EVR-5086 Assignment 1 - Calculus Review

Adyan Rios

2025-09-09

Table of contents

Introduction

I learned that there are various ways that I can create and execute Python chunks in R Studio (Velásquez 2021). Although EVR-5086 class is being taught using python, I have advanced skills in R that I hope will be complementary to the course. I am also fond of sharing science on GitHub. I have learned how GitHub pages combined with Quarto and R Studio are an extraordinary resource for developing and maintaining lab notebooks. In order to get better at using these tools and the reproducibility and accessibility of my future research, I have created a html quarto book and pdf to show my work associated with the course assignments.

Set Up

I started by creating a GitHub account (arios101-fiu) and a GitHub repository with a gitignore and readme.md (EVR-5086-Assignment1). I cloned the repository into R Studio, thereby creating a R project. I copied in a _quarto.yml and index file from another project. I simplified the index file and inserted a reference to create a new reference.bib. I updated the yml, rendered, committed and pushed. Next, I turned on GitHub pages and updated the URLs in the yml and repository. On to the assignment…

1 Polynomial Plot

Observations and the limitations of the numerical derivative. One method is an instantaneous change (associated with he tangent line. The other is an average rate of change between two points.

```
# Define variables
a <- 1
n <- 1
b <- 1
p <- 2
c <- 1
q <- 3

# Create x vectors from -1 to 1
x <- seq(from = -1, to = 1, by = 0.1)

# Calculate the value of y for each value of x
y <- (a * (x^n)) + (b * (x^p)) + (c * (x^q))

# Calculate the analytical derivatives for each value of x
dy_dx <- (a * n * (x^(n - 1))) + (b * p * (x^(p - 1))) + (c * q * (x^(q-1)))

# Calculate the numerical derivatives
deltay <- diff(y)
deltax <- diff(x)
deltay_deltax <- deltay / deltax

# For plotting purposes, derive the midpoint
deltax_vec <- x[-length(x)] + deltax/2

# Build dataframe for plot

# Plot the analytically derivative as a solid line and the numerical derivative as open symb
```

2 Solve the 2-D Laplace in Excel

I created a 29 by 29 grid of the 2-D Laplace Equation. I included three internal “boundary values”; one high value of 4 and two low values of -2 and -3. I allowed excel to iteratively calculate with 10,000 iterations and a minimum change of 0.0001. Saving as a CSV file, surrounded by explicit zeros, the dimensions of my data were 30 by 30. Additionally, I rounded to four significant digits to see if it would mean that the stagnation areas would be more pronounced.

3 Read in and plot contours

Plot streamlines

Velásquez, Isabella. 2021. "Posit." https://www.posit.co/.