Computational Project Final

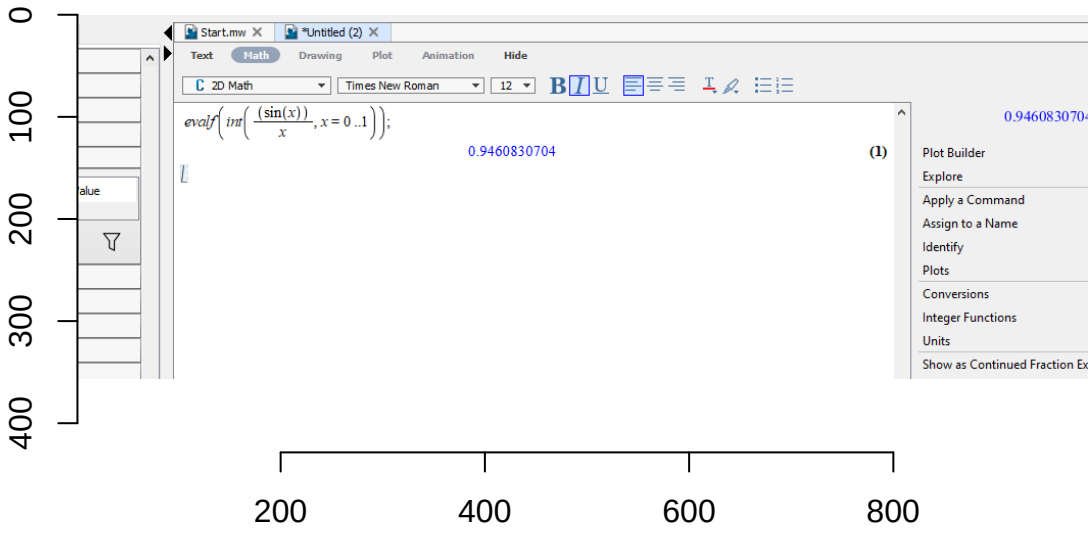
Jefferson Ong

December 08, 2019 @ 02:19 PM

Problem Statement

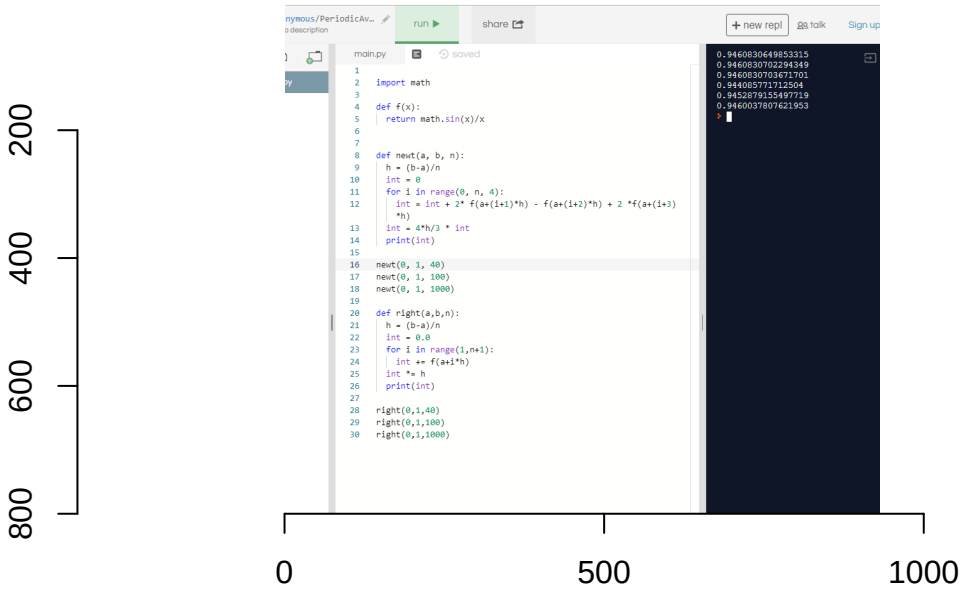
We are estimating the integral of a particular function, in this case $sin(x)/x$. We cannot integrate directly but can come up with a numerical solution using taylor series functions. This function also has additional problems with endpoints, particularly at zero. So we are going to use a specific taylor function called the "open newton-Cotes" in python to approximate $sin(x)/x$. We will compare this to the true solution in maple and see how it changes as we increase in n iterations. Finding the relative error and comparing to the right sum method for similar n

Maple Solution



As we can see here, maple uses its own function to solve the integral and gives us the "true" solution of .9460830704

Python Code

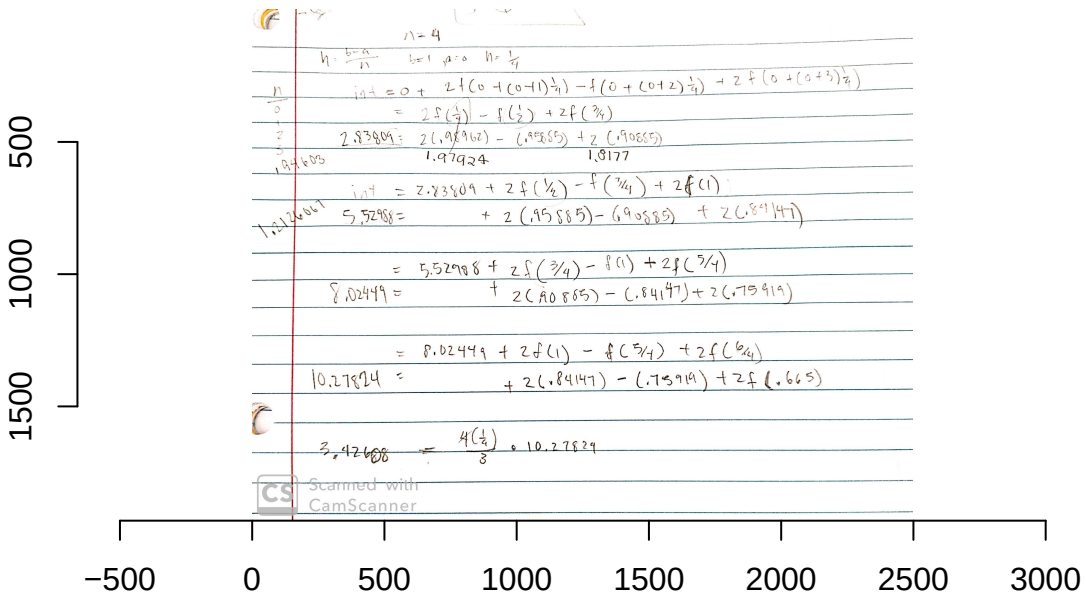


We can see that the open-newton and the right sum formulas are not too far off from the true solution, all are within 2 decimal places, which is good since it shows us that the variability with the function isn't wildly swinging. As in theres no .94 then 1.01, etc.

Here is the link to the replit webpage: https://repl.it/repls/PeriodicAvariciousSubweb

Results

By hand n = 4



This is doing the Newton-Cotes function by hand. Where $h = 1/4$, $a = 0$, and $b = 1$ and the *int* function changes with each iteration of n. I started at $i = 0, 1, 2, 3$ since it corresponds with $n = 4$. However the

output leads me to 3.42608 a value too high for the true function value of around .94. It is only 4 iteration and there might require more iteration to get to around .94 however, the function values only seen to increase so its more likely there is an issue with the way I am calculating it.

Relative errors

The relative error formula is $ |estimated - true|/true$

These are the solutions for different n values in replit for the Newton and the right-sum. Aswell as the true solution from maple.

```
# Maple solution
true = .9460830704

# newton solutions
n40 = 0.9460830649853315
n100 = 0.9460830649853315
n1000 = 0.9460830649853315

# Right sum solution

r40 = .944085771712504
r100 = .9452879155497719
r1000 = .9460037807621953
```

We can see that the relative error for the Newton-Cote function is within 10^-9 in terms of error compare to the true solution. However the Right sum method is much higher in error with only at $n = 1000$ being a reasonable error with 10^-5, while lower iterations yields higher error.

```
abs(n40 - true)/true
```

```
## [1] 5.723248e-09
```

```
abs(n100 - true)/true
```

```
## [1] 5.723248e-09
```

```
abs(n1000 - true)/true
```

```
## [1] 5.723248e-09
```

```
abs(r40 - true)/true
```

```
## [1] 0.002111124
```

```
abs(r100  - true)/true
```

```
## [1] 0.0008404704
```

```
abs(r1000 - true)/true
```

```
## [1] 8.380833e-05
```

In conclusion, the Newton-Cote method is good with much lower n values compare to the right sum method and is within 10^-9 relative error to the true solution from maple.