Math242 Lab 08: Series

Lab Instructions

Complete the following lab examples and exercises.
Make sure each question is clearly labeled and all questions are answered completely.
Submit both the Mathematica .nb file and a .pdf file with the titles

lastnameLab08.nb and lastnameLab08.pdf

on Canvas before the deadline. Late submissions will not be accepted.

Introduction

Determining if a sum

$$\sum_{n=0}^{\infty} a_n$$

converges can be re-phrased as determining if the sequence of partial sums

$$\left\{ \sum_{n=0}^{0} a_n, \sum_{n=0}^{1} a_n, \ldots, \sum_{n=0}^{S_N} a_n, \ldots \right\}$$

converges to a limit. Consider, for example, the infinite series

$$S = \sum_{n=1}^{\infty} \frac{1}{n} = \lim_{k \to \infty} \sum_{n=1}^{k} \frac{1}{n}.$$

We can construct a table of values for the first 101 partial sums of S and see if they seem to be approaching a limit. Instead of viewing the whole table, we will only view every 10th entry, as we did with the lab on sequences. Below we will make a function `s(n)` that computes the n^{th} partial sum.

```
def s(n):
        # do a partial sum of the first n values of
        # \sum 1 / k
        k_values = list(range(1, n))
        partial_sum = sum([1 / k for k in k_values])
        return partial_sum

def print_table_sum(n_values, sum_values, step):
        # helper function which prints a pretty table
        print("n \t sum to n")
        print("----------------------")
        for index in range(0, len(n_values), step):
                print(n_values[index], "\t", sum_values[index])
```

Another way to see if a sequence may be converging or diverging is to plot the sequence of partial sums using matplotlib's `plot`. We can do this with the first 101 terms of the above sequence:

```
 only put imports at the beggining of a python file!
import matplotlib.pyplot as plt

# plot
plt.plot(n_values, sum_values, ".")

# pretty it up
plt.xlabel("n")
plt.ylabel("s(n)")

# display the plot
plt.show()
```

Does this plot look like a familiar function? Based on the plot, does the sequence of partial sums appear to be converging or diverging?
Core Python does not understand symbolic math, but the SymPy package gives it many of the same capabilities as Mathematica.

```
 Again, only have imports at the top of a python file
# this imports the symbolic k, which can be used as a variable
from sympy.abc import k
# import summations, and oo is infinity
from sympy import Sum, oo

a_k = 1 / k
# create an expression for the sum, notice the capital "S" in "Sum"
sum_expression = Sum(a_k, (k, 1, oo))
# doit() actually performs the operation through computation
print(sum_expression.doit())
```

which gives us "oo," indicating the series goes to infinity, and therefore does not converge.
To see everything put together, see the python file `template.py`. To install packages in the Thonny IDE, see https://www.youtube.com/watch?v=Oo-B98WWre_8.

Lab Questions

Questions 1, 2, and 3 can all be run in one Python file. However, it is ultimately up to you how you would like to organize your work.

General Debugging Tip in Python

If your computer is taking too long and you want to cancel a calculation, simply push "Ctrl + C." In Python this terminates a command which is taking too long.

Question 1

For each of the following series, perform the steps outlined below. An example can be seen in either `template.py` or `generalized_template.py`. It is fine if all three tes cases are in the same Python file.

$$\sum_{k=1}^{\infty} \frac{1}{k^2}, \qquad \sum_{k=1}^{\infty} \left(\frac{\ln(k)}{k}\right)^2, \qquad \sum_{k=0}^{\infty} \frac{1}{k!}$$

i) Define a sequence of partial sums.

ii) Use the function `print_table` to print an abbridged table of values.

iii) Plot the partial sums.

iv) Use SymPy to evaluate the exact sum of the series. Does each series converge or diverge? (Put the answer in either a comment or a print statement, e.g. `print("This series converges to e")`).

The Journey of an Infinite Sum Begins with a Single Index . . .

Make sure you are careful with the starting value of the last series–the sum starts at zero instead of one.

Question 2

Let us look more carefully at the last series in Question (1). You should have found that the series converged to $e \approx 2.718$. This is a well-known series we will revisit later in the semester.

i) Modify the code in `sum_errors.py`. It helps you to compute an error list for a partial sum against a known exact value for the infinite sum.

ii) Use the command `plt.loglog` to plot the errors on a $\log - \log$ plot. What do you notice?

iii) Does evaluating the partial sums seem to be a good way to find an approximation to e? Again, put your answer in either a comment or print statement.

Question 3

i) Repeat Question 2 with the series

$$\sum_{k=1}^{\infty} \frac{1}{k^2}.$$

The sum of this series is known to be $\frac{\pi^2}{6}$, so this should be used as your value for `exact`.

It actually takes a lot of effort to prove the series converges to this value (you can read about the Basel Problem on Wikipedia if you are interested—and it illustrates why you are only asked to determine convergence or divergence for most series, not their limits).

ii) Does this series converge as quickly as the one in Question 2? Make sure your sum starts at $k = 1$ instead of $k = 0$.

Question 4

For this question we will build a new python file/module.
We wish to sum the following series:

$$\sum_{n=2}^{\infty} \frac{1}{\ln(n) \cdot 2^n}.$$

Since it is not possible to do by hand, we will instead approximate the value of the series.

i) For large enough S_N, each term in the series can be compared to a geometric series.

$$\sum_{n=S_N}^{\infty} \frac{1}{\ln(n) \cdot 2^n} \leq \sum_{n=S_N}^{\infty} r^n = F(S_N).$$

So determine the appropriate r and $F(S_N)$.

ii) Given a value ϵ for the tolerance (allowed error) we wish to determine the series value to within the allowable error. This will be done by

$$\begin{aligned}
\sum_{n=2}^{\infty} \frac{1}{\ln(n) \cdot 2^n} &= \sum_{n=2}^{S_N-1} \frac{1}{\ln(n) \cdot 2^n} + \sum_{n=S_N}^{\infty} \frac{1}{\ln(n) \cdot 2^n} \\
&\leq \sum_{n=2}^{S_N-1} \frac{1}{\ln(n) \cdot 2^n} + \sum_{n=S_N}^{\infty} r^n \\
&= \sum_{n=2}^{S_N-1} \frac{1}{\ln(n) \cdot 2^n} + F(S_N) \\
&= \sum_{n=2}^{S_N-1} \frac{1}{\ln(n) \cdot 2^n} + \epsilon, \quad \text{provided} \quad F(S_N) \leq \epsilon
\end{aligned}.$$

So given ϵ, solve for S_N, and record this expression, we will use it in the next part.

iii) Write a function which can take a tolerance, ϵ, and returns the approximate value of the infinte series by summing the first S_N terms. Your function should use the given ϵ and compute the appropriate S_N. It should then evaluate and return the value of $\sum_{n=2}^{S_N} \frac{1}{\ln(n) \cdot 2^n}$.

To verify your answer, you can either use Mathematica or Sympy to sum the entire series, and compare your answers.