Computing for mathematics handout 7 - The class test, srange, tangents and more applications of functions or classes to data.

Lecturer: Vince Knight

Office: M1.30

email: knightva@cf.ac.uk

Office hours: Thursday 1300-1500

What you have learnt this week:

- How to plot in Sage (domain of plot and ‘addition’ of plots);
- How to obtain limits in Sage;
- How to differentiate in Sage;
- How to integrate in Sage;
- How to to upload data to Sage.

Obtaining the values from a solution of an equation

Let’s consider the following equation:

$$x^2 - x - 1$$

To find the roots of the equation we can simply use the solve function:

```
sols = solve(x ^ 2 - x - 1 == 0, x)
```

Before viewing the solutions of our equation what type of object is sols?

```
type(sols)
```

sols is a ‘generic sequence’ (a type of list).

```
sols
```

We see that our solutions are given in the form of a list of relationships. The solutions to exercise 3 show a way of extracting the solutions. Here is another:

```
sols = solve(x ^ 2 - x - 1 == 0, x, solution_dict=True)
```

Sols is now a list of dictionaries. Let us try and extract the positive solution to our equation:

```
phi = [k[x] for k in sols if k[x] >= 0][0]
```

If you’re not familiar with what ϕ is try the following:

```
for n in range(1):
    print expand((phi ^ n - (1 - phi) ^ n) / sqrt(5))
```

srange

The srange command is a Sage ‘wrapper’ for the Python range command. It allows us to obtain lists of non integer values with control of step size, start value and end value.

```
# General syntax
srange(startingvalue, endvalue, stepsize)
```

The following gives the numbers from 0 to 4 (not inclusive) with steps of .5:

```
srange(0, 5, .5)
```

Question 9

Question 9 was a tricky task. The solution file shows a function that takes a function and a point and outputs a plot. If the solution is not clear: come and speak to me.

Importing data

Let's carry out the following exercise:

1. Use Python to obtain a list of the Fibonacci numbers;
2. Write those numbers to file;
3. Import that data file in to Sage;
4. Plot the ratio of the differences between two consecutive Fibonacci numbers.
5. Here's the python script:

```
import csv  # Use the csv library

def fib(n):
    """
    A function that returns the nth Fibonacci number.

    Arguments: n (an integer)

    Outputs: The nth Fibonacci number (an integer)
    """
    if n == 0:
        return 0
    if n == 1:
        return 1
    return fib(n-1) + fib(n-2)

f = open('fibonaccinumber.csv', 'w')  # Open a file in write mode
csvwrtr = csv.writer(f)  # Create a writer object (see exercise 10 of sheet 2)
for n in range(31):  # Loop n over the first 30 integers
    csvwrtr.writerow([fib(n)])  # Write the nth Fibonacci number

f.close()  # Close the file
```

Now let us import that file in to Sage and use the following code to obtain the ratios of two successive numbers:

```
import csv  # Use the csv library

f = open(DATA + 'fibs', 'r')  # Open the newly loaded file in Sage.
csvrdr = csv.reader(f)  # Create a reader object

data = [float(row[0]) for row in csvrdr]  # Read in the data and convert to float
f.close()  # Close the file

ratios = []  # Create a new list
for k in range(1, len(data) - 1):  # Iterate over integers
    ratios = [[k, data[k + 1] / data[k]]]  # Add a tuple with the ratio of two consecutive numb

list_plot(k) # A list plot
```

We could do all of the above using Sage but this is just an example of using data written to file.

What you should do next:

- **Start the next sheet**: make sure you spend time working on the sheet **BEFORE** the labs.
- Contribute to the wiki.
- To make the best use of the lab sessions turn up having finished your sheets;
- If anything is still unclear **please** come and see me during office hours.