



UNIVERSIDAD DE SONORA

División de Ciencias Exactas y Naturales

Licenciatura en Física

Física Computacional I

Actividad 9

"Sistema de Algebra Computacional Maxima"

Michelle Contreras Cossio

Profr. Carlos Lizárraga Celaya

Hermosillo, Sonora

28 de Abril de 2018

1 Introducción

En este reporte se presenta un breve manual sobre el software Maxima, donde se darán a conocer funciones básicas que a criterio propio, serán las más utilizadas.

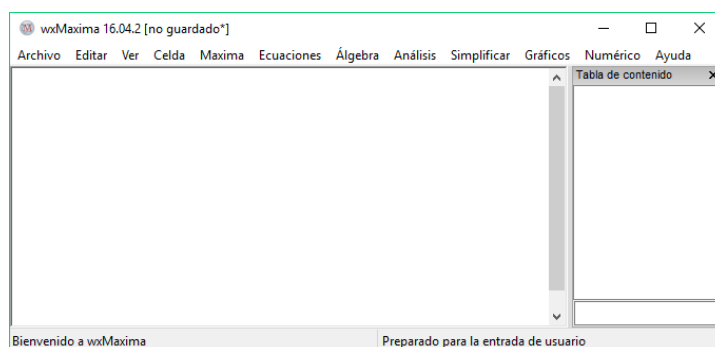


Maxima es un software libre, de los más antiguos CAS (Computer Algebra Systems), fue creado en el MIT en la década de los 60, llamado Macsyma en ese entonces, pero adquirió popularidad hasta mediados de los 70. Este software permite resolver cálculos de diferentes áreas de la matemática, desde aritmética hasta ecuaciones diferenciales o incluso gráficos en 2D y 3D.

Durante este curso estaremos utilizando wxMaxima, que es una versión simplificada o más amigable para un usuario principiante,

2 Infraestructura de Maxima

En esta sección se presentan algunos conceptos y funciones básicas para entender el funcionamiento y poder utilizar Maxima.



2.1 Data I/O

Cuando se inicia una sesión en Maxima aparece " $(\%i1)$ ", que indica la entrada (input) 1, en esa fila se puede escribir un comando y al terminarlo siempre es necesario agregar un punto y coma ";", así se crea una variable interna llamada " $(\%i1)$ ", que puede ser llamada posteriormente. Análogamente, el resultado obtenido será en el siguiente renglón con el tag de " $(\%o1)$ ", refiriéndose a la salida (output) 1.

```

[ (%i1) 2+2;
  (%o1) 4

[ (%i2) %i1+3;
  (%o2) 7

```

2.2 Números, Variables y Constantes

Maxima acepta números reales (enteros, racionales e irracionales) y complejos. Los números racionales se pueden presentar como fracciones o números de punto flotante. Por su lado, los números irracionales como $\sqrt{2}$ o $\log(2)$, se presentan en el output de igual manera (`sqrt(2)`, `log(2)`, respectivamente), es decir, no se aproximan, para evitar errores de redondeo, sin embargo, si se desea una aproximación numérica, se puede utilizar la función **"float"**, que permite convertir a decimal el número ingresado, con una aproximación de 15 a 17 dígitos significativos. Aunque con la función **"fpprec"** y **"bfloat"**, se puede ajustar el número de dígitos significativos que se desean

También, con la función de **"rationalize"**, es posible escribir el número en forma fraccionaria, esto permite evitar errores, ya que finalmente, se trabaja con un sistema de computadora que no permite representar ciertos números y tiende a redondearlos.

Para asignar el valor a una variable, se utiliza el símbolo ":", ya que "=" se utiliza para ecuaciones. Los caracteres aceptados para nombrar variables son letras, números, % o _, con la condición de que el primer caracter no puede ser un número. Los valores asignados a las variables pueden ser números u objetos como expresiones.

```

[ (%i3) a:3;
  (a) 3

[ (%i4) b:2*x+3*y;
  (b) 3 y+2 x

```

Para desasignar su valor a la variable se puede utilizar la función **"remvalue"**, para hacerlo para todas, se puede utilizar el comando `remvalue(all)`. A las variables también se les puede asignar una lista de datos, definiéndola de la misma manera, pero utilizando corchetes, ej: `a:[1,2,3]`.

```

[ (%i5) c:[1,2,3];
  (c) [1,2,3]

```

Por su parte las constantes son valores predefinidos por Maxima, para llamarlos se utiliza el símbolo %. Por ejemplo %pi, %e, %i.

```
(%i6) 3*%pi;
(%o6) 3 π
```

2.3 Aritmética

Los símbolos utilizados para realizar operaciones son:

- Suma: +
- Resta: -
- Multiplicación: *
- División: /
- Exponentes: ^
- Raíz cuadrada: sqrt()
- Raíz enésima: $\hat{1/n}$
- Logaritmo natural: log(a)
- Logaritmo base n de a: $\log(a)/\log(n)$.

2.4 Plots

Maxima también permite graficar en 2D y 3D.

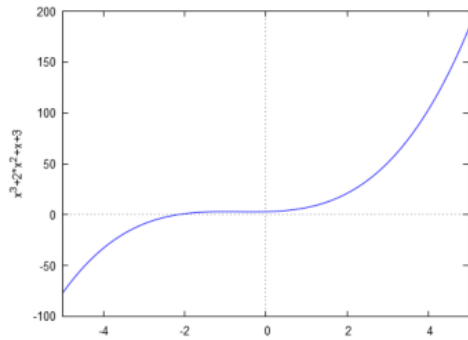
2.4.1 Plot en 2D

Para graficar en el plano (una variable), se utiliza la función **"plot2d"**. Es posible graficar una o varias funciones:

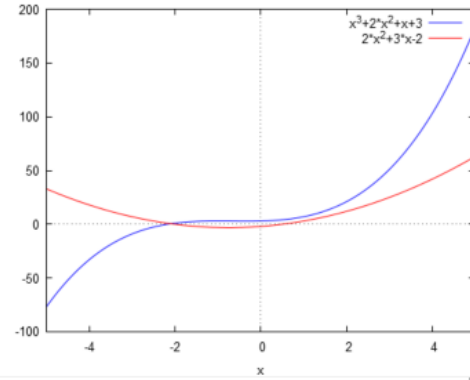
```
plot2d([f1(x), f2(x),...,fn(x)], [x,a, b], [png_file, "archivo.png").
```

El comando anterior permite crear, en una sola imagen, las gráficas de n funciones que dependen de x, donde x irá de a hasta b y se guardará en un archivo png en la carpeta donde se esta corriendo maxima.

```
wxplot2d([x^3+2*x^2+x+3], [x,-5,5])$
```



```
wxplot2d([x^3+2*x^2+x+3, 2*x^2+3*x-2], [x,-5,5])$
```



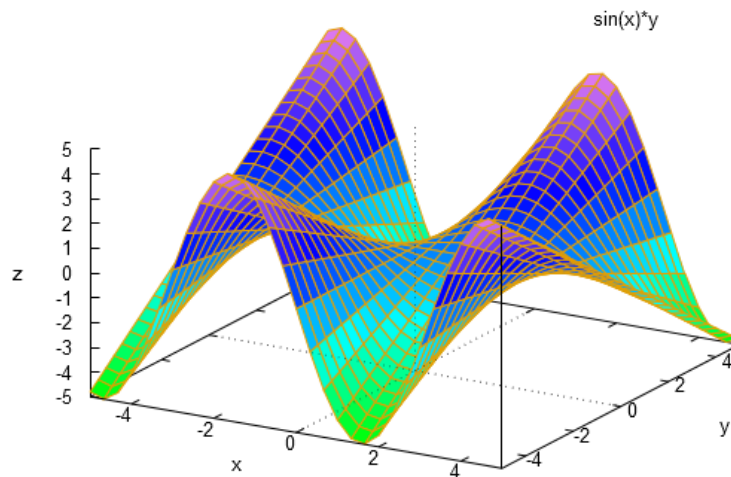
También es posible graficar puntos: `plot2d([discrete, p])`. Donde `p` almacena los puntos a graficar.

2.4.2 Plot en 3D

Por otro lado, también es posible graficar funciones que dependan de dos variables, de manera análoga, se utiliza el comando `plot3d`:

```
plot3d(f(x,y), [x,a,b], [y,c,d])
```

```
wxplot3d(sin(x)*y, [x,-5,5], [y,-5,5])$
```



3 Áreas de matemáticas de interés

En esta sección se presentan posibles comandos o funciones a utilizar en las áreas en las que más trabajo.

3.1 Álgebra

En los renglones de input o output, además de operaciones aritméticas, también se pueden añadir ecuaciones o expresiones algebraicas, utilizando los mismos símbolos mencionados en la sección 2.3.

Para encontrar las raíces de un polinomio se utiliza la función "**allroots**", donde se pone como argumento el polinomio o la variable que corresponde a este.

```
(%i12) f:3*x^4+2*x^3-6*x^2+8*x-7;
(f)      3 x^4+2 x^3-6 x^2+8 x-7

(%i13) allroots(f);
(%o13) [[x=0.9689109596708385 %i+0.3004132341849592, x=
0.3004132341849592-0.9689109596708385 %i, x=1.0, x=-
2.267493135036585 ]]
```

Para resolver una ecuación para cierta incógnita, se utiliza la función "**solve**".

```
(%i14) g:(x^2+3*x+8)^3=0;
(g)      (x^2+3 x+8)^3=0

(%i15) solve(g);
(%o15) [[x=-sqrt(23)%i+3/2, x=sqrt(23)%i-3/2 ]]
```

Para desarrollar una expresión se utiliza la función "**expand**" y viceversa, para factorizarla, se utiliza la función "**factor**".

```
(%i16) expand(g);
(%o16) x^6+9 x^5+51 x^4+171 x^3+408 x^2+576 x+512=0

(%i17) factor(%o16);
(%o17) (x^2+3 x+8)^3=0
```

La función "**subst**" sustituirá valores en la ecuación dada.

```
(%i18)  subst(x=2,f);
(%o18)  49
```

3.2 Cálculo

Para encontrar la derivada de cierta función se utiliza el comando **diff(f,x)**, donde se calcula la derivada de la función f con respecto a x.

```
(%i19)  diff(f,x);
(%o19)  12 x^3+6 x^2-12 x+8
```

De igual manera, para calcular la integral definida de f con respecto a x se utiliza **integrate(f,x)**.

```
(%i20)  integrate(f,x);
(%o20)  3 x^5/5 + x^4/2 -2 x^3+4 x^2-7 x
```

Para calcular la integral definida, se utiliza el mismo comando **integrate(f,x,a,b)**, donde a y b son los límites superior e inferior, respectivamente, de la evaluación

```
(%i21)  integrate(f,x,0,2);
(%o21)  66/5
```

Por otro lado, para realizar una integración numérica, se puede utilizar la función **quad_qags**.

```
(%i22)  quad_qags(f,x,0,2);
(%o22)  [13.2, 2.32568678964188 10^-13, 21, 0]
```

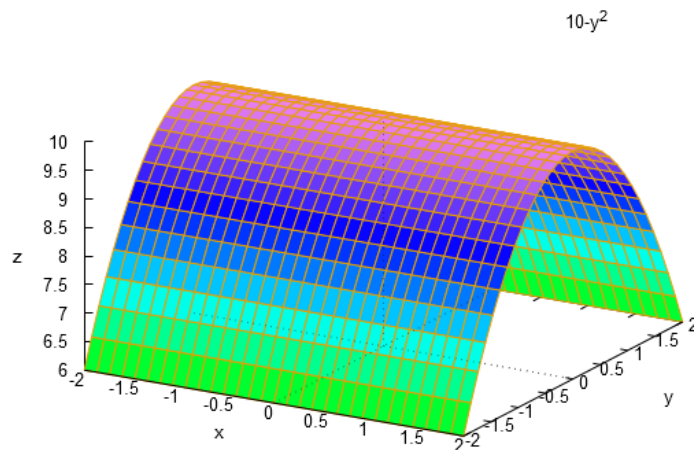
4 Ejemplo

El siguiente ejemplo se tomó del examen parcial 1 de Cálculo IV, del tema Integrales Dobles:

Se pretende usar varios modelos de carpas para una celebración (dos de estas se muestran en la figura de abajo). Determine el volumen que encierra cada una de ellas con respecto al suelo, si el techo esta dado por a) $z + y^2 = 10$ sobre una base cuadrada $[-2,2] \times [-2,2]$, b) $z = 6 + 5e^{(-x^2 - y^2)}$ con sombra $x^2 + y^2 \leq 4$, y compare...

a) Primeramente se grafica la función:

```
f: 10-y^2;  
10-y^2  
  
wxplot3d(f, [x,-2,2], [y,-2,2])$
```



Y posteriormente, se realiza la integración de la función.

```
integrate(integrate(f,x,-2,2),y,-2,2);  

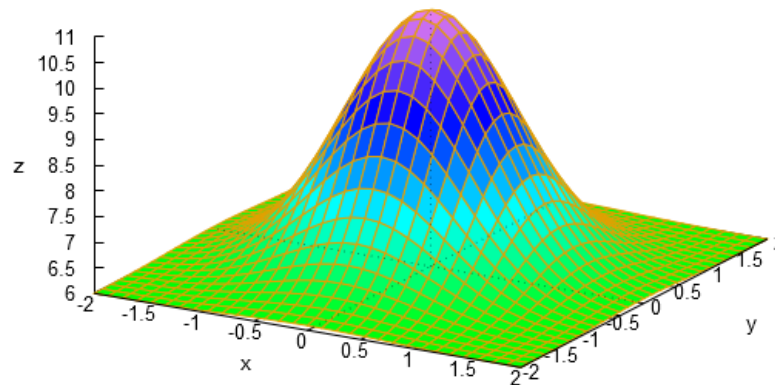
$$\frac{416}{3}$$

```

b) De igual manera, se realiza la gráfica de la función.:


```
g:6+5*%e^(-x^2-y^2);
5 %e-y2-x2+6

wxplot3d(g, [x,-2,2], [y,-2,2])$
```

$$5 \cdot e^{-(y^2-x^2)} + 6$$


Y se integra, pero realizando antes, a mano, un cambio a coordenadas polares:

```
h:6+5*%e^(-r^2);
5 %e-r2+6

integrate(integrate(h*r,r,0,2),r,0, 2*%pi);
2  $\left( \frac{e^{-4} (24 e^4 - 5)}{2} + \frac{5}{2} \right) \pi$ 

float(%);
90.81848557121074
```

Concluyendo que la carpa que abarca más volumen es la del inciso a).

5 Diccionario de funciones básicas

En la siguiente sección se presentan algunas funciones, que se espera sean las más utilizadas:

- `coeff(a,b,c)`: da el coeficiente de b, elevado a la c en la expresión a.
- `cons(a,b)`: añade a, a la lista b, como el primer elemento.
- `denom(a)`: da el denominador de a.

- `desolve(a,b)`: resuelve un sistema a , de ODE, de b ecuaciones usando transformadas de Laplace.
- `determinant(a)`: determinante de a .
- `ident(a)`: regresa la función de identidad $a \times a$.
- `imagpart(a)`: parte imaginaria de a .
- `integrate(a,b)`: integral indefinida de a con respecto a b .
- `invert(a)`: invierte una matriz.
- `kill(a)`: elimina una variable.
- `limit(a,b,c)`: límite de a , cuando b tiende a c .
- `loadfile(a)`: carga un archivo en el directorio en el que se encuentra.
- `matrix(a1,a2,...,an)`: crea una matriz.
- `num(a)`: numerador de a .
- `ode2(a,b,c)`: resuelve EDO de primero o segundo orden.
- `realpart(a)`: parte real de a .
- `taylor(a,b,c,d)`: serie de Taylor.
- `transpose(a)`: matriz transpuesta de a .
- `trigsimp(a)`: simplificación trigonométrica.

6 Conclusiones

En lo particular, no conocía Maxima y tampoco cuento con softwares como Wolfram o Maple, debido al alto costo, suelo utilizar una plataforma online llamada Symbolab e incluso Geogebra, sin embargo, en ocasiones se encuentra bastante limitado, por lo que el hecho de que se me haya presentado este software libre y muy trabajado, ya que no es nuevo y creo poder explotar su potencial, lo poco que conocí, al nivel de conocimiento que hemos adquirido en programación, me parece ser bastante manejable, útil y confiable.

7 Bibliografía

- Maxima 5.41.0 Manual (2017). Consultado: 28 de Abril del 2018, de Maxima Source Forge. Sitio web: http://maxima.sourceforge.net/docs/manual/intromax.html#tth_sEc8
- A partial list of Maxima functions(2010). Consultado: 28 de Abril del 2018, de Maxima Source Forge. Sitio web: http://maxima.sourceforge.net/docs/manual/maxima_singlepage.html#SEC_Top
- Maxima Tutorial (2018). Consultado: 28 de Abril del 2018, de Jaime E. Villate. Sitio web: https://def.fe.up.pt/dynamics/maxima_tutorial.html
- Maxima by Example(2009). Consultado: 28 de Abril del 2018, de Edwin L. (Ted) Woollett. Sitio web: <http://web.csulb.edu/~woollett/>

8 Apéndice

1. ¿Cuál fue tu primera impresión de wxmaxima?

Me pareció bastante digerible, fácil de usar, aunque con interfaz algo anticuada o básica.

2. ¿Crees que esta herramienta puede ser útil en otros de tus cursos?

Sí, ya que, como mencioné en las conclusiones, yo si hago uso de la tecnología para resolver problemas matemáticos que se me presentan.

3. ¿Qué se te dificultó mas en esta actividad?

Realmente nada, ya que fue únicamente leer, resumir y entender, así como resaltar funciones importantes.

4. ¿Se te hizo compleja esta actividad? ¿Cómo la mejorarías?

No, bastante fácil y no la mejoraría porque creo que es necesario empezar a usar un programa con un poco de literatura previa, para poder entender lo básico, incluso recomendaría una actividad así al empezar a trabajar con Python.