**Octave Line Integral Project \*\*draft\*\***

*Line Integrals, Green’s Theorem, and Octave Computing*

Summary: In typical homework problems, expressions usually end up “nice” and can be solved by hand. But in this project, you will create a realistically tedious math problem and then use Octave to solve it numerically in 2 different ways.

Step 1: Set up a Domain and

Create a simply connected region R that is either vertically simple or horizontally simple, but not both. It can be bounded by 1, 2, or 3 or smooth curves. Sketch it here in pencil and get Mr. Malan’s approval before proceeding. You will also need to know the vector valued functions for the boundary. Feel free to use Desmos to get started. It is okay if they “almost” connect (decimal values can be off by 0.01).

Region VVF(s) for Boundary

Next you will have to create a 2D Vector Field . Rules:

* It should be a little bit “gross” to evaluate the final integral. The whole point of this project is to use Octave to do a problem that we cannot easily do on our own.
* In the interest of making it “gross,” it must involve at least one of the following:
  + an exponential function
  + a trig function
  + a reciprocal function
  + a log function
* It should be non-conservative.

My Vector Field 2d Curl

Step 2: Set up the Math by Hand

Octave will ultimately to the final calculations for you, but you will need to set it up and get everything to the point of just regular scalar integrals. Look at the following steps in a simple line integral

1. 🡨 Least effort to type into Octave ☺

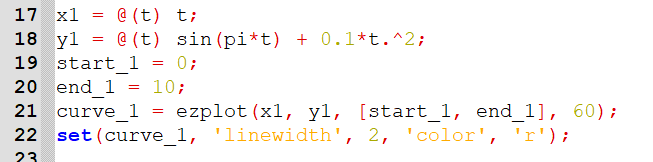
Octave’s **integral** command can only understand #3 and #4, so you must work to get everything into a form like one of those[[1]](#footnote-1).

**Setup for Direct Line Integral (no GT):**

You will need one integral for each of your boundaries from Step 1

**Setup for Using Green’s Theorem:**

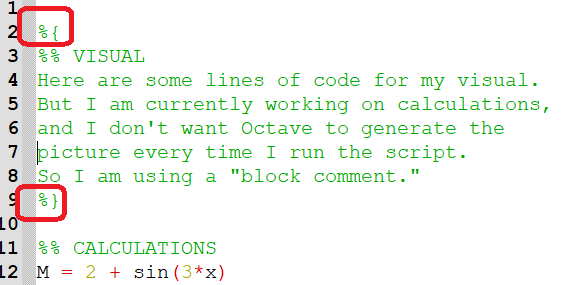
Step 3 Octave Visual

Start a script in Octave

First, use **ezplot** to draw each curve. This will help you make sure that your parametrizations were correct. Here is some of my code, in case it helps you remember how all the commands work.

Then use **hold on** and draw your next field.

Then use **quiver** to draw your field on top of this. See the Chapter 15 Octave instructions for help.

Step 4: Calculate Line Integral Directly

In the **same** script, you will now do 2 calculations.

(Tip: use a “block comment” when you want to “turn off” one part of your script when working on another part.   
Use **%{** and then **%}** )

Once you have written your line integral(s) in the correct form, you will then use the **integral** command. Here is a reminder of how it works:

can be evaluated with the following code:

**integrand\_1 = @(t) 3\*t - t.^2;**

**integral(integrand\_1, 4, 8)**

In order to make your output readable, consider using the printf command

**printf('By directly evaluating the line integral: \n')**

**integral(……**

Step 5: Green’s Theorem

In the **same** script, you will now use Green’s theorem to set up and find the answer using a double integral.

Step 6: Submitting

On the due date, you will turn in this paper and you will upload a single .m file to Canvas.

Rubric:

|  |  |  |  |
| --- | --- | --- | --- |
| **Components** | | **Earned** | **Possible** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| Other | Readable Code (comments, line length, etc) |  | 1 |
| Deductions |  | 0 |
|  |  |  |  |
|  | **Total** |  | **30** |

Idea for future years (or for any 21-22 students who are reading this ☺):

The line integral is potentially really awful.

Use a copy-paste method. Type out something like the following:

r1 i-component: 4t-sin(t)

r1 j component: 3-2t

r1primt i-component: 4-cos(t)

r1prime j-component: -2

Integrand = @(t)

in this anonymous function, set it up with all the red and green parenthesis from the mixed integral above. Separate each much with a “.”

Then paste in.

Another tip: A function can be defined on many rows:

g = @(t) = 17\*t.^2 +…

3\*sin(t);

1. Actually, I am certain that there is a way to put functions into a dot product and then, but I have not yet figured it out. [↑](#footnote-ref-1)