Universidad de Sonora

Departamento de ciencias exactas

Física Computacional

Sistema de álgebra computacional Maxima

Alumno:
Martha Anahí Iñiguez Beltrán

214202804

20 de abril del 2018

Contents

1 Introducción

En esta actividad se centra en el uso de una nueva herramienta computacional, como lo es wxmaxima. El sistema de álgebra computacional Maxima es un motor de cálculo simbólico escrito en lenguaje Lisp publicado bajo licencia GNU GPL.

Tiene un amplio conjunto de funciones para manipulación simbólica de polinomios, matrices, funciones racionales, integración, derivación, manejo de gráficos en 2D y 3D, manejo de números de coma flotante grandes, expansión en series de potencias y expansiones de Fourier, entre otras funcionalidades. Además cuenta con un depurador a nivel de fuente para el código de Maxima.

En este trabajo haremos una pequeña síntesis acerca del tutorial de la sección de ecuaciones diferenciales.

1.1 Introducción a las ecuaciones diferenciales

La sección describe las funciones disponibles en Maxima con las cuales se obtienen soluciones análiticas para tipos específicos de ecuaciones diferenciales de primer y segundo orden. Para solucionar sistemas de ecuaciones diferenciales se debe adicionar el paquete dynamics y para las representaciones grafícas es necesario incluir el paquete plotdf.

1.2 Funciones y variables para ecuaciones diferenciales

Function: bc2 (solution, xval1, yval1, xval2, yval2)

Resolver una ecuación diferencial de segundo orden con condiciones iniciales. *solution* is una solución general de la ecuación, encontrada por $ode2$; $xval1$ especifica los valores de la variable independiente en un primer punto, en la forma $x = x1$, y $yval1$ da los valores de la variable dependiente en un punto, de la forma $y = y1$. La expresión $xval2$ y $yval2$ dan los valores de estas variables en un segundo punto, usando la misma forma.

Function: desolve

```
desolve (eqn, x)
desolve ([eqn_1, ..., eqn_n], [x_1, ..., x_n])
```

La función *desolve* resuelve el sistema de ecuaciones diferenciales lineales usando transformadas de Laplace. Aqui *eqns* son ecuaciones diferenciales en la variable

dependiente $x_1, ..., x_n$.

La dependencia funcional $x_1, ..., x_n$ en la variable independiente, para el caso de x, debe ser explícitamente indicada en las variables y las derivadas. Por ejemplo la manera correcta de expresarlo es:

```
eqn_1: 'diff(f(x),x,2) = sin(x) + 'diff(g(x),x);
eqn_2: 'diff(f(x),x) + x^2 - f(x) = 2*'diff(g(x),x,2);
```

y para llamar a la función *desolve* escribimos

```
desolve([eqn_1, eqn_2], [f(x),g(x)]);
```

Si conocemos las condiciones iniciales $x = 0$, estas pueden ser suministrada antes de llamar *desolve* usando *atvalue*, por ejemplo.

```
(%i1) 'diff(f(x),x)='diff(g(x),x)+sin(x);
                d           d
(%o1)           -- (f(x)) = -- (g(x)) + sin(x)
                dx          dx
(%i2) 'diff(g(x),x,2)='diff(f(x),x)-cos(x);
                 2
                d            d
(%o2)           --- (g(x)) = -- (f(x)) - cos(x)
                  2          dx
                dx
(%i3) atvalue('diff(g(x),x),x=0,a);
(%o3)                           a
(%i4) atvalue(f(x),x=0,1);
(%o4)                           1
(%i5) desolve([%o1,%o2],[f(x),g(x)]);
                  x
(%o5) [f(x) = a %e  - a + 1, g(x) =

                                                    x
                                    cos(x) + a %e  - a + g(0) - 1]
(%i6) [%o1,%o2],%o5,diff;
             x       x      x                 x
(%o6)   [a %e  = a %e , a %e  - cos(x) = a %e  - cos(x)]
```

Si *desolve* no puede obtener una solución , este devuelve un $false$ como respuesta.

Function: ic1 (solution, xval, yval)

Resolver una ecuación diferencial de primer orden, en esta caso *solution* es la solución

general, como es encontrada por $ode2, xval$ dando las condiciones iniciales para la variable independiente en la forma $x = x0$, y $yval$ da las condiciones iniciales para la variable dependiente en la forma $y = y0$.

Function: ic2 (solution, xval, yval, dval)

Resolver un problema con condiciones iniciales de una ecuación diferencial de segundo orden, en este caso *solution* indica la solución general de la ecuación encontrada por $ode2$, $xval$ da el valor inicial de la variable independiente en la forma $x = x0$, $yval$ da el valor inicial de la variable dependiente en la forma $y = y0$, y $dval$ da los valores iniciales para la primer derivada de la variable dependiente con respecto a la variable independiente, en la forma $diff(y, x) = dy0$(diff hace que no tenga que ser citado)

Function: ode2 (eqn, dvar, ivar)

La función $ode2$ resuelve una ecuación diferencial ordinaria de primer y segundo orden. Esta función toma los tres argumentos: una ODE da eqn, la variable dependiente $dvar$, y la variable independiente $ivar$. Cuando el calculo tiene exito este da como resultado en la salida una solución explíta o implicita de la variable dependiente. % es usado para representar la constante de integracón en el caso de una EDO de primer orden mientas que %k1 y %k2 se usa para denotar las constantes en el caso de segundo orden. La dependencia de la variable dependiente y la independiente no tiene que ser escrita explicítamente.

Si $ode2$ no puede obtener una solución esta devuelve un $false$ despues de imprimir un mensaje de error. El metódo implementado para la dcuación de primer orden es probar si es: lineal, separable, exacta, homogénea, ecuación de Bernoulli y un metódo homogéneo generalizado, mientras que para las ecuaciones de segundo orden pueden ser resueltas por: coeficientes constantes, exactas, homogéneas lineales con coeficientes no constantes que pueden ser transformada a coeficientes constantes, ecuación de Euler, variación de parámetros entre otras.

Varias variables se establecen solo con fines informativos:$method$, denota el metódo de soluciones usado, $infractor$ denota cualquier factor de integración usado, $odeindex$ denota el indíce para el metódo de Bernoulli para el metódo homogéneo generalizado y yp denota la solución particular cuando se utiliza variación de parámetros.

Para resolver problemas de valores iniciales (IVP) las funciones $ic1$ y $ic2$ están disponibles para ecuaciones de primer y segundo orden, y para resolver problemas de valores límite de segundo orden (BVP) se puede usar la función $bc2$.Ejemplo de la utilización de estos comandos:

```
(%i1) x^2*'diff(y,x) + 3*y*x = sin(x)/x;
```

```
                      2 dy           sin(x)
(%o1)                x  -- + 3 x y = ------
                        dx             x
(%i2) ode2(%,y,x);
                             %c - cos(x)
(%o2)                     y = -----------
                                   3
                                  x
(%i3) ic1(%o2,x=%pi,y=0);
                               cos(x) + 1
(%o3)                    y = - ----------
                                    3
                                   x
(%i4) 'diff(y,x,2) + y*'diff(y,x)^3 = 0;
                          2
                         d y      dy 3
(%o4)                    --- + y (--)  = 0
                           2      dx
                         dx
(%i5) ode2(%,y,x);
                      3
                     y  + 6 %k1 y
(%o5)                ------------ = x + %k2
                           6
(%i6) ratsimp(ic2(%o5,x=0,y=0,'diff(y,x)=2));
                             3
                           2 y  - 3 y
(%o6)                      - ---------- = x
                                6
(%i7) bc2(%o5,x=0,y=1,x=1,y=3);
                          3
                         y  - 10 y       3
(%o7)                    --------- = x - -
                             6           2
```

2 Conclusión

En ésta actividad pudimos explorar una herramienta muy útil que da un ambiente de trabajo amplio. La sintáxis no es muy compleja y tiene distintas aplicaciones

matemáticas. Añadiendo tambien que wxmaxima es un free software lo que la vuelve gratuita y sin nececidad de comprar licencia para su uso completo como Wolfram Mathematica y otros softwares para uso matemático.0

3 Apendíce

¿Cuál fue tu primera impresión de wxmaxima?

Me parece una buena opción y muy completa para ser un software libre

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

Si, para el desarrollo de actividades o tareas.

¿Qué se te dificultó mas en esta actividad?

Comprender de primera la sintaxis de Maxima.

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

Fue sencilla aunque me hubiera gustado tener más referencias o videos un poco más completos.