

MAT 126.01, Prof. Bishop, Tuesday, Nov 5, 2020

Section 3.5 Other Strategies for Integration

Section 3.6 Numerical Integration

Today's material will not be on the midterm or final.

We have been learning how to integrate, but in practice, most integrals that are easy enough to have a simple formula have been computed already.

A table of common integrals is found in your textbook; more extensive lists can be found in reference books or online.

There are various software packages like **MATLAB**, **MAPLE** and **MATHEMATICA** that can compute integrals symbolically or numerically.

Symbolic = gives a formula

Numerical = gives a decimal answer

I have posted links to some tables of integrals on the class webpage.

Of the various computer packages, I am most familiar with **MATLAB** and **MATHEMATICA**.

You can use these on the SBU virtual SINC site or download them from SBU DoIT (free while you are a student using the university license).

I will demonstrate **MATLAB**.

Go to SBU Virtual SINC Site:

<https://vdi.cloud.stonybrook.edu/appblast/webclient/index.html#/launchitems>

The link is also on the class webpage.

Enter NetID and password

Click “Virtual SINC Site”

You should get a Desktop (you may be asked to install some VM software the first time you do this).

On the Desktop, click the Windows icon in the lower left corner

To download onto your computer:

Go to SBU DoIT Software Catalog:

<https://it.stonybrook.edu/services/software-catalog/browse>

Scroll down to **MATLAB** and click on name.

This leads to a page with three options: create account, download, activate. You will have to create a Mathworks account (this is the company that makes **MATLAB**), then download the program. When you download, there are many extra add-ons available (machine learning, signal processing,..) but you won't need most of these. The Symbolic Math Package is important for doing symbolic integration. Once downloaded, you will be prompted to activate the software (and this has to be done every year).

There are detailed instructions on the DoIT page.

Let's assume you have MATLAB running.

Symbolic indefinite integrals:

```
>> syms x
>> int( x^3 * sin(x))
ans =
cos(x)*(- x^3 + 6*x) + sin(x)*(3*x^2 - 6)
```

The line `syms x` tells the program to treat `x` as a symbol, not a number.

Note that `^` is used for powers and `*` for multiplication.

Symbolic definite integrals:

```
>> int(sin(x))
```

```
ans =
```

```
-cos(x)
```

```
>> int(sin(x),0,pi)
```

```
ans =
```

```
2
```



```
>> int((x^3+x^2+1)/(x^4-8))
ans = log(x - 2^(3/4))*(2^(1/4)/8 + 2^(3/4)/32 + 1/4)
+ log(9 - (9*2^(1/2))/4 - 9*x*(1/4 - (9*2^(1/2))/32)^(1/2))*
  ((1/4 - (9*2^(1/2))/32)^(1/2)/4 + 1/4)
- log(9*x*(1/4 - (9*2^(1/2))/32)^(1/2) - (9*2^(1/2))/4 + 9)*
  ((1/4 - (9*2^(1/2))/32)^(1/2)/4 - 1/4)
- log(x + 2^(3/4))*(2^(1/4)/8 + 2^(3/4)/32 - 1/4)
```

```
>> int((x^3+x^2+1)/(x^4-8),0,1)
ans = log(14^(1/4)/2) - (2^(3/4)*(atan(2^(1/4)/2)
+ atanh(2^(1/4)/2)))/16 + (2^(1/4)*(atan(2^(1/4)/2)
- atanh(2^(1/4)/2)))/4
```

```
>> double(ans)
ans = -0.2058
```

The command **double** converts symbolic answer to double precision floating point number.

```

>> int((x^3+x^2+1)/(x^4-8),0,1)
ans =
log(14^(1/4)/2) - (2^(3/4)*(atan(2^(1/4)/2)
+ atanh(2^(1/4)/2)))/16 + (2^(1/4)*(atan(2^(1/4)/2)
- atanh(2^(1/4)/2)))/4

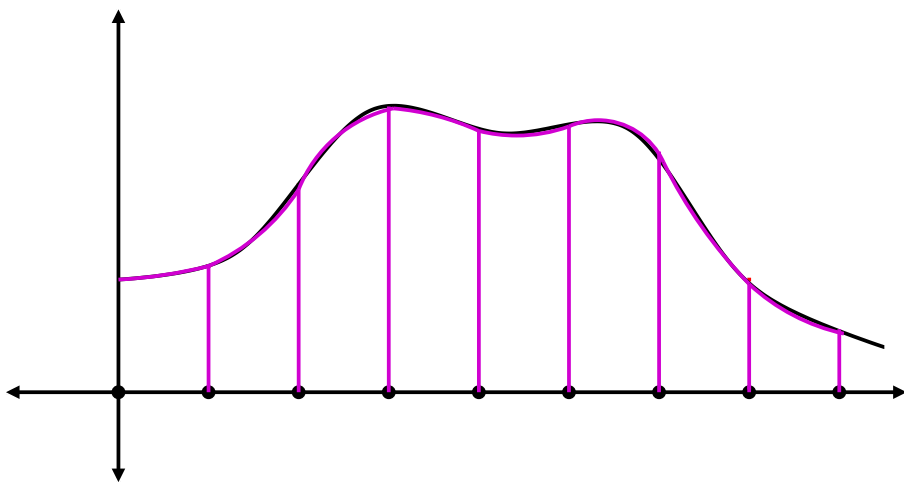
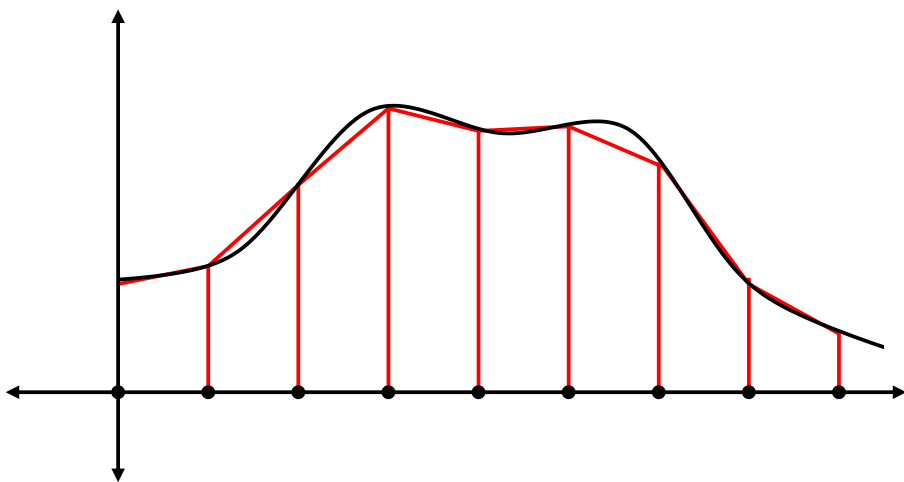
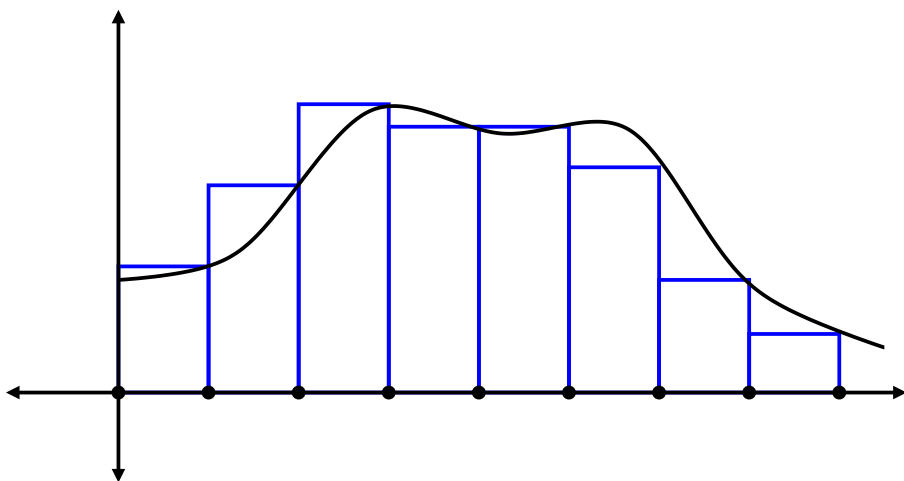
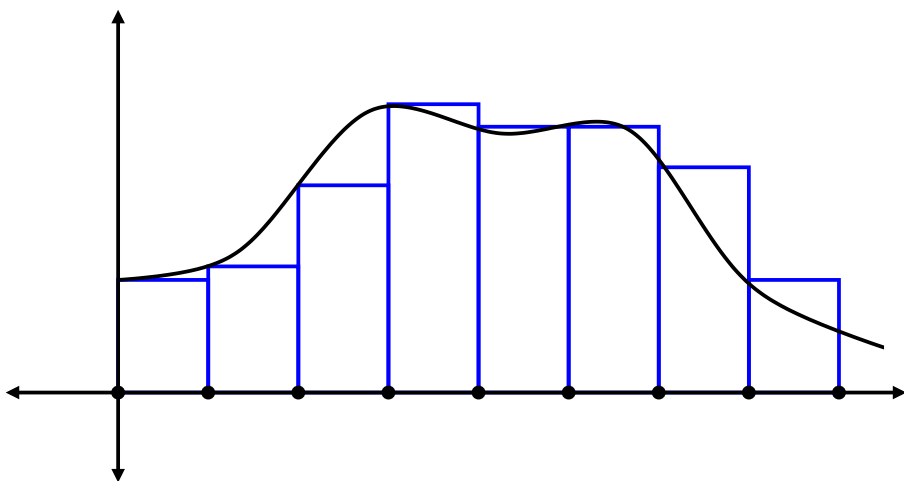
>> vpa(ans,200)
ans =
-0.20584007453153161881859664341809700818486543465
42064715504953977454785268975440193037449190568527
74280200288818487822721982317748992169722262087568
62993858461704228727050757582511446753995605432352426

```

vpa converts to decimal of given precision.

Section 3.6: Numerical Integration

- Left and Right hand rules
- Trapezoid Rule
- Midpoint Rule
- Simpson's Rule



Error bound for Left and Right hand Rules:

if $M = \max |f'|$ over $[a, b]$.

$$\text{error} \leq \frac{M(b-a)^2}{n}.$$

Trapzoid Rule is average of Left and Right hand rules.

Error bound for Traezoid Rule: if $M = \max |f''|$ over $[a, b]$.

$$\text{error} \leq \frac{M(b-a)^3}{12n^2}.$$

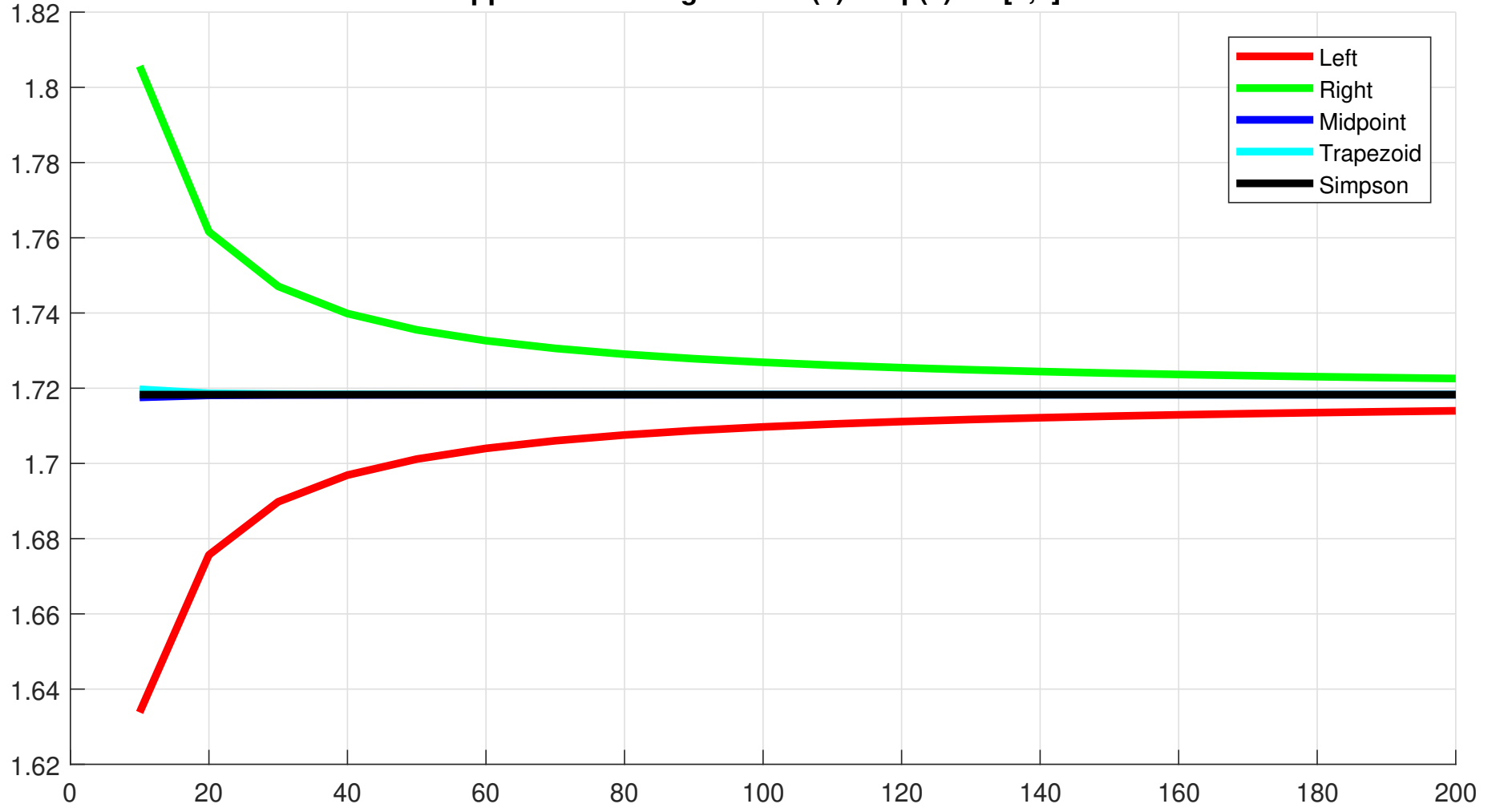
Simpson's rule approximates graph by parabolas. For n even,

$$\text{Simpson} = \frac{\Delta x}{3}(f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \cdots + 4f(x_{n-1}) + f(x_n)).$$

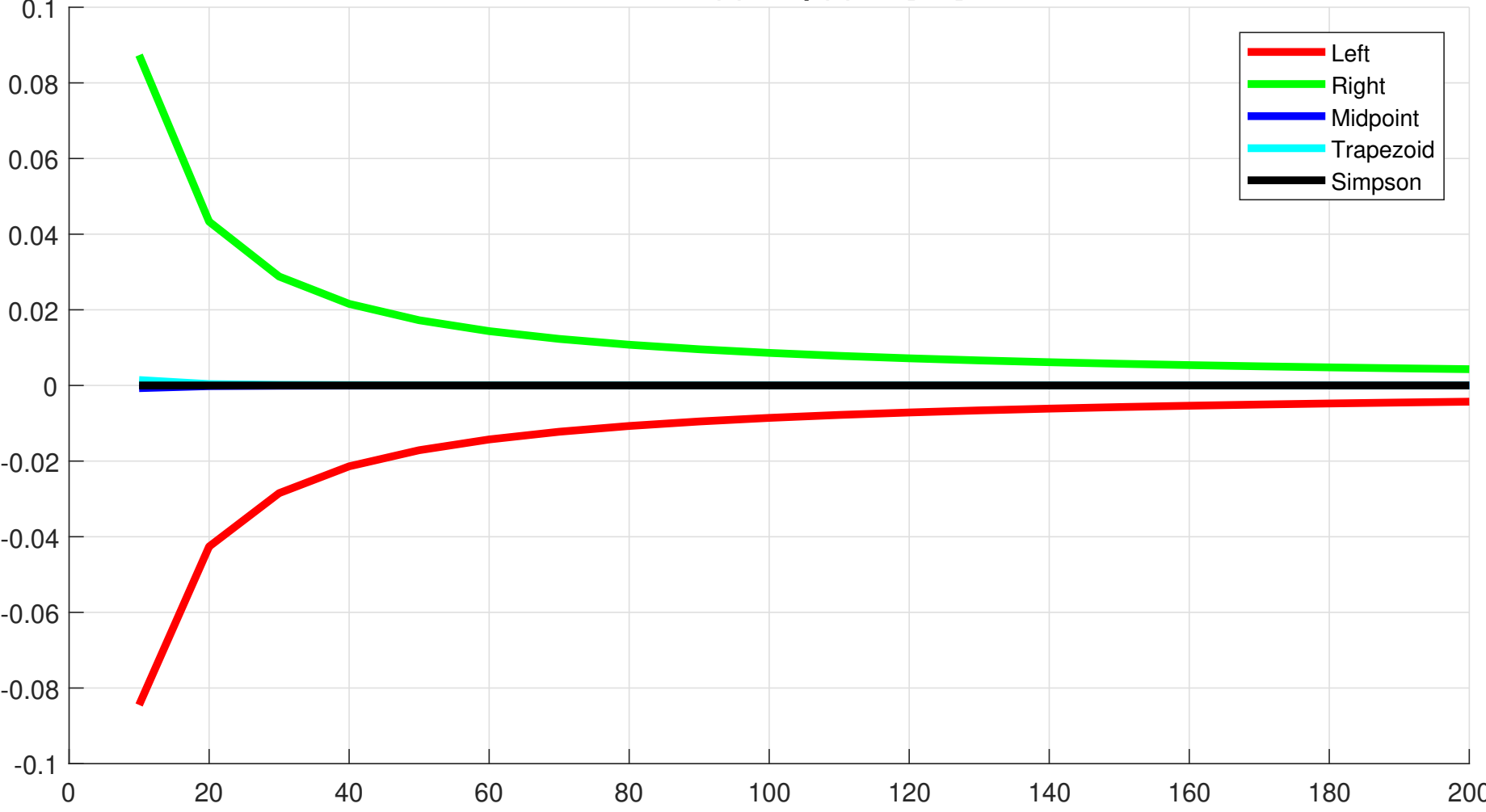
Error bound for Simpson's Rule: If $M = \max |f''''|$ on $[a, b]$.

$$\text{error} \leq \frac{M(b-a)^5}{180n^4}.$$

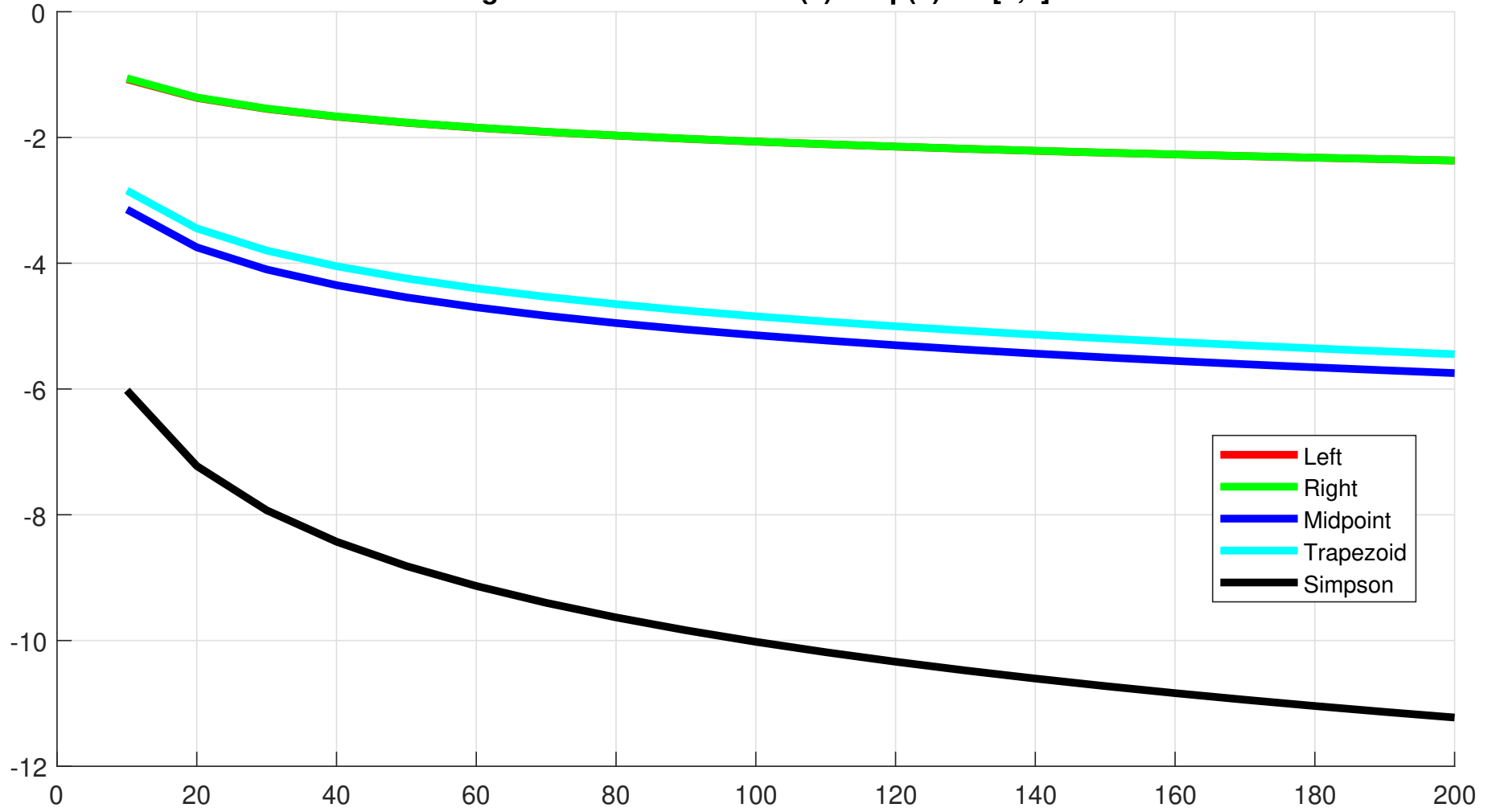
Approximate integrals for $f(x)= \exp(x)$ on $[0,1]$



Errors for $f(x)= \exp(x)$ on $[0,1]$



Log base 10 of errors for $f(x)= \exp(x)$ on $[0,1]$



Are there methods better than Simpsons's Rule?

Are there methods better than Simpson's Rule?

Definitely yes. But these are a bit too complex to describe here. I sometimes cover these in MAT 331.

Gauss quadrature will estimate the integral of a “nice” function like e^x with

$$\text{error} \approx \frac{1}{n!},$$

using n specially chosen points

$$\int_a^b f(x)dx \approx \sum_{k=1}^n w_k f(x_k).$$

Figuring out the correct points $\{x_k\}$ and weights $\{w_k\}$ requires more advanced math, like linear algebra. Most serious integration programs use methods like this.

