

Actividad 9: Sistema de Algebra Computacional Maxima

Melissa Matrecitos Avila

28 de Abril de 2018

1 Introducción



El siguiente reporte corresponde a la actividad 9 del curso de Física Computacional 1, la cual se enfoca a la solución simbólica de ecuaciones, mediante la utilización de software especializado.

Un sistema de álgebra computacional (CAS), es una pieza de software matemático. Permite a un usuario escribir álgebra, cálculo y preguntas estadísticas, usando comandos que son muy parecidos al lenguaje humano, e intenta responder usando la potencia computacional de la computadora.

Una de las CAS más antiguas que existe es una llamada Macsyma. Comenzó su vida de software en 1968 por investigaciones en el MIT. De este se derivaron algunos softwares, uno de ellos fue Maxim. Esta versión fue abierta en 1999 y continúa desarrollándose como un proyecto de código abierto hasta el día de hoy. En específico, WxMaxima es la cara bonita de la vieja escuela Maxima CAS. Nos permite escribir texto, hacer informes y, lo que es más importante, tiene menús y botones que nos ayudan a recordar los comandos que necesitamos.

EL reporte presenta un pequeño manual sobre uno de los aspectos básicos de este, la derivación. También se incluye un problema dónde se aplicará lo explicado en el manual. Por último se presentan las secciones de conclusión, bibliografía y apéndice.

2 Manual

En esta sección se presenta el manual sobre como derivar utilizando WxMaxima. Primeramente se da una introducción a WxMaxima, como escribir texto explicativo, comando matemáticos, guardar e imprimir. Después se muestra una pequeña introducción a la derivación, también como guardar las salidas en una nueva función, gráfica de derivadas y como obtener valores numéricos.

2.1 Bases

2.1.1 Texto explicativo

Para escribir texto, agregar un título, una sección, una subsección, etc. tiene dos opciones:

1. Haciendo uso del menu:

Cell \longrightarrow Insert [Text/Heading/etc]

Y esto colocará un cuadro de texto o cuadro de encabezado en la línea debajo de donde el cursor se encuentre.

2. Usando Ctrl:

Si escribe cualquier parte de texto, puede hacerlo más rápido con estos accesos directos:

- Ctrl+1 : Comienza un nuevo cuadro de texto
- Ctrl+2 : Agrega un título principal
- Ctrl+3 : Agrega un título de sección
- Ctrl+4 : Agrega un título de subsección
- Ctrl+5 : Agrega un título de subsubsección

2.1.2 Comandos matemáticos

Entrada y salida de información:

Cuando se inicia una sesión, aparecerá la etiqueta "% i1", que significa entrada 1 (input 1). Se debe escribir un comando válido al lado de esa etiqueta, que termina con un punto y coma y cuando se pulsa la tecla Intro, esa entrada se analizará, simplificado, vinculado a una variable interna "% i1" y su resultado se mostrará después de una etiqueta "% o1", que significa salida 1 (output2).

(%i1)	2*3;
(%o1)	6
(%i2)	sqrt(2);
(%o2)	$\sqrt{2}$

Para buscar información de una función o variable especial en el manual, por ejemplo, el registro de funciones que se acaba de utilizar, se usa la función de descripción, que se puede abreviar con un signo de interrogación seguido de espacio y el nombre de la función:

(%i4) ? exp;

-- **Function: exp (<x>)**

Represents the exponential function. Instances of 'exp (<x>)' in input are simplified to '%e^<x>'; 'exp' does not appear in simplified expressions.

'demoivre' if 'true' causes '%e^(a + b %i)' to simplify to '%e^(a (cos(b) + %i sin(b)))' if 'b' is free of '%i'. See 'demoivre'.

'%emode', when 'true', causes '%e^(%pi %i x)' to be simplified.

See '%emode'.

'%enumer', when 'true' causes '%e' to be replaced by 2.718... whenever 'numer' is 'true'. See '%enumer'.

Números:

Maxima acepta números reales y complejos. Los números reales en Maxima pueden ser enteros, racionales, o números de coma flotante (incluso notación científica). Los números irracionales se dejan en esa forma, sin ser aproximados por números de coma flotante, para que en cálculos posteriores la salida coincida con el resultado exacto. Sin embargo que lo que se quiere es una aproximación del número, solo basta en realizar las operaciones en ese formato o bien usar la función "float":

(%i2) sqrt(2);

(%o2) $\sqrt{2}$

(%i5) float(%o2);

(%o5) 1.414213562373095

Variables:

Para vincular un objeto a una variable, como un valor numérico, una expresión algebraica o a cualquier otro objeto, se usa el símbolo ":" y no el signo igual "=", que se usará para definir ecuaciones matemáticas. El nombre de las variables puede ser cualquier combinación de letras, números y uno de los símbolos % o _, pero el primer carácter no puede ser un número.

(%i8) a:8\$

(%i9) b:sqrt(a);

(b) $2^{3/2}$

Otras acciones que se pueden llevar a cabo con variables son: eliminar el valor vinculado a esta con la función "remvalue" y para eliminar todos los valores vinculados a varias variables, se utiliza el comando "remvalue (all)". Otra acción

es la de sustituir una variable en una expresión por un valor dado, para lo cual se usa el comando "subst".

Lista:

Una variable también se puede vincular a una lista de valores, como un vector renglón, que se colocan entre corchetes y separados por comas. Por ejemplo, el siguiente comando vincula a "primos" a una lista con los número primos entre el 1 y el 10:

```
(%i2)      primos[2, 3, 5, 7]$
```

A las listas se les pueden realizar muchas de las operaciones que se dan entre los números. También a los elementos de una lista se les numera por números enteros que comienzan por 1. Para hacer referencia a un elemento de la lista, se escribe el nombre de la lista y el índice correspondiente se escribe entre corchetes.

Una función muy útil para crear listas es "makelist", que expande una expresión, reemplazando varios valores diferentes para una variable determinada. El primer argumento dado a "makelist" debe ser la expresión a expandir y el segundo argumento es el nombre de la variable que será reemplazada por una secuencia de valores desde un valor inicial hasta un valor máximo definido por el tercer y cuarto argumentos. Si se proporciona un quinto argumento, se usará como el incremento en la secuencia de valores que se reemplazarán; de lo contrario, se usará el incremento predeterminado de 1. Dos ejemplos del uso de esta función son los siguientes:

```
(%i4)      cuadrados: makelist(m^2, m, 1, 4, 0.5),
(cuadrados) [1, 2.25, 4.0, 6.25, 9.0, 12.25, 16.0]
```

Constantes:

Hay algunas constantes matemáticas predefinidas en Maxima. Los nombres de variable vinculados a esas constantes generalmente comienzan con el símbolo %. Tres constantes importantes son el número π , vinculado a %pi, el número de Euler e , la base de los logaritmos naturales, vinculado a %e y el número imaginario i , vinculado a %i.

```
(%i5)      radio:4 $
```

```
(%i6)      area:%pi*radio^2;
(area)     16  $\pi$ 
```

2.1.3 Guardar trabajo

Para guardar el trabajo realizado se tienen dos opciones:

1. Haciendo uso del menu:

File → Save
File → Save As

2. Usando Ctrl:

Ctrl+s: Save
Ctrl+Shift+s: Save As

2.2 Introducción a la derivada

El comando para calcular la derivada de cualquier expresión en Maxima es **diff** (), la cual tiene se utiliza utilizando la siguiente forma:

diff (expresión, variable, [orden de derivada (solo si es mayor que 1)])

```
(%i7) diff(4·x^3-2·x^2+7·x, x);
(%o7) 12 x^2-4 x +7

(%i8) diff(4·x^3-2·x^2+7·x,x,2);
(%o8) 24 x-4
```

Otra manera de encontrar la derivada de un función, es definiendo esta y aplicando el comando anterior de la siguiente manera:

```
(%i9) f(x):=4·x^3-2·x^2+7·x$
(%i11) diff(f(x),x); diff(f(x),x,2);
(%o10) 12 x^2-4 x +7
(%o11) 24 x-4
```

2.3 Guardando la salida en una nueva función

Hay veces en las que es necesario tener a la mano la derivada de una función, ya sea para saber cuando ésta es cero o simplemente calcular su valor. Para eso es necesario guardar el valor de la derivada en una nueva función, esto se logra de la siguiente manera:

```
(%i12) g(x):=4·sin(sqrt(x^2-4));
(%o12) g(x):=4 sin(√x^2-4)

(%i13) h(x):=subst(t=x,diff(g(t),t));
(%o13) h(x):=subst(t=x, d/dt g(t))

(%i14) h(x);
(%o14) 4 x cos(√x^2-4)
         √x^2-4
```

Es decir, diferenciamos $f(t)$ con respecto a t , y luego reemplazamos a t por x . Esto es para que a momento de querer un valor de la derivada, no haya problema con los argumentos de las funciones. Así si queremos encontrar la derivada evaluada en algún punto:

$$\begin{aligned}
 (\%i14) \quad & h(x); \\
 (\%o14) \quad & \frac{4x \cos(\sqrt{x^2-4})}{\sqrt{x^2-4}} \\
 (\%i15) \quad & h(5); \\
 (\%o15) \quad & \frac{20 \cos(\sqrt{21})}{\sqrt{21}}
 \end{aligned}$$

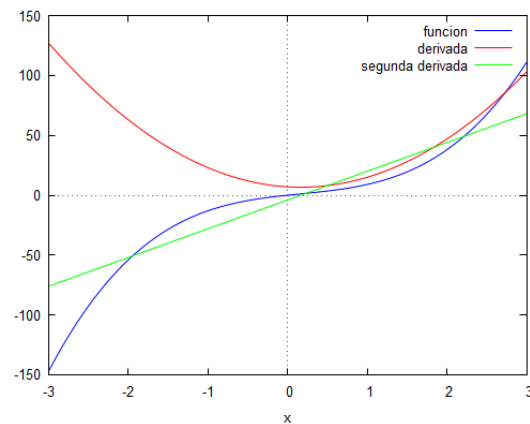
2.4 Graficando derivadas

Ya que se pueden guardar las derivadas en nuevas funciones, se puede hacer con ellas todo lo que se puede hacer con cualquier función, tal como ser graficadas, por ejemplo:

```

(%i25) f(x),f2(x):=subst(t=x,diff(f(t),t)),f3(x):=subst(t=x,diff(f(t),t,2));
(%o23) 4 x^3-2 x^2+7 x
(%o24) f2(x):=subst(t=x,d/dt f(t))
(%o25) f3(x):=subst(t=x,d^2/dt^2 f(t))
(%i26) wxplot2d([f(x),f2(x),f3(x)],x,-3,3],[legend,"funcion","derivada","segunda derivada"]);

```



2.4.1 Racionalizando el proceso de sustitución

Para evitar tener que realizar el procedimiento de diferencias y sustituir varias veces, se puede crear una función que realice todos esos pasos por nosotros, tal

como se muestra a continuación:

```
(%i1) DIFF(f):=subst(t=x,diff(f(t),t));
(%o1) DIFF(f):=subst(t=x,d/dt f(t))
```

Lo que se hizo fue una nueva operación (llamada DIFF ()) que toma una función $f(x)$ y realiza el proceso de diferenciación y sustitución con esta. A continuación un ejemplo de como se utiliza:

```
(%i2) f(x):=(x^3-5)/(x^4+6*x-1);
(%o2) f(x):=x^3-5/x^4+6*x-1
(%i3) g(x):=DIFF(f(x));
(%o3) g(x):=DIFF(f(x))
(%i5) g(x),g(1);
(%o4) 3*x^2/x^4+6*x-1 - (x^3-5)(4*x^3+6)/(x^4+6*x-1)^2
(%o5) 29/18
```

Esta función se puede generalizar para calcular derivadas de orden mayor, solo que será necesario especificar el orden de la derivada:

```
(%i6) NDIFF(f,n):=subst(t=x,diff(f(t),t,n));
(%o6) NDIFF(f,n):=subst(t=x,d^n/dt^n f(t))
```

Con estas nuevas funciones, el proceso de graficar las derivadas se simplifica muchísimo, permitiéndonos hacer ejercicios como el siguiente:

```
(%i10) f(x);
(%o10) x^3-5/x^4+6*x-1
(%i7) h(x):=NDIFF(f(x),2);
(%o7) h(x):=NDIFF(f(x),2)
(%i9) h(x),h(2);
(%o8) 6*x/x^4+6*x-1 - 6*x^2(4*x^3+6)/(x^4+6*x-1)^2 - 12*x^2(x^3-5)/(x^4+6*x-1)^2 + 2(x^3-5)(4*x^3+6)^2/(x^4+6*x-1)^3
(%o9) -3700/6561
```

2.4.2 Una forma aún más simplificada de obtener derivados de orden superior

Una forma diferente de obtener derivados de orden superior se basa en la creación de una lista vacía, donde, en el termino 0 se se escribe a función a diferenciar, después, se define el elemento n de la lista como la derivada del elemento n-1;

```

(%i1)  f:={};
(%o1)  f = {}

(%i2)  f[0](x):=x*sin(sqrt(x^3+2));
(%o2)  f_0(x) := x sin(√(x^3+2))

(%i3)  f[n](x):=subst(t=x, diff(f[n-1](t),t));
(%o3)  f_n(x) := subst(t=x, d/dt f_{n-1}(t))

```

Un ejemplo de esto es:

```

(%i10)  f[0](x);f[1](x);f[2](x);
(%o8)  x sin(√(x^3+2))
(%o9)  sin(√(x^3+2)) + (3 x^3 cos(√(x^3+2)) / (2 √(x^3+2)))
(%o10)  - (9 x^5 sin(√(x^3+2)) / (4 (x^3+2))) + (6 x^2 cos(√(x^3+2)) / √(x^3+2)) - (9 x^5 cos(√(x^3+2)) / (4 (x^3+2)^(3/2)))

```

2.5 Obteniendo valores numéricos

Si se desea obtener una aproximación a una derivada, lo único que se debe de hacer es utilizar el comando float a la evaluación de la derivada, tal como se muestra a continuación:

```

(%i1)  f(x):=x*exp(x^2);
(%o1)  f(x) := x exp(x^2)

(%i2)  g(x):=subst(t=x, diff(f(t),t));
(%o2)  g(x) := subst(t=x, d/dt f(t))

(%i3)  g(2);
(%o3)  9 %e^4

(%i4)  float(g(2));
(%o4)  491.3833502982981

```

Sin embargo, si solo va a pedir un valor de la derivada, entonces es más rápido simplemente sustituirlo directamente como un valor flotante, en el siguiente ejemplo se muestra una comparación de un numero exacto y uno flotante:


```
(%i6) f(x):=(x-3)*sin(x^2-6*x-5);
(%o6) f(x):=(x-3)*sin(x^2-6*x-5)

(%i8) subst(t=2,diff(f(t),t)), subst(t=2.0,diff(f(t),t));
(%o7) 2*cos(13)-sin(13)
(%o8) 1.394726526073752
```

O bien, si se va a solicitar una gran cantidad de salidas en modo flotante de una misma función, simplemente se fuerza el comando float () en la definición:

```
(%i10) f(x);
(%o10) (x-3)*sin(x^2-6*x-5)

(%i11) g(x):=float(subst(t=x,diff(f(t),t)));
(%o11) g(x):=float(subst(t=x,d/dt f(t)))

(%i13) g(1);g(2);
(%o12) -6.16855112172225
(%o13) 1.394726526073752
```

3 Problema de Aplicación

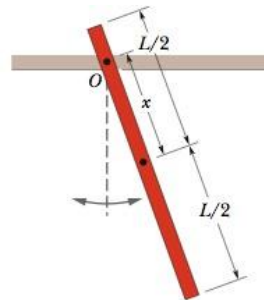
Se forma un péndulo físico por medio de una barra de $L=0.5\text{m}$, de masa $m=0.8\text{ kg}$ pivotada a una distancia $x=0.18\text{ m}$ medida desde su centro de masa. Encontrar:

1. El periodo para pequeñas oscilaciones.
2. La distancia x , tal que el periodo sea mínimo

Del problema tenemos que:

```
(%i15) m:0.8; L:0.5; x:0.18; g:9.81;
(m) 0.8
(L) 0.5
(x) 0.18
(g) 9.81

(%i16) lcm:(1/12)*m*L^2;
(lcm) 0.01666666666666667
```



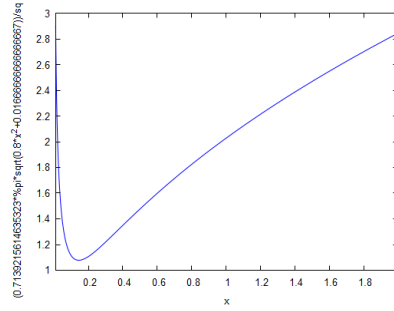
Entonces para el inciso 1, sustituimos los datos en la ecuación para el periodo y obtenemos:

```
(%i17) T:=2*pi*sqrt((lcm+m*x^2)/(m*g*x));
(T) 0.3472570488820396 pi

(%i18) float(T);
(%o18) 1.090940193675087
```

Para el inciso 2, definimos la función del periodo y realizamos su gráfica, para observar su comportamiento:

```
(%i20) T(x):=(2*pi)*sqrt((lcm+m*x^2)/(m*g*x));
(%o20) T(x):=2*pi*sqrt((lcm+m*x^2)/(m*g*x))
(%i26) wxplot2d([T(x)], [x, 0.01, 2]);
```



Se obtienen la primera y segunda derivada de la función:

```
(%i31) d1(x):=subst(t=x, diff(T(t), t));
(%o31) d1(x):=subst(t=x, d/dt T(t))

(%i32) d1(x);
(%o32) (0.5711372491708259 pi*sqrt(x))/sqrt(0.8 x^2 + 0.01666666666666667) - (0.3569607807317661 pi*sqrt(0.8 x^2 + 0.01666666666666667))/x^(3/2)

(%i37) d2(x):=subst(t=x, diff(T(t), t, 2));
(%o37) d2(x):=subst(t=x, d^2/dt^2 T(t))

(%i41) d2(x);
(%o41) (0.5354411710976492 pi*sqrt(0.8 x^2 + 0.01666666666666667))/x^(5/2) - (0.4569097993366607 pi*x^(3/2))/(0.8 x^2 + 0.01666666666666667)^(3/2)
```

Después se encuentra la raíz de la primera derivada y se evalúa en la primera y segunda derivada para verificar que sea mínimo:

```
(%i36) h:=find_root(d1(x), x, 0.01, 1);
(h) 0.1443375672974064
```

```
(%i44)  d1(h); d2(h);
(%o43)  0.0
(%o44)  8.234010608071113  $\pi$ 
```

Cómo la primera derivada es igual a cero y la segunda es un número positivo, entonces con $x=0.144$ m se obtendrá un periodo mínimo.

4 Conclusión

Consideró que ha sido una de las mejores actividades del curso, ya que se ve directamente la aplicación de los CAS a los problemas con los que me enfrenté en mis demás materias. Además realizar yo este manual me permitió entender más a fondo como utilizar WxMaxima, llegando a formar parte de una de las herramientas que seguramente utilizaré durante la carrera.

5 Bibliografía

- A. Maxima Tutorial. (2018). Def.fe.up.pt. Recuperado el 28 de Abril de 2018, de https://def.fe.up.pt/dynamics/maxima_tutorial.html
- wxMaxima Tutorials. (2018). Scotchchildress.com. Recuperado el 28 de Abril de 2018, de <http://www.scotchchildress.com/wxmaxima/>
- Logo de Maxima: <https://commons.wikimedia.org/wiki/File:Maxima-new.svg>

6 Apéndice

1. ¿Cuál fue tu primera impresión de wxmaxima?
Me gustó muchísimo, ya que es muy fácil de utilizar, sobre todo por el lenguaje de los comandos y como expresa el resultado.
2. ¿Crees que esta herramienta puede ser útil en otros de tus cursos?
Sí, por que simplifica mucho el trabajo, además de que puedes comparar si los resultados que obtuviste son los correctos.
3. ¿Qué se te dificultó mas en esta actividad?
Nada, me pareció una actividad muy sencilla.
4. ¿Se te hizo compleja esta actividad? ¿Cómo la mejorarías?
No, así como está me parece muy bien.