Fausto Fabian Crespo Fernandez

Deber Seminario

Resolución EDO usando series y cálculo simbólico

Debido a que no tengo el software Mathematica investigue como hacer integración simbólica de EDO con Python mediante la librería Simpy.

Por ejemplo podemos usar

from sympy import *

dsolve(f(x).diff(x, x) + f(x), f(x), hint='2nd_power_series_ordinary')

para resolver la ecuación $y'' + y = 0$ lo que da :

f(x) == C1*x*(-x**2/6 + 1) + C0*(x**4/24 - x**2/2 + 1) + O(x**6)

Mientras que

dsolve(f(x).diff(x, x) + f(x), f(x)) da

f(x) == C1*sin(x) + C2*cos(x)

Con lo que vemos el desarrollo en serie del sen(x) y cos(x)

La lista completa de los hint del comando dsolve es:

`sympy.solvers.ode.`**`allhints`** = *('separable', '1st_exact', '1st_linear', 'Bernoulli', 'Riccati_special_minus2', '1st_homogeneous_coeff_best', '1st_homogeneous_coeff_subs_indep_div_dep', '1st_homogeneous_coeff_subs_dep_div_indep', 'almost_linear', 'linear_coefficients', 'separable_reduced', '1st_power_series', 'lie_group', 'nth_linear_constant_coeff_homogeneous', 'nth_linear_euler_eq_homogeneous', 'nth_linear_constant_coeff_undetermined_coefficients', 'nth_linear_euler_eq_nonhomogeneous_undetermined_coefficients', 'nth_linear_constant_coeff_variation_of_parameters', 'nth_linear_euler_eq_nonhomogeneous_variation_of_parameters', 'Liouville', '2nd_power_series_ordinary', '2nd_power_series_regular', 'separable_Integral', '1st_exact_Integral', '1st_linear_Integral', 'Bernoulli_Integral', '1st_homogeneous_coeff_subs_indep_div_dep_Integral', '1st_homogeneous_coeff_subs_dep_div_indep_Integral', 'almost_linear_Integral', 'linear_coefficients_Integral', 'separable_reduced_Integral', 'nth_linear_constant_coeff_variation_of_parameters_Integral', 'nth_linear_euler_eq_nonhomogeneous_variation_of_parameters_Integral', 'Liouville_Integral')*

De los cuales hay 3 relacionados con series de potencias: *'1st_power_series', '2nd_power_series_ordinary', '2nd_power_series_regular'.*

Resolviendo algunos ejemplos de ecuaciones diferenciales por el método de series del capitulo 6 del libro Ecuaciones diferenciales de Dennis Zill:

17 de 6.1 $y'-x^2y=0$

dsolve(f(x).diff(x) -x*x* f(x), f(x), hint='1st_power_series')

f(x) == C0 + C0*x**3/3 + O(x**6)

22 de 6.1 $y''-y=0$

dsolve(f(x).diff(x, x) - f(x), f(x), hint='2nd_power_series_ordinary')

f(x) == C1*x*(x**2/6 + 1) + C0*(x**4/24 + x**2/2 + 1) + O(x**6)

23 de 6.1 $y''-y'=0$

dsolve(f(x).diff(x, x) - f(x).diff(x), f(x), hint='2nd_power_series_ordinary')

f(x) == C1*x + C0*(x**3/6 + x**2/2 + x + 1) + O(x**6)

El método de Frobenius se puede usar con la opción hint='*2nd_power_series_regular'*.Por ejemplo

Para el prob 22 de 6.2

dsolve(x*x*f(x).diff(x, x) +x*f(x).diff(x)+(x*x-4/9) *f(x), f(x), hint='2nd_power_series_regular')

f(x) == C0*(x**4/64 - x**2/4 + 1) + O(x**6)