Sagemath Quiz Pointers

Jia Cheng

November 2020

Quiz Instructions

1. The quiz covers all materials in SageMath Notes Lesson 1-5.
2. There are 8 sections, and the total mark is 8.
3. The LumiNUS quiz system will select one question from each section to generate a paper for you.
4. All questions are fill-in-blank questions, and all answers are positive numbers.
5. **If the answer is an integer, fill in that integer.** For example, the answer of 1+1 is 2. Note that 2.0 is **wrong**!
6. **If the answer is not an integer, correct to 5 decimal places.** For example, if the answer is the Euler number e = 2.718281828..., then only numerical answer between 2.71828 and 2.71829 are acceptable.
7. If the answer is 0.5, since it is not an integer, how can I key in the answer? The answers like 0.5, 0.50, 0.500, 0.5000, 0.50000 are all acceptable.

Pointers

- Suggested variables
  - summation index i
  - summation limits m, n
  - derivative limit h
  - eval function at values x=a, y=b
  - function f(x), first derivative g(x), second derivative h(x)
- Remember to use var(”y”) when creating expressions that use y as a variable
- Plotting functions
  - Plotting 2 single variable functions
    ```
    plot((f(x), g(x)), (x, left_limit, right_limit), ymin=..., ymax=..., plot_points=...,
    color=("red", "violet"))
    ```
  - Plotting 2 implicit functions
    ```
    A = implicit_plot(f(x, y), (x, left_limit, right_limit), (y, left_limit, right_limit))
    ```

```
B = implicit_plot(g(x, y), (x, left_limit, right_limit), (y, left_limit, right_limit))
A+B
```

- `f(x).derivative(x)`
  - This is the variable of differentiation.
- `f(x, y).implicit_derivative(y, x)`
  - Note that ...(y, x) is for finding dy/dx. So if we want to find dx/dy, then use ...(x, y) as the params.
- The right way to use implicit_derivative
  - Suppose $f(x, y) = x^2 + y^2 == 1, g(x, y) = x^2 + y^2 - 1$
  - f(x, y).implicit_derivative will cause errors. Instead, do g(x, y).implicit_derivative
  - In other words, implicit_derivative accepts functions instead of equations
- Integrals
  - Indefinite f(x).integral(x)
  - Definite f(x).integral(x, a, b)
  - Algorithms: Default(Maxima), `sympy`, `mathematica_free`, `giac`
    `f(x).integral(x, a, b, algorithm="name of algorithm")`
    sympy is recommended
  - When differentiating an integral, use hold=True
    As an example, `cos(t^2).integral(t, cos(x), 5*x, hold=True).derivative(x).show()`
- Generic function
  - To declare f as a function in variable t,
    ```
    var("t")
    function("f")(t)
    ```
- For complicated expressions
  - `.factor()`
  - `.full_simplify()`
  - `.expand()`
- Differential Equations

  -
    ```
    function("y")(x)
    desolve(y(x).derivative(x) == x+y(x), y(x))
    ```
  - With initial conditions
    ```
    ode = y(x).derivative(x) == x+y(x)
    desolve(ode, y(x), ics=[1,2])
    ```
    In general, `ics=[x,y,dy/dx]`
- Evaluation functions
  - `.find_root(left_limit, right_limit)`
    Example: `(log(x) == sin(x)).find_root(1, 3)`

  - `.n(digits=...)`
    For numerical approximation

  - `.solve(x)`
    Example: `(g(x)==x).solve(x, algorithm="sympy")`

- Other helper functions

  - `.is_prime()`

  - `.is_real()`