostradas anteriormente que realizan integrales definidas resueltas numéricamente con Scilab calculan los valores realizando una aproximación rectangular. La función `inttrap()` realiza una aproximación trapezoidal, mejorando la precisión de cálculo. Una mejor aproximación es la realizada mediante una interpolación de splin, mediante la función `intsplin()`. A continuación se muestra un ejemplo de integración con dichas funciones.

```
-->t=-1:0.2:1;

-->intsplin(t,exp(t^2))
 ans  =

    2.9260053

-->inttrap(t,exp(t^2))
 ans  =

    2.9613091
```

Código 3.20 Integración definida en Scilab, otras aproximaciones.

Cambio de variable

Maxima permite definir cambios de variable que facilitan la resolución simbólica de las integrales. Para hacer un cambio de variable se emplea la función `changevar()`. La sintaxis de la función es la siguiente,

```
chagevar(función, función a sustituir, nueva variable, variable antigua)
```

En el siguiente ejemplo se observa como se puede realizar un cambio de variable.

```
(%i41) 'integrate (x^2/(1+2*x)^(1/3), x);
       changevar (%, 1+2*x-t^3, t, x);
       ev (%, nouns);
```

(%o41) $$\int \frac{x^2}{(2\,x+1)^{1/3}}dx$$

(%i42)

(%o42) $$\frac{\int 3\,t^7-6\,t^4+3\,t\,dt}{8}$$

(%o43) $$\frac{\frac{3\,t^8}{8}-\frac{6\,t^5}{5}+\frac{3\,t^2}{2}}{8}$$

Código 3.21 Cambio de variable en Maxima.

La primera sentencia define la integral sin evaluarla.

$$\int \frac{x^2}{\sqrt[3]{1+2x}}dx$$

La segunda sentencia realiza la sustitución,

$$f(x,t)=0$$
$$1+2x-t^3=0$$

y define que la nueva variable independiente es t. Finalmente se realiza la evaluación de la nueva integral.

Funciones especiales

Tanto Maxima como Scilab admiten algunas funciones especiales, como las funciones de Bessel, la función Gamma y Beta entre muchas otras.

En el ejemplo que se muestra a continuación Maxima resuelve la integral de gamma y de beta. Cuando es capaz de calcular el resultado exacto, lo devuelve como salida. En el último caso, como no es capaz de hacerlo, la salida es la función beta con los parámetros seleccionados, que puede ser manipulada simbólicamente.

```
(%i6) gamma(-1/2);
(%o6) -2 √π
(%i7) beta(1/2,1);
(%o7) 2
(%i8) beta(1/3,1/2);
(%o8) β(1/3, 1/2)
```

Código 3.22 Función Beta y Gamma en Maxima.

Scilab también es capaz de calcular numéricamente estas funciones, en el siguiente ejemplo, se muestran las mismas operaciones que se han realizado con Maxima.

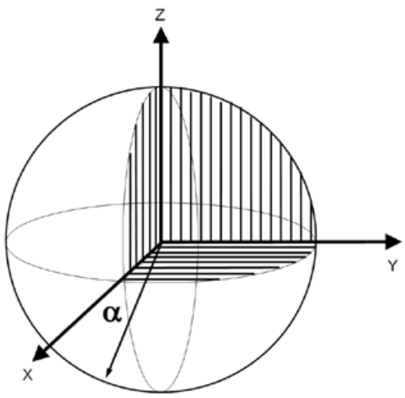
```
-->gamma(-1/2)
 ans  =

  - 3.5449077

-->beta(1/2,1)
 ans  =

    2.

-->beta(1/3,1/2)
 ans  =

    4.2065463
```

Código 3.23 Función Beta y Gamma en Scilab.

Integrales múltiples

Scilab resuelve integrales dobles y triples. Se ilustra estas capacidades con el ejemplo mostrado a continuación. De manera similar a uno de los ejemplos anteriores, se calcula el volumen de una sección de una esfera empleando una integral doble.

Ejemplo 3.7 *Calcular el volumen de una porción de la esfera correspondiente a* $x=[0,\cos\alpha]$, $y=[0,\mathrm{sen}\,\alpha]$, $\alpha=[0,\pi/2]$.



Para calcular el 1/8 del volumen de la esfera se debe calcular la siguiente integral doble.

$$\iint_S \sqrt{r^2 - x^2 - y^2}\,dxdy$$

La manera de representar esta función no es la más adecuada para representar los límites de la integral, sería más recomendable usar coordenadas polares. Sin embargo para emplear la función `int2d()` de Scilab es mejor el uso de coordenadas cartesianas. La razón es que se debe realizar un mallado triangular del espacio a integrar. En nuestro problema se divide la superficie en triángulos tal y como se puede ver en la siguiente figura.

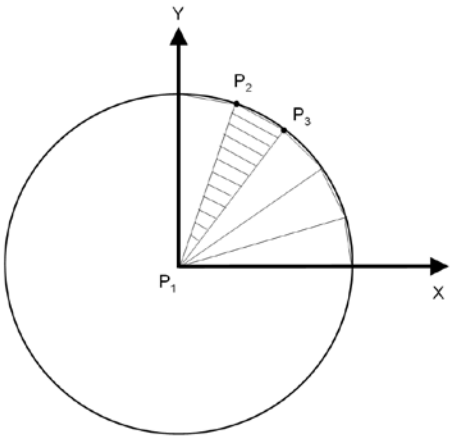


Figura 3.4 Aproximación geométrica a la solución.

Se observa como el cálculo será más preciso cuando el número de triángulos sea mayor. La sintaxis de la función `int2d()`es la siguiente,

```
[Resultado integral, error]= int2d(X, Y, función a integrar)
```

donde X e Y son las matrices de puntos de los triángulos que conforman la superficie a integrar. El formato de X e Y es el siguiente

$$X = \left[x_1^1, x_1^2, ..., x_1^N ; x_2^1, x_2^2, ..., x_2^N ; x_3^1, x_3^2, ..., x_3^N \right]$$
$$Y = \left[y_1^1, y_1^2, ..., y_1^N ; y_2^1, y_2^2, ..., y_2^N ; y_3^1, y_3^2, ..., y_3^N \right]$$

donde x_i^j es la coordenada x del punto i del triangulo j. En el problema propuesto, se calcula el error que supone la aproximación del sector de círculo a integral por medio de triángulos, viendo como afecta al error el número de triángulos que van a ser empleados.

```
Tri_min=3;
Tri_max=10;

for j=Tri_min:Tri_max

    N=j;
    r=1;

    X1=zeros(1,N);
    Y1=zeros(1,N);
    for i=0:N-1
        X2(i+1)=r*cos(i*%pi/(2*N));
        Y2(i+1)=r*sin(i*%pi/(2*N));
        X3(i+1)=r*cos((i+1)*%pi/(2*N));
        Y3(i+1)=r*sin((i+1)*%pi/(2*N));
    end

    X=[X1;X2';X3'];
    Y=[Y1;Y2';Y3'];

    deff('z=f(x,y)','z=(r^2-x^2-y^2)^0.5');
    [I(j-Tri_min+1),e(j-Tri_min+1)]=int2d(X,Y,f);
end

plot(I-(%pi/6))
```

Código 3.24 Solución del ejemplo en Scilab.

Para realizar los cálculos propuestos se emplea el código anterior. Se define el número de triángulos mínimo y el máximo (3 y 10).

- Un bucle `for` recorre los distintos casos de número de triángulos.
- Se declara el radio de la esfera `r=1`.
- Se asigna las coordenadas del primer punto, común a todos los triángulos, correspondiente a la coordenada (0,0).
- Se construye otro bucle `for` que calcula los puntos de triángulos de la misma superficie.
- Una vez calculados los puntos, se almacenan los puntos en matrices con el formato adecuado.
- Se define la función a integrar mediante la función `deff()`.
- Se aplica la función de integración doble `int2d()`.
- Finalmente se crea una representación gráfica que muestra el error entre las distintas aproximaciones por el número de triángulos y el valor exacto de la integral.

La gráfica obtenida se muestra a continuación. Se puede observar que la abscisa de la gráfica muestra la posición del dato en la matriz, el valor 1 corresponde al cálculo con 3 triángulos y así sucesivamente.

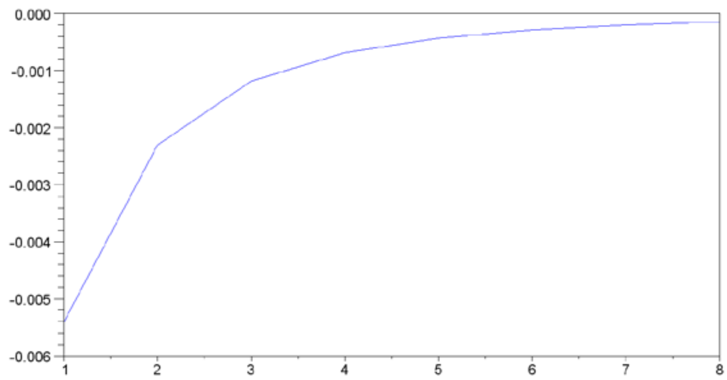


Figura 3.5 Representación gráfica del error de la solución en función del número de triángulos.

De manera similar, Scilab permite resolver integrales triples empleando la función `int3d()`, esta función precisa la definición del mallado por tetraedros del espacio a integrar.

<u>Integrales de variable complejas</u>

Scilab permite calcular integrales de variable compleja de Cauchy. La sintaxis de la función `intc()`que realiza esta tarea se muestra a continuación,

```
intc(límite inferior, limite superior, función a integrar)
```

Esta función realiza la integración de la función de variable compleja recorrido en línea recta en el plano complejo del punto inferior del límite al superior.

Ejemplo 3.8 Calcular la siguiente integral definida,

$$\int_0^{\frac{\pi}{4}} e^{it}dt$$

Para calcular esta integral se emplea la función `intc()`,

```
-->deff('y=f(t)','y=%e^(%i*t)')

-->intc(0,%pi/4,f)
 ans  =

    0.7071068 + 0.2928932i
```

Código 3.25 Integral de Cauchy en Scilab.

EJERCICIOS:

Ejercicio 3.1 *Demostrar las siguientes identidades,*

- $\lim_{x\to 0}(1+x)^{\frac{1}{x}} = e$
- $\lim_{x\to 0}\frac{\tan(x)}{x} = 1$

Ejercicio 3.2 *Calcular las derivadas de las siguientes funciones,*

- $f(x) = (2x^{\frac{2}{3}} + \tan(x))(1 - e^{-x^2})$ *x=1*
- $f(x) = \dfrac{3ax^3 + 2bx + 4}{8x^4 + 6x^2 + 1}$

Ejercicio 3.3 *Calcular las ecuaciones de las rectas tangentes a la función en los puntos x=2 x=2.5,*

$$f(x) = x^2 e^{-x} \sin(x)$$

Ejercicio 3.4 *Calcular el valor de x donde la función alcanza el valor máximo,*

$$f(x) = 4 - (x-2)^2$$

Ejercicio 3.5 *Determinar las dimensiones del cono de volumen mínimo circunscrito a una semiesfera de radio R, y donde el plano de la base del cono coincida con la base de la semiesfera.*

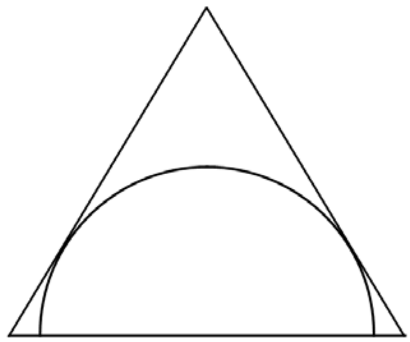


Figura 3.7 Perfil del cono y de una semiesfera circunscrita a éste.

Ejercicio 3.6 *La posición de un objeto en el eje x en función del tiempo está definida mediante función,*

$$x(t) = 7\cos(t)t^2$$
$$t \in [0,4]$$

Calcular la expresión de la velocidad v(t) y de la aceleración a(t) y mostrar su representación gráfica.

$$v(t) = \frac{dx(t)}{dt}$$
$$a(t) = \frac{dv(t)}{dt}$$

Ejercicio 3.7 *Calcular las siguientes integrales,*

- $\int (1-x)e^{x^2+3x+7}dx$
- $\int 2^x \sin(x)dx$

- $\int_{b}^{a} \frac{x}{x^2+1} dx$
- $\int_{0}^{1} x^{\frac{1}{2}} \cos^2(x) dx$

Ejercicio 3.8 *Obtener una expresión general de la superficie comprendida entre una parábola que corta en la abscisa en 0 y 2 y la rectas tangentes a la parábola en esos puntos,*

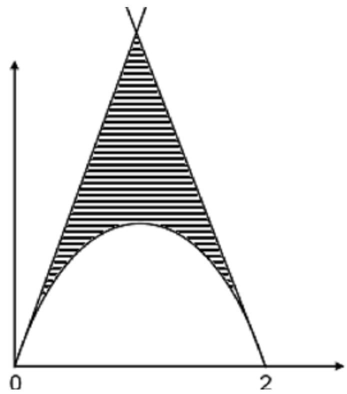


Figura 3.6 Superficie comprendida entre parábola y rectas.

3.2 FACTORIZACIÓN, SIMPLIFICACIONES Y DESARROLLOS EN SERIE

3.2.1 Factorización y simplificaciones

Factorización

La factorización de una expresión consiste en la construcción de otras expresiones equivalentes formadas mediante el producto de expresiones más pequeñas. La factorización puede ser realizada sobre números, polinomios u otras expresiones simbólicas. Por ejemplo, un caso particular de factorización es el cálculo de las raíces de un polinomio, en cuyo caso se calculan los factores del producto que aportan información sobre los valores que hacen *0* dicho polinomio.

En este apartado se mostrará la factorización y las simplificaciones, ya que en su conjunto, estas funciones permiten obtener distintas expresiones de una misma solución. Este apartado se centrará especialmente en el cálculo simbólico para la factorización y las simplificaciones. Por esta razón en este apartado se empleará especialmente Maxima.

La función principal de Maxima para la factorización es `factor()`. En el siguiente ejemplo se muestra la factorización de número natural, entero, decimal de coma flotante y racional.

```
(%i17) factor(100);
(%o17) 2^2 5^2

(%i18) factor(-102);
(%o18) -2 3 17

(%i19) factor(74.43);
rat: replaced 74.43000000000001 by 7443/100 = 74.43000000000001
(%o19) 3^2 827 / (2^2 5^2)

(%i20) factor(1085/27);
(%o20) 5 7 31 / 3^3
```

Código 3.26 Factorización en Maxima.

Los factores obtenidos con esta función serán irreductibles mediante enteros. Maxima puede realizar la factorización de polinomios empleando la función `factor()`. A continuación se factoriza el polinomio $y(x) = x^4 - 1$.

```
(%i18) factor(x^4-1);
(%o18) (x-1)(x+1)(x^2+1)
```

Código 3.27 Factorización de polinomios en Maxima.

La función `factor()` admite el uso de más de una variable simbólica. En el siguiente ejemplo se ilustra la factorización de los siguientes polinomios,

$$y(x,a) = x^2 - a^2$$
$$z(x,y,a,b) = yx^2 - x^2b - a^2y + ba^2$$

```
(%i31) factor(x^2-a^2);
(%o31) (x-a)(x+a)

(%i32) factor(y*x^2-x^2*b-a^2*y+a^2*b);
(%o32) (x-a)(x+a)(y-b)
```

Código 3.28 Factorización de polinomios en Maxima.

Como ya se ha comentado anteriormente, la factorización puede ser empleada para calcular las raíces de un polinomio. Sin embargo, la función `factor()` está orientada a la factorización de números enteros.

Se muestra a continuación el empleo de la función `factor()`para factorizar el polinomio $y(x) = x^2 + 0.75x + 0.125$.

```
(%i50) factor(x^2+0.75*x+0.125);
rat: replaced 0.125 by 1/8 = 0.125
rat: replaced 0.75 by 3/4 = 0.75
(%o50) (2 x+1)(4 x+1)
       --------------
             8

(%i51) allroots(x^2+0.75*x+0.125);
(%o51) [x=-0.25,x=-0.5]
```

Código 3.29 Cálculo de las raíces de un polinomio en Maxima.

En el ejemplo anterior se observa que al aplicar la función `factor()` sobre polinomios con coeficientes reales, se realiza de manera automática una conversión numérica previa a la operación de factorización para así poder operar con números enteros.
En el caso particular del ejemplo, multiplica todo el polinomio por el valor *8*, de esta manera se puede obtener las mismas raíces del polinomio que al emplear la función `allroots()`, $x=-0.25$ y $x=-0.5$, lo que significa que el resultado de la factorización del polinomio es,

$$y(x)=(x+0.25)(x+0.5)$$

A continuación se estudia la factorización del polinomio de segundo grado $y(x)=x^2+3x-7$. En este caso los coeficientes son enteros, por lo que la función `factor()` no realiza ninguna modificación del polinomio. Tampoco es capaz de encontrar ninguna solución. En este caso, la mejor manera de realizar la factorización con números reales o complejos es empleando la función `allroots()`,

```
(%i52) factor(x^2+3*x-7);
(%o52) x^2+3 x-7

(%i53) allroots(x^2+3*x-7);
(%o53) [x=1.54138126514911,x=-4.54138126514911]
```

Código 3.30 Raíces de un polinomio en Maxima.

La factorización del polinomio del ejemplo sería por lo tanto,

$$y(x)=(x-1.54138126514911)(x+4.54138126514911)$$

Ejemplo 3.9 *Calcular de los autovalores de la matriz.*

$$\begin{pmatrix} 2 & 1 & 7 \\ 5 & 2 & 4 \\ -2 & 4 & 3 \end{pmatrix}$$

Se pudo ver en el ejemplo del tema 2, donde se calcularon los autovalores de una matriz de 2x2 se emplea la función `solve()`, y mediante la función específica `eigenvalues()`. En este ejemplo, se obtienen los autovalores calculando las raíces del polinomio característico, que puede ser calculado realizando la factorización del polinomio.

Las operaciones que a realizar serán las mismas,

```
(%i56) M:matrix([2,1,7],[5,2,4],[-2,4,3]);
```

$$(\%o56)\quad \begin{bmatrix} 2 & 1 & 7 \\ 5 & 2 & 4 \\ -2 & 4 & 3 \end{bmatrix}$$

```
(%i60) M-l*ident(3);
```

$$(\%o60)\quad \begin{bmatrix} 2-l & 1 & 7 \\ 5 & 2-l & 4 \\ -2 & 4 & 3-l \end{bmatrix}$$

```
(%i61) q_l:determinant(%);
(%o61) -5(3-l)+((2-l)(3-l)-16)(2-l)+7(2(2-l)+20)-8
```

```
(%i62) factor(q_l);
```

$$(\%o62)\quad -(l^3-7\,l^2+9\,l-125)$$

```
(%i63) allroots(q_l);
(%o63) [l=3.960507873676131 %i-0.43668206679136,l=-3.960507873676131 %i-
0.43668206679136,l=7.873364133582728]
```

Código 3.31 Cálculo de autovalores de una matriz en Maxima.

Se obtiene finalmente el polinomio característico `q_l`, donde se emplea la función `factor()` para factorizar en el caso de que el polinomio tenga raíces enteras. Se puede observar que no se obtiene resultado. Se emplea la función `allroots()` sobre la función `q_l`. Se obtienen dos raíces imaginarias conjugadas y una raíz real.

La factorización expandida del polinomio característico, con los coeficientes truncados a 3 decimales sería el siguiente,

$$y(x) = \left(x + (0.436 - 3.960\mathrm{i})\right)\left(x + (0.436 + 3.960\mathrm{i})\right)(x - 7.873)$$

Se puede realizar una factorización dirigida a la agrupación de variables de un polinomio. La función `factorout()`permite realizar esta factorización por variables agrupadas. La sintaxis de la función es la siguiente,

```
factorout(expresión, var1,var2,…,varN)
```

A continuación se propone un ejemplo. Se pretende la factorización del siguiente polinomio, agrupando las variables *x* e *y*.

$$z(x,y,a,b,c) = ac(y+x)^2 + bxy + 2by + ax^2 + bx + 2ax + 2b + a$$

```
(%i67) factorout(ac(y+x)^2+b*x*y+2*b*y+a*x^2+b*x+2*a*x+2*b+a,x,y);
```

$$(\%o67)\quad ac(y+x)^2+b(x+2)(y+1)+a(x+1)^2$$

Código 3.32 Agrupación de variables en Maxima.

Se obtiene la siguiente factorización,

$$z(x,y,a,b,c) = ac(y+x)^2 + b(x+2)(y+1) + a(x+1)^2$$

En algunas ocasiones, puede ser de interés obtener la factorización de un polinomio a partir de un determinado factor. Para realizar esta tarea se puede emplear la función de Maxima `gdc()`. Esta función puede tener como argumentos dos polinomios, donde la salida es el máximo común divisor. La salida 1 se produce cuando no se encuentra un máximo común divisor distinto de 1. Esta función, unida a la función `divide()`, permite realizar una factorización dirigida. La función `divide()` tiene como argumentos los polinomios dividendo y divisor. La salida es el cociente y el resto de la división. Por lo tanto, la función `divide()` también ofrece información sobre si un polinomio es factor de otro polinomio, pudiendo evaluar que el resto sea igual a *0*.

***Ejemplo 3.10** Buscar en los siguientes polinomios el factor común* $(x+1)$.

$$y = x^3 + 3x^2 + 3x + 1$$
$$y = x^4 + bx^3 + 3x^3 + 3bx^2 + 3x^2 + 3bx + x + b$$
$$y = x^2 + 4x + 4$$

```
(%i88) gcd(x^3+3*x^2+3*x+1,x+1);
(%o88) x+1

(%i89) gcd(x^4+b*x^3+3*x^3+3*b*x^2+3*x^2+3*b*x+x+b,x+1);
(%o89) x+1

(%i90) gcd(x^2+4*x+4,x+1);
(%o90) 1

(%i91) divide(x^3+3*x^2+3*x+1,x+1);
(%o91) [x^2+2 x+1 , 0 ]

(%i92) divide(x^4+b*x^3+3*x^3+3*b*x^2+3*x^2+3*b*x+x+b,x+1);
(%o92) [x^3+(b+2) x^2+(2 b+1) x+b , 0 ]

(%i93) divide(x^2+4*x+4,x+1);
(%o93) [x+3 , 1 ]
```

Código 3.33 Búsqueda de factores comunes en Maxima.

Se puede observar como los dos primeros polinomios del ejemplo, empleando la función `gcd()`, tienen como factor $(x+1)$. Con la función `divide()` se puede conocer que es factor, y además muestra como salida el valor del otro factor. A continuación se muestran los polinomios de los ejemplos factorizados.

$$y = x^3 + 3x^2 + 3x + 1 = (x+1)(x^2 + 2x + 1)$$
$$y = x^4 + bx^3 + 3x^3 + 3bx^2 + 3x^2 + 3bx + x + b = (x+1)(x^3 + (b+2)x^2 + (2b+1)x + b)$$

Simplicación

En este apartado se muestra la capacidad de Maxima para representar de distintos modos una misma expresión. No todas las funciones mostradas a continuación obtienen expresiones más compactas que las originales, a veces puede ser útil expandir las expresiones.

La función `expand()`, como indica el nombre, expande una expresión en sumandos. Permite la expansión de polinomios y de expresiones racionales tal y como se ve en el ejemplo.

```
(%i2) expand((x+1)*(x+b)*(y+c));
```

(%o2) $x^2\,y+b\,x\,y+x\,y+b\,y+c\,x^2+b\,c\,x+c\,x+b\,c$

```
(%i4) expand((x+1)^2/(x+2)^3);
```

(%o4) $\dfrac{x^2}{x^3+6\,x^2+12\,x+8}+\dfrac{2\,x}{x^3+6\,x^2+12\,x+8}+\dfrac{1}{x^3+6\,x^2+12\,x+8}$

Código 3.34 Empleo de la función `expand()` en Maxima.

La función `rat()` convierte una expresión racional a la forma canónica. Para ilustrar el uso de esta función, en el siguiente ejemplo se define una función con 2 variables. Se aplica la función `rat()` primero sobre la variable `a` y en la siguiente sentencia sobre la variable `x`.

```
(%i28) F:(20*a*x^3+a)*(x+a)/(27*a*x+3*a+x^4)/(12*a^2*x);
```

(%o28) $\dfrac{(x+a)(20\,a\,x^3+a)}{12\,a^2\,x\,(x^4+27\,a\,x+3\,a)}$

```
(%i29) rat(%,a);
```

(%o29)/R/ $\dfrac{(20\,x^3+1)\,a+20\,x^4+x}{(324\,x^2+36\,x)\,a^2+12\,x^5\,a}$

```
(%i30) rat(%,x);
```

(%o30)/R/ $\dfrac{20\,x^4+20\,a\,x^3+x+a}{12\,a\,x^5+324\,a^2\,x^2+36\,a^2\,x}$

Código 3.35 de la función `rat()` en Maxima.

Otra función que puede resultar de utilidad en el proceso de la simplificación es la función `partfrac()`. Esta función descompone una función racional en suma de formas racionales mediante la expansión en fracciones parciales.

```
(%i43) F:x/(x^2+3*x+2);
(%o43) x/(x^2+3x+2)
(%i44) partfrac(F,x);
(%o44) 2/(x+2)-1/(x+1)
(%i45) F:x/(x^2+(2+a)*x+2*a);
(%o45) x/(x^2+(a+2)x+2a)
(%i46) partfrac(F,x);
(%o46) a/((a-2)(x+a))-2/((a-2)(x+2))
(%i47) partfrac(F,a);
(%o47) x/((x+2)(x+a))
```

Código 3.36 Descomposición de función racional en fracciones parciales en Maxima.

La función `partfrac()` permite definir la variable principal sobre la que realizar la simplificación, permitiendo que pueda ser empleada en expresiones con más de una variable.

El uso de las funciones de simplificación y las funciones de factorización mostradas en los apartados anteriores, y de otras funciones con estas características, pueden consultarse en el manual de ayuda de Maxima, especialmente en los apartados *7:Simplificación* y *12:Polinomios*.

3.2.1 Desarrollos en serie

Los desarrollos en serie, al igual que la factorización de funciones y la simplificación, mostradas en apartados anteriores, están en el contexto del cálculo simbólico. Por esta razón, se emplea Maxima como herramienta para realizar este tipo de cálculos.

En primer lugar se mostrará como se definen los sumatorios y los productorios. En Maxima se emplea la función `sum()` para representar fácilmente los sumatorios. La sintaxis de la función es la siguiente,

`sum(expresión(contador),contador,valor_min_contador,valor_max_contador)`,

A continuación se muestra un ejemplo de su uso para series geométricas con un número finito de elementos. En el primer caso, donde la razón es $1/6$, se puede observar que se ha obtenido un resultado numérico. En el segundo caso, empleando coeficientes simbólicos, Maxima muestra como salida la suma de manera expandida,

```
(%i15) sum(1.7/6^i,i,0,12);
(%o15) 2.039999999843806
```

```
(%i16) sum(a/z^i,i,0,5);
```

(%o16) $\frac{a}{z}+\frac{a}{z^2}+\frac{a}{z^3}+\frac{a}{z^4}+\frac{a}{z^5}+a$

```
(%i6) sum(1/2^i,i,0,inf),simpsum;
(%o6) 2
```

Código 3.37 Sumatorio con un número finito de términos.

Seguidamente se muestra un ejemplo de un sumatorio de infinitos términos de una serie geométrica.

```
(%i5) sum(1/2^i,i,0,inf);
```

(%o5) $\sum_{i=0}^{\infty} \frac{1}{2^i}$

```
(%i6) sum(1/2^i,i,0,inf),simpsum;
(%o6) 2
```

Código 3.38 Sumatorio con un número infinito de términos.

Maxima obtiene una simplificación del resultado siempre que sea posible mediante la expresión `simpsum`.

Es posible realizar operaciones entre sumatorios, como multiplicación o sumas. A continuación se muestra un ejemplo,

```
(%i32) sum(1/2^i,i,0,inf)+sum(1/3^i,i,0,inf);
```

(%o32) $\left(\sum_{i=0}^{\infty} \frac{1}{3^i}\right)+\sum_{i=0}^{\infty} \frac{1}{2^i}$

```
(%i33) sum(1/2^i,i,0,inf)^2;
```

(%o33) $\left(\sum_{i=0}^{\infty} \frac{1}{2^i}\right)^2$

Código 3.39 Operaciones básicas entre sumatorios.

Maxima es capaz de expandir expresiones en series de Taylor. Una expansión en serie de Taylor de una función $f(x)$ alrededor de un punto x_0, se puede expresar,

$$f(x)=\sum_{n=0}^{\infty}\frac{f^{n}(x_0)}{n!}(x-x_0)^n$$

La función que realiza esta tarea tiene la siguiente sintaxis,

```
taylor(expresión, variable, x_0, nivel_de_truncamiento)
```

Se va a ilustrar el uso de esta función mediante un ejemplo.

Ejemplo 3.11 *Obtener la expansión de la serie de Taylor para la función,*

$$f(x) = x\sin(x)$$
$$x_0 = 0.5$$

En este caso se desea obtener hasta el tercer grado de la serie. Por lo tanto n debe ser *3*.

```
(%i4) f:x*sin(x);
(%o4) x sin(x)

(%i5) taylor(f,x,0.5,3);
rat: replaced 0.5 by 1/2 = 0.5
rat: replaced 0.5 by 1/2 = 0.5
rat: replaced 0.5 by 1/2 = 0.5
rat: replaced 0.5 by 1/2 = 0.5
rat: replaced 0.5 by 1/2 = 0.5
rat: replaced 0.5 by 1/2 = 0.5
rat: replaced 0.5 by 1/2 = 0.5
```

$$\text{(\%o5)/T/}\quad \frac{\sin\left(\frac{1}{2}\right)}{2}+\frac{\left(2\sin\left(\frac{1}{2}\right)+\cos\left(\frac{1}{2}\right)\right)(x-0.5)}{2}-\frac{\left(\sin\left(\frac{1}{2}\right)-4\cos\left(\frac{1}{2}\right)\right)(x-0.5)^2}{4}-\frac{\left(6\sin\left(\frac{1}{2}\right)+\cos\left(\frac{1}{2}\right)\right)(x-0.5)^3}{12}+...$$

Código 3.40 Serie de Taylor sobre un valor fijo.

En el ejemplo anterior se ha planteado la obtención numérica de la serie de Taylor. La función `taylor()` permite obtener un resultado simbólico de la serie.

Ejemplo 3.12 *Obtener la expansión de la serie de Taylor para la función,*

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

En este ejemplo se ha empleado una variable simbólica para representar el punto sobre el que es evaluada la serie, obteniendo una expresión general de los 4 primeros términos de la serie.

```
(%i8) f:x*sin(x);
(%o8) x sin(x)

(%i9) taylor(f,x,xo,3);
```

$$\text{(\%o9)/T/}\quad \sin(xo)\,xo+(\cos(xo)\,xo+\sin(xo))(x-xo)-\frac{(\sin(xo)\,xo-2\cos(xo))(x-xo)^2}{2}-\frac{(\cos(xo)\,xo+3\sin(xo))(x-xo)^3}{6}+...$$

Código 3.41 Serie de Taylor simbólica.

Se puede observar que Maxima representa el truncamiento de la serie mediante la expresión "…".

Se pueden generar expresiones que representan un truncamiento de una serie de Taylor, empleando la función `trunc()`. El uso de esta función se ilustra a través del ejemplo mostrado a continuación.

```
(%i31) trunc(1-x^2+x^4);
(%o31) 1-x^2+x^4+...
```

Código 3.42 Expresión truncada.

Una potente capacidad que ofrece Maxima es la obtención de las funciones que comparten un truncamiento de su serie de Taylor empleando la función `pade()`. Podría entenderse como la función inversa al desarrollo de Taylor de la función. La función resuelve las funciones racionales. La sintaxis de la función `pade()` es la siguiente.

```
pade(expresión, grado_numerador, grado_denominador)
```

En el siguiente ejemplo el grado del numerador es *0* y el del denominador es *2*. Puede observar que al seleccionar como tercer argumento el valor *1* no se obtiene ninguna salida.

```
(%i32) f:1/(x^2+1);
(%o32) 1/(x^2+1)
(%i33) T:taylor(f,x,0,5);
(%o33)/T/ 1-x^2+x^4+...
(%i34) pade(T,1,2);
(%o34) [1/(x^2+1)]
(%i35) pade(T,1,1);
(%o35) []
```

Código 3.43 Primitiva de un desarrollo de Taylor.

Maxima permite obtener una serie de potencias de una expresión alrededor de un punto x_0, con la forma,

$$f(x)=\sum_{i=0}^{\infty} a_i (x-x_0)^i$$

La función que lleva a cabo esta tarea es `powerseries()`, la sintaxis a emplear es la siguiente,

```
powerseries(expresión, variable, x_0)
```

A continuación se muestran unos ejemplos del uso de esta función.

Ejemplo 3.13 *Calcular el desarrollo en serie de potencias de las siguientes expresiones,*

$$f(x) = \frac{1}{x^3} \qquad x_0 = 1$$
$$f(x) = e^{x^2} \qquad x_0 = 0$$
$$f(x) = \sin^2(x) \qquad x_0 = 0$$

```
(%i43) powerseries(1/x^3,x,1);
       powerseries(exp(x^2),x,0);
       powerseries(sin(x)^2,x,0);
```

$$(\%o43)\quad \frac{\sum_{i8=0}^{\infty} (i8+1)(i8+2)(-1)^{i8}(x-1)^{i8}}{2}$$

$$(\%o44)\quad \sum_{i9=0}^{\infty} \frac{x^{2\,i9}}{i9!}$$

$$(\%o45)\quad -\frac{\left(\sum_{i10=0}^{\infty} \frac{(-1)^{i10}\, 2^{2\,i10}\, x^{2\,i10}}{(2\,i10)!}\right)-1}{2}$$

Código 3.44 Desarrollo en serie de potencias.

EJERCICIOS:

Ejercicio 3.9 *Calcular el error en la aproximación del polinomio de Taylor de la función* $f(x) = x^3$ *, en función del número de términos del polinomio.*

Ejercicio 3.10 *Calcular la aproximación por series de potencias de la función* $f(x) = a\sin(x)e^{bx}$ *en función de las variables b y x.*

3.3 SIMPLIFICACIONES TRIGONOMÉTRICAS

Maxima permite realizar manipulación de expresiones trigonométricas con el objetivo de obtener otras expresiones más compactas o que permitan una mejor comprensión. El título del apartado se refiere a la simplificación de expresiones, pero también se incluye algunas funciones que permiten expandir expresiones.

La función `trigreduce()` obtiene una expresión compuesta de productos de funciones senos y cosenos en sumas de funciones de senos y cosenos. A continuación se muestran algunos ejemplos.

Ejemplo 3.14 *Convertir las siguientes funciones en sumas de senos y cosenos,*

$$\sin^3(x)$$
$$\sin(x)\cos^2(x)$$

```
(%i66) expr:sin(x)^3;
```
(%o66) $\sin(x)^3$

```
(%i67) trigreduce(expr);
```
(%o67) $\dfrac{3\sin(x)-\sin(3x)}{4}$

```
(%i68) expr:sin(x)*cos(x)^2;
```
(%o68) $\cos(x)^2\sin(x)$

```
(%i69) trigreduce(expr);
```
(%o69) $\dfrac{\sin(3x)+\sin(x)}{4}$

Código 3.45 Ejemplo de uso de la función `trigreduce()`.

Otra transformación trigonométrica que ofrece Maxima, es la posibilidad de convertir una expresión con funciones trigonométricas de sumas de ángulos en otra expresión expandida de funciones trigonométricas de dichos ángulos. La función empleada para realizar esta tarea es `trigexpand()`. Para ilustrar las capacidades de esta función, a continuación se muestran algunos ejemplos de su uso,

```
(%i99) expr: (sin(4*x)*cos(3*y))/tan(2*x);
```
(%o99) $\dfrac{\sin(4x)\cos(3y)}{\tan(2x)}$

```
(%i100) trigexpand(expr);
```
(%o100) $\dfrac{(4\cos(x)^3\sin(x)-4\cos(x)\sin(x)^3)(1-\tan(x)^2)(\cos(y)^3-3\cos(y)\sin(y)^2)}{2\tan(x)}$

```
(%i101) expr:tanh(7*x)^3;
```
(%o101) $\tanh(7x)^3$

```
(%i102) trigexpand(expr);
```
(%o102) $\dfrac{(\tanh(x)^7+21\tanh(x)^5+35\tanh(x)^3+7\tanh(x))^3}{(7\tanh(x)^6+35\tanh(x)^4+21\tanh(x)^2+1)^3}$

```
(%i103) expr:tan(x+%pi/4);
```
(%o103) $\tan\left(x+\dfrac{\pi}{4}\right)$

```
(%i104) trigexpand(expr);
```
(%o104) $\dfrac{\tan(x)+1}{1-\tan(x)}$

Código 3.46 Ejemplos del uso de la función `trigexpand()`.

La función `trigsimp()`, realiza la simplificación de funciones trigonométricas como `sec()`, `tan()`, `cot()` o `csc()`, entre otras, en expresiones de senos y cosenos. A continuación se muestran algunos ejemplos de la capacidad de simplificación de esta función.

```
(%i42) expr:sec(x)^2+tan(y)^3;
```
(%o42) $\tan(y)^3+\sec(x)^2$

```
(%i43) trigsimp(expr);
```
(%o43) $\dfrac{\cos(x)^2\sin(y)^3+\cos(y)^3}{\cos(x)^2\cos(y)^3}$

```
(%i44) expr:cot(x+y^2)^2/csc(x^2+y)^(1/2);
```
(%o44) $\dfrac{\cot(y^2+x)^2}{\sqrt{\csc(y+x^2)}}$

```
(%i45) trigsimp(expr);
```
(%o45) $\dfrac{\sqrt{\sin(y+x^2)}\cos(y^2+x)^2}{\sin(y^2+x)^2}$

Código 3.47 Ejemplos del uso de la función `trigsimp()`.

Maxima cuenta con la función `trigrat()` que expande funciones trigonométricas en otra expresión trigonométrica con argumentos que son combinaciones lineales de sus argumentos. A continuación se muestran algunos ejemplos.

```
(%i72) expr:sin(3*x)*cos(x)/(cos(x+%pi/3)*tan(2*x));
```
(%o72) $\dfrac{\cos(x)\sin(3\,x)}{\tan(2\,x)\cos\left(x+\frac{\pi}{3}\right)}$

```
(%i73) trigrat(expr);
```
(%o73) $-\dfrac{\cos(4\,x)+\cos(2\,x)+1}{\sqrt{3}\sin(x)-\cos(x)}$

```
(%i74) expr:sin(3*x)*cos(y)/(cos(3*x+%pi/3)*tan(2*z));
```
(%o74) $\dfrac{\sin(3\,x)\cos(y)}{\cos\left(3\,x+\frac{\pi}{3}\right)\tan(2\,z)}$

```
(%i75) trigrat(expr);
```
(%o75) $\dfrac{\sin(2\,z+y+3\,x)-\sin(2\,z+y-3\,x)+\sin(2\,z-y+3\,x)-\sin(2\,z-y-3\,x)}{\sin(2\,z+3\,x)+\sqrt{3}\cos(2\,z+3\,x)+\sin(2\,z-3\,x)-\sqrt{3}\cos(2\,z-3\,x)}$

Código 3.48 Ejemplos del uso de la función `trigrat()`.

Se puede observar en los ejemplos que se puede emplear más de una variable para aplicar la función `trigrat()`.

Puede ser de gran utilidad convertir expresiones trigonométricas en funciones exponenciales. Maxima ofrece una función que realiza esta tarea, `exponentialize()`.

```
(%i84) expr:cos(n)+sin(n);
(%o84) sin(n)+cos(n)
```

```
(%i85) exponentialize(expr);
```

$$(\%o85)\quad \frac{\%e^{\%i\,n}+\%e^{-\%i\,n}}{2}-\frac{\%i\,(\%e^{\%i\,n}-\%e^{-\%i\,n})}{2}$$

```
(%i86) expr:tan(3*x)*sec(y);
(%o86) tan(3 x) sec(y)
```

```
(%i87) exponentialize(expr);
```

$$(\%o87)\quad -\frac{2\,\%i\,(\%e^{3\,\%i\,x}-\%e^{-3\,\%i\,x})}{(\%e^{3\,\%i\,x}+\%e^{-3\,\%i\,x})(\%e^{\%i\,y}+\%e^{-\%i\,y})}$$

Código 3.49 Ejemplos del uso de la función `exponentialize()`.

Ejemplo 3.15 *Demuestre la siguiente identidad,*

$$\frac{\operatorname{sen}(2x)}{1+\cos(2x)}=\tan(x)$$

Para demostrar la identidad propuesta se han encadenado dos funciones de simplificación, en primer lugar se obtiene una expresión con funciones trigonométricas con argumentos de las variables sin coeficiente, `trigrat()`, y posteriormente se obtiene una expresión más compacta con la función `trigreduce()`.

```
(%i158) expr:sin(2*x)/(1+cos(2*x));
```

$$(\%o158)\quad \frac{\sin(2\,x)}{\cos(2\,x)+1}$$

```
(%i159) trigrat(expr);
```

$$(\%o159)\quad \frac{\sin(x)}{\cos(x)}$$

```
(%i160) trigreduce(%);
(%o160) tan(x)
```

Código 3.50 Resolución de ejemplo de simplificación trigonométrica.

Ejemplo 3.16 *Demuestre la siguiente identidad,*

$$\frac{\sin(a+b)}{\cos(a)\cos(b)}=\tan(a)+\tan(b)$$

El ejemplo propuesto se resuelve de manera similar al ejemplo anterior, en este caso en concreto, como la parte derecha de la ecuación está expresada en función de las dos variables separadas, se aplica la función `trigexpand()` para separar la suma de ángulos. Como la solución es simplificable, se emplea la función `expand()`. Finalmente se aplica la función `trigreduce()` para obtener una forma más compacta.

```
(%i161) expr:(sin(a+b)/cos(a)/cos(b));
(%o161)  sin(b+a)
        ---------------
        cos(a) cos(b)

(%i162) expr2:trigexpand(expr);
(%o162)  cos(a) sin(b)+sin(a) cos(b)
        ------------------------------
               cos(a) cos(b)

(%i163) expand(expr2);
(%o163)  sin(b)   sin(a)
        ------ + ------
        cos(b)   cos(a)

(%i164) trigreduce(%);
(%o164) tan(b)+tan(a)
```

Código 3.51 Resolución de ejemplo de simplificación trigonométrica.