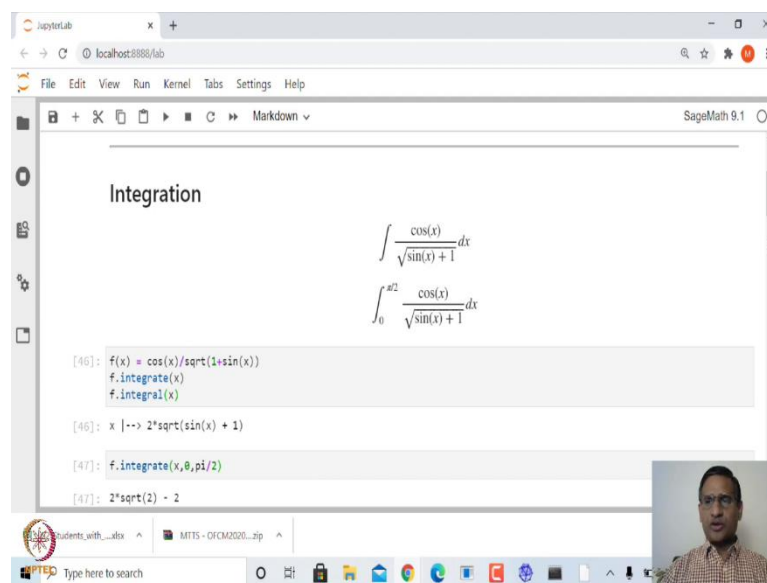


**Computational Mathematics with SageMath**  
**Prof. Ajit Kumar**  
**Department of Mathematics**  
**Institute of Chemical Technology, Mumbai**

**Lecture – 21**  
**Integration with SageMath**

Welcome to the 21st lecture on Computational Mathematics with SageMath. In previous lectures, we looked at finding derivatives and some of their applications. In the next few lectures, we will look at dealing with integrals and their applications.

(Refer Slide Time: 00:37)

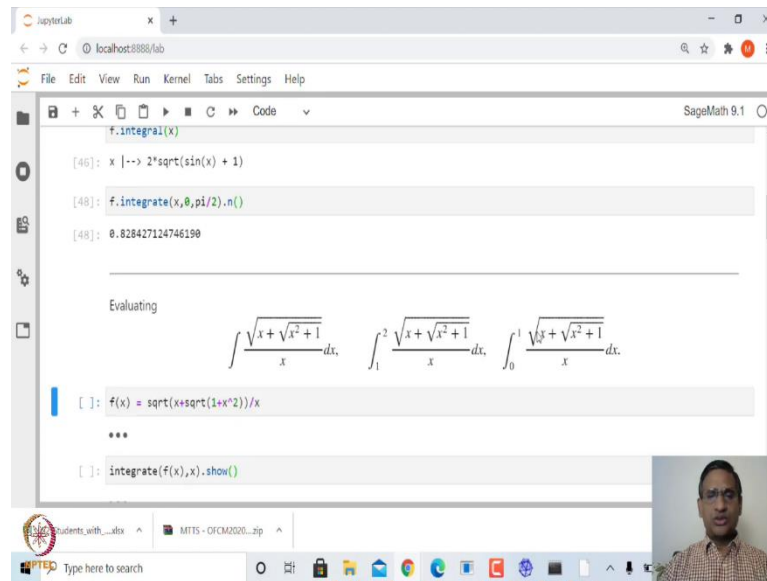


Suppose you have to find integration or integral of this kind, the integral of  $\frac{\cos(x)}{\sqrt{\sin(x)+1}} dx$  and the integral of the same function from 0 to  $\pi/2$ . This is called indefinite integral; this is what is called a definite integral. How does one find this in SageMath?

First, you need to define the function. The function I am calling as  $f(x) = \frac{\cos(x)}{\sqrt{\sin(x)+1}}$  and to find its integral indefinite integral, you simply need to use `f dot integrate` or `f dot integral` with respect to the variable  $x$ .

You could also use `f dot integral`, `f dot integral` that will also work with respect to  $x$ . Both give you the same answer. Now in case you want to find this 2nd integral, that is a definite integral where the limit is 0 to  $\pi$  by 2, then you need to supply that limit after  $x$ . You say `comma 0 comma  $\pi$  by 2`. This is a definite integral of the same function with the limit going from 0 to  $\pi$  by 2.

(Refer Slide Time: 02:01)



```
f.integral(x)
[46]: x |--> 2*sqrt(sin(x) + 1)
[48]: f.integrate(x,0,pi/2).n()
[48]: 0.828427124746190
```

Evaluating

$$\int \frac{\sqrt{x+\sqrt{x^2+1}}}{x} dx, \quad \int_1^2 \frac{\sqrt{x+\sqrt{x^2+1}}}{x} dx, \quad \int_0^1 \frac{\sqrt{x+\sqrt{x^2+1}}}{x} dx.$$

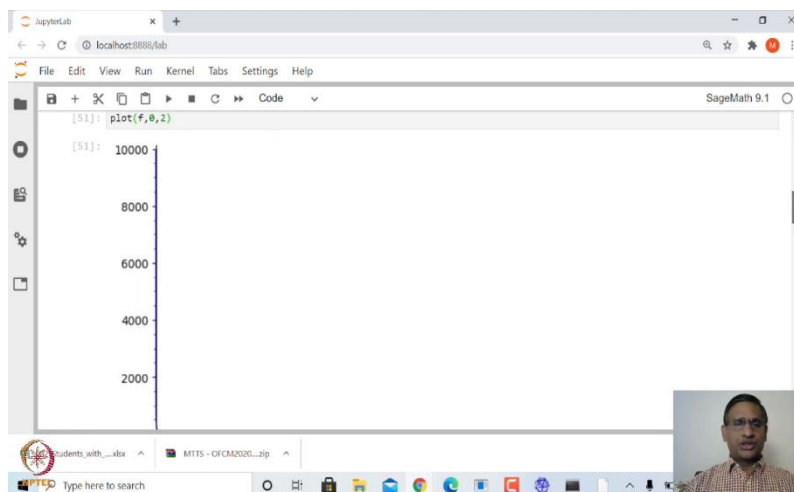
```
[ ]: f(x) = sqrt(x+sqrt(1+x^2))/x
***
[ ]: integrate(f(x),x).show()
```

This is  $2\sqrt{2} - 2$ . If you want the numerical value, you can just say dot n, this will give you numerical value. Now let us look at, suppose we need to find integral of this function. This is the integrand and so we want to find integral of  $\frac{\sqrt{x+\sqrt{x^2+1}}}{x} dx$  and its definite integral between 0 to 1 and 1 to 2.

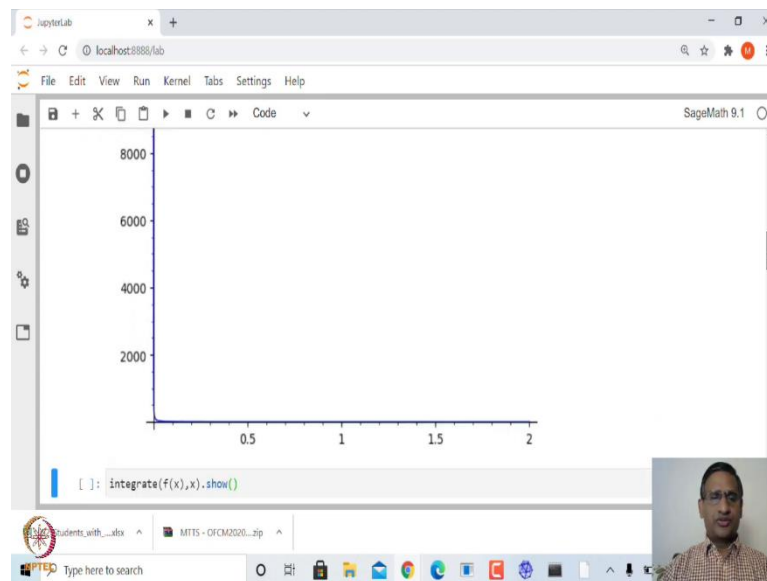
This is a fairly complicated function, if you try to do it by hand, it may take some efforts. Now let us see

how Sage finds these integrals. Let me again define this function  $f(x) = \frac{\sqrt{x+\sqrt{x^2+1}}}{x}$ .

(Refer Slide Time: 02:44)

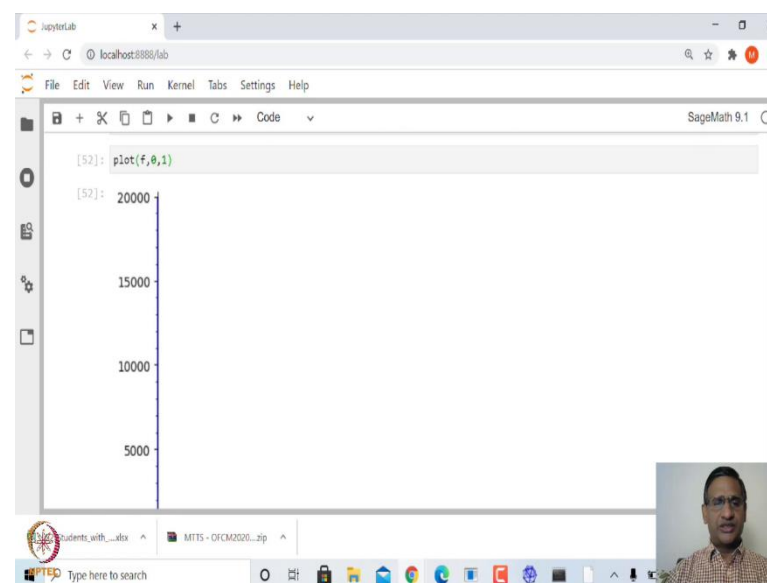


(Refer Slide Time: 02:55)

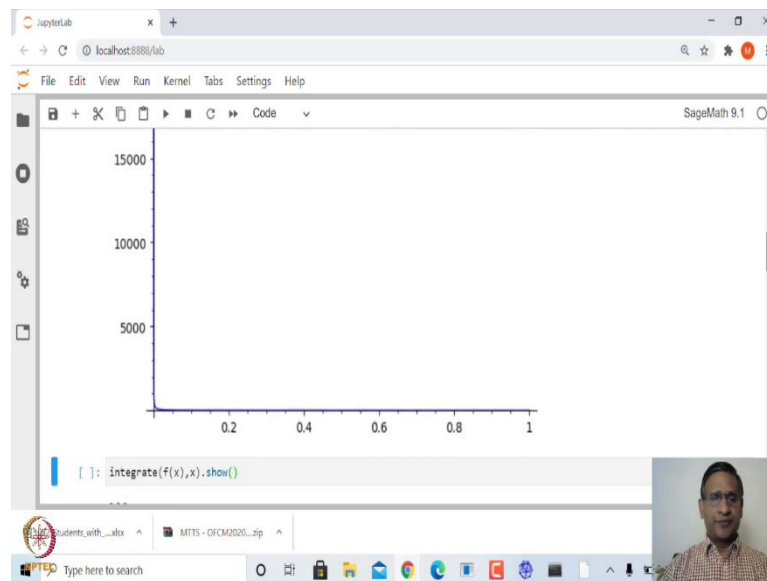


And let us plot the graph of this function. This is the function  $f(x)$  and its graph. You can see here this graph, near 0 it goes to very high value and when  $x$  increases this goes to close to 0. That is what you can see from this graph. However, let us look at what happens to this between 0 and 1.

(Refer Slide Time: 03:14)

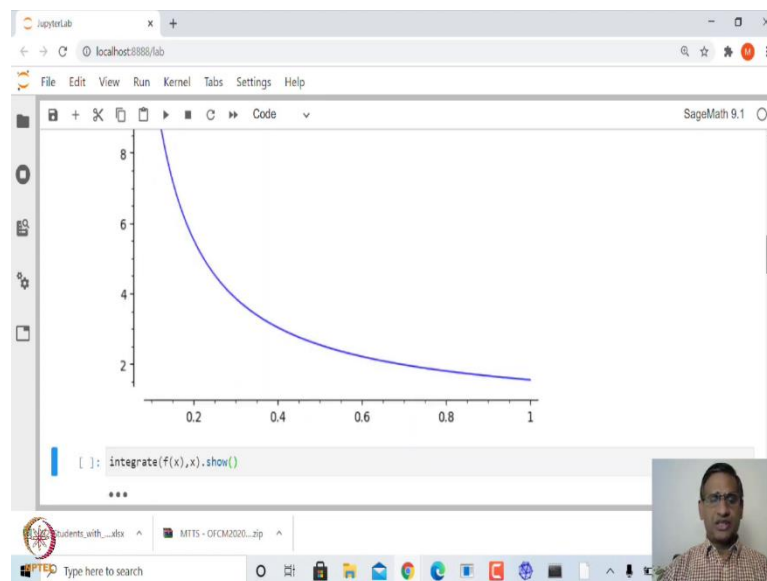


(Refer Slide Time: 03:17)



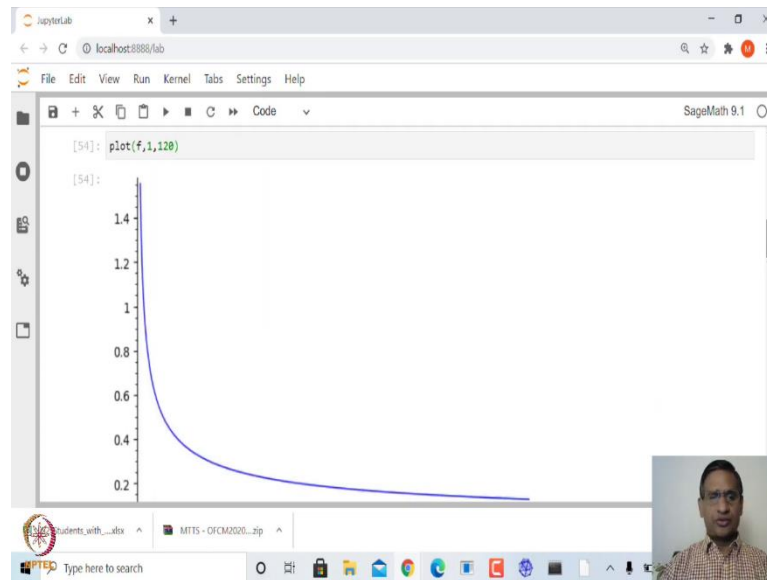
Again at 0, since the behavior is that, it is going to increase to  $\infty$ .

(Refer Slide Time: 03:32)



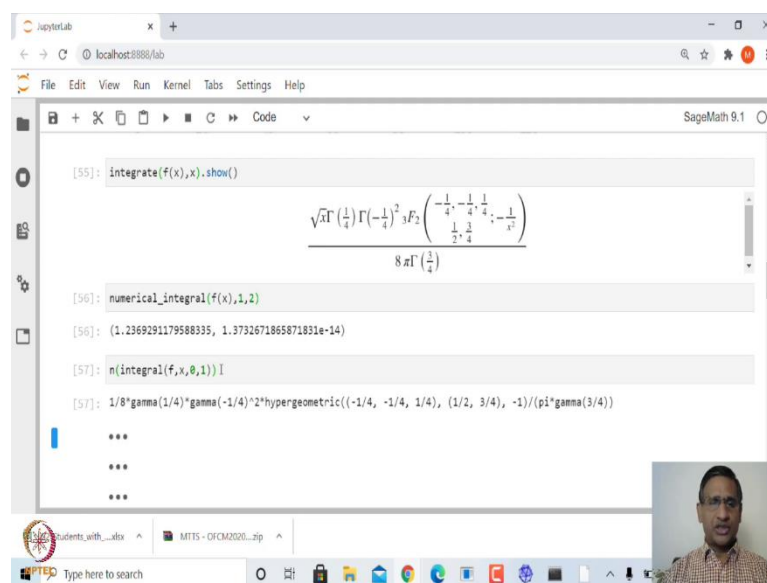
That is what you see this graph. Let me plot, may be between 0.1 and 1, it may be somewhat better. That is how the graph looks like between 0.1 and 1

(Refer Slide Time: 03:39)



If I say what is a graph between 0 and let us say 1 and 2, or 1 and let us say 20, this became 120. This is the graph, so you can see here this goes very close to 0. Now suppose if you want to find the integral of this function.

(Refer Slide Time: 04:09)

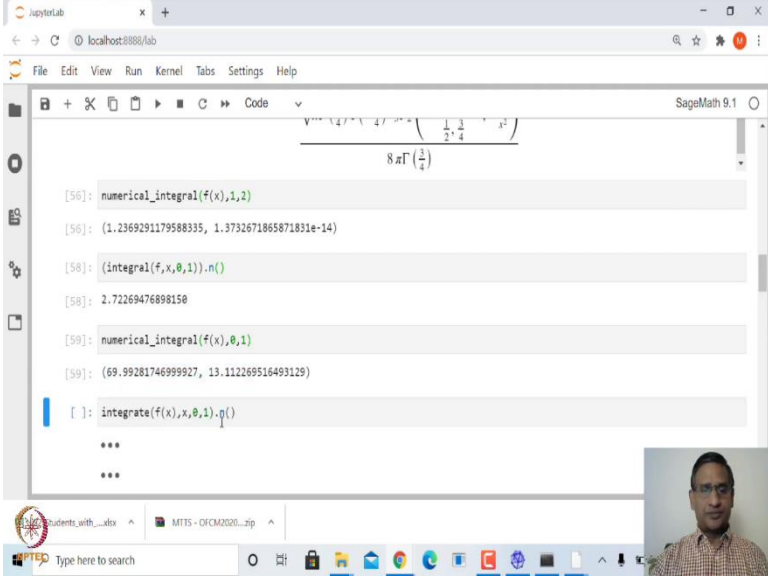


And let us ask it to show what is the integral value. The Sage is able to find integral, but this integral is given in terms of the gamma function and f2 function et cetera. This seems very complicated. I am not sure whether you will be able to make sense out of this but let us see, suppose if I want to find out this integral between 1 and 2.

Sage has an inbuilt function called numerical integral. This finds the integral numerically and this will give you the value of the integral along with error. In this case, you can see here this integral is 1.2369 and whereas, the error is 10 to the power - 14.

Whereas the same thing if you try to find out integral between, between 0 and 1. Let us say take the numerical value of this, this is not actually numerical integration it is finding the numerical integral and it is finding in terms of hypergeometric function and gamma function. This is what is called the hypergeometric function.

(Refer Slide Time: 05:19)



The screenshot shows a JupyterLab window with a SageMath 9.1 kernel. The code in the cell is as follows:

```

f(x) = 1/(4*x^2 - 1)
numerical_integral(f(x), 1, 2)
(integral(f(x), 0, 1)).n()
numerical_integral(f(x), 0, 1)
integrate(f(x), x, 0, 1).p()

```

The output shows the following results:

```

[56]: (1.2369291179588335, 1.3732671865871831e-14)
[58]: 2.72269476898158
[59]: (69.99281746999927, 13.112269516493129)
[ ]:

```

If I say dot n, if I say dot n of this, this may give me the numerical value of whatever it has found, it takes time because this function is fairly complicated. This is 2.72269 whereas if you apply numerical integral to the same function between 0 and 1 and then see what you are getting.

The integral is 69.98 whereas, from this, we got 2.72 and this is 69.99 whereas, the error this is quite high the one cannot accept this kind of error in case of numerical computation. This is very bad behavior in this case. What I am trying to tell you with this example is that just do not believe whatever output that you get with any software. You should try to understand what is happening and then only you accept. It is very important to learn the concept and then use this kind of software to explore it. This we have already done.

(Refer Slide Time: 06:41)

```

[58]: 2.72269476898158
[59]: numerical_integral(f(x),0,1)
[59]: (69.99281746999927, 13.112269516483129)
[ ]:
[60]: numerical_integral(sin(x)/x,0,1)
[60]: (0.946083070367183, 1.0503632079297887e-14)


$$\int_0^{\infty} \frac{\sin x}{x} dx$$


[ ]: integrate(sin(x)/x,x, 0,oo)
...

```

Similarly, if you try to find, for example, the integral of  $\sin(x)$  by  $x$  between let us say 0 to 1. It gives you the value, this is integral and this is the error term, which is quite alright. If you try to look at this integral  $\sin(x)$  upon  $x$ , this is actually a very important and famous integral. And this integral exists. If one limit here is infinity, such integrals are called improper integral.

(Refer Slide Time: 07:25)

```

[ ]:
[60]: numerical_integral(sin(x)/x,0,1)
[60]: (0.946083070367183, 1.0503632079297887e-14)

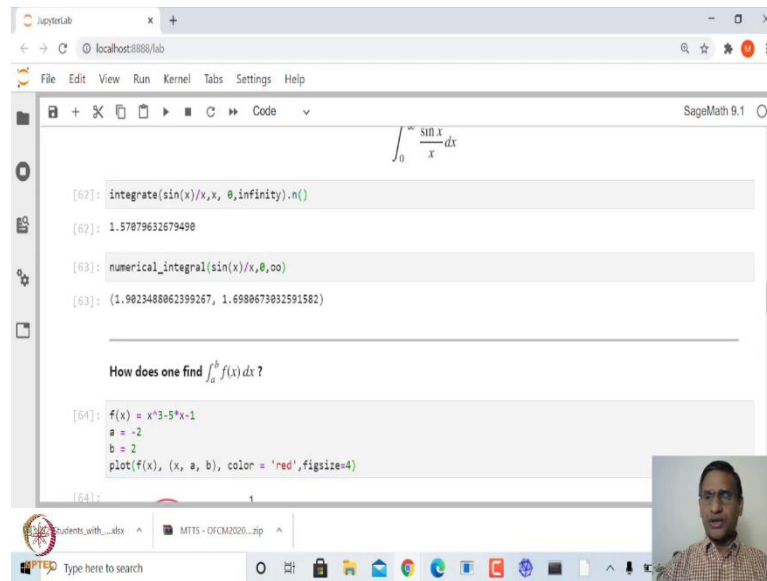

$$\int_0^{\infty} \frac{\sin x}{x} dx$$


[61]: integrate(sin(x)/x,x, 0,infinity).n()
[61]: 1/2*pi
...

```

This one can find integral of this  $\sin x$  upon  $x$  with respect to  $x$  from 0 to  $\infty$ ,  $\infty$  means infinity, or you can write  $\infty$ , that also you can do. This is half into  $\pi$  or this is  $\pi$  by 2. If you look at what is the numerical value of  $\pi$  by 2, then this gives you 1.5707.

(Refer Slide Time: 07:56)

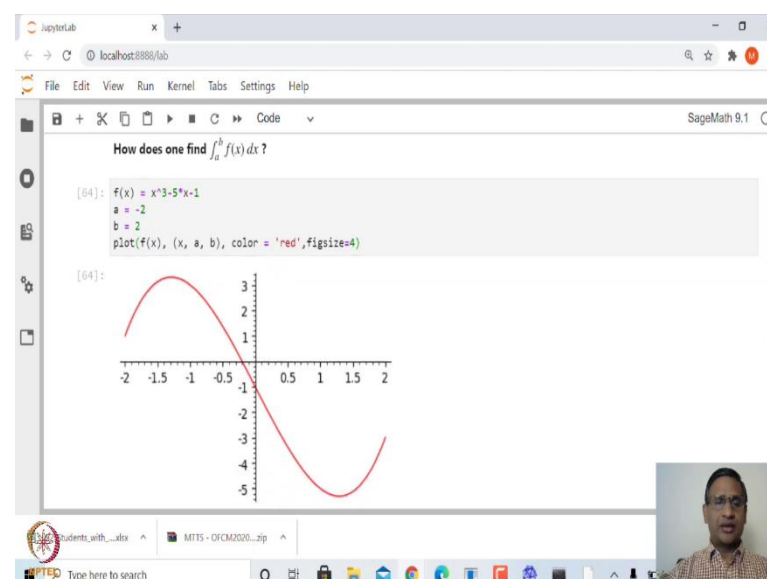


Whereas, if you try to find numerical integral using the same function between 0 to  $\infty$ . When one wants to find the numerical integral, one has to divide the interval. In that case, the interval of infinite length will not be very convenient.

That is why you can see here when you integrate this now you are getting the answer which is 1.90, whereas, actual answer is  $\pi$  by 2. Of course, you can see here again, the error term is very high 1.698, quite close to 1.70 which is a very high error.

Again, when you want to compute numerical integral, even with Sage or for that matter any other software of this kind, you should be careful. Now, the question is how does one find such an integral. For example, SageMath, how does it find this kind of integrals.

(Refer Slide Time: 09:09)

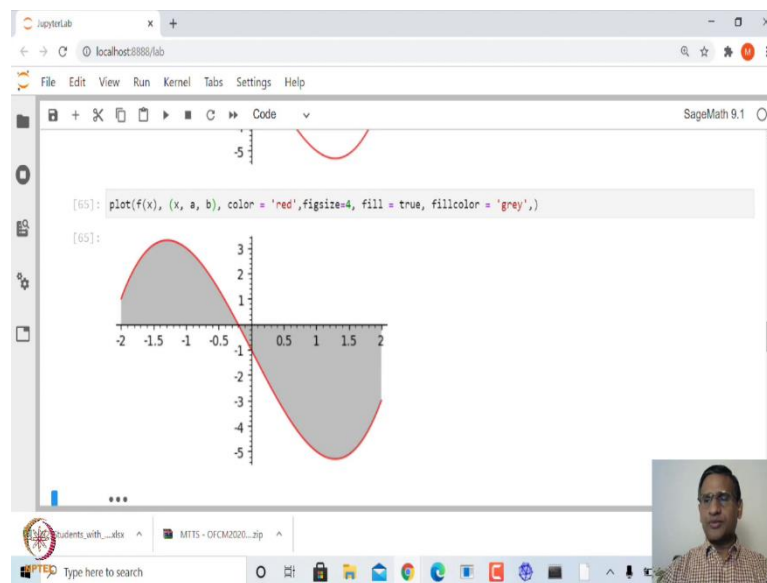




Let us try to understand when we say integral of a function in an interval  $a$  to  $b$ . What it means is, for example, there is an integral of this function, we want to find out between let us say minus 2 to 2. Then what it means is that you have to find the area between this curve and the x-axis, below this curve from - 2 to 2.

That is actually, you have to say signed area because this side it will be negative. The integral can be negative, whereas, the area is always a positive quantity, non-negative quantity. The signed area is nothing, but integral. That is what we need to do.

(Refer Slide Time: 09:54)



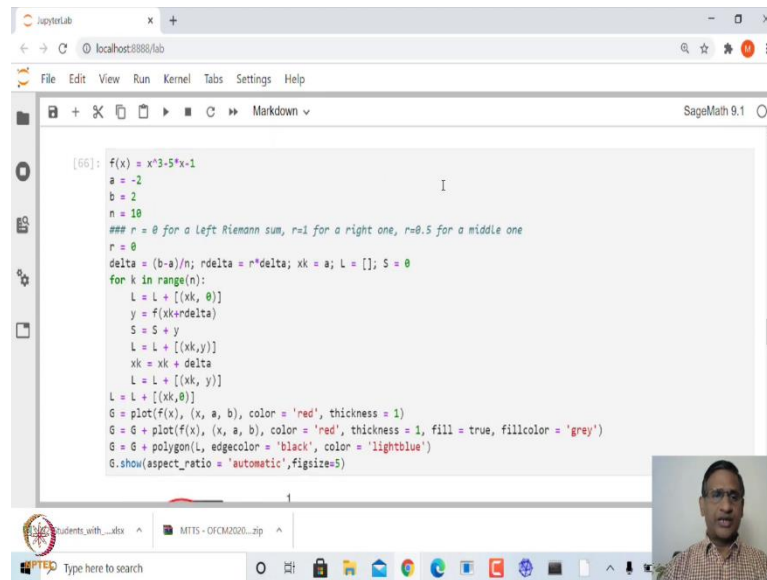
If I look at them and try to shade this region, this is what this area of this signed area of this, which you need to find out. For example, here will be the area under this curve this will be negative of the area under this curve and when you add these two. This should give you the integral of this function from minus 2 to 2.

From the knowledge of computing area from school, you know certain geometric figure and their area. For example, the way one has to calculate the area under this curve is by splitting this into some region whose area we know.

For example, one of the things we can do is, we can try to approximate this area by means of the rectangle. For that what one can do is, one can divide this domain from minus 2 to 2 in equal parts, let us say,  $n$  equal parts and for each part try to draw a rectangle.

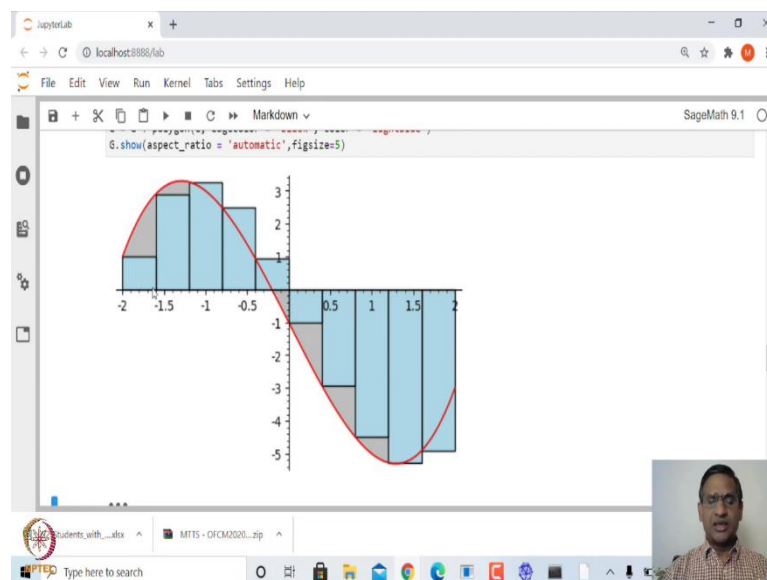
For example, if I take between minus 2 and minus 1.5, this is the region you can approximate. You can take the triangle by taking the left endpoint or from the right endpoint or the middle point or for that matter any point as a height of that rectangle.

(Refer Slide Time: 11:25)



```
[66]: f(x) = x^3-5*x-1
a = -2
b = 2
n = 10
## r = 0 for a Left Riemann sum, r=1 for a right one, r=0.5 for a middle one
r = 0
delta = (b-a)/n; rdelta = r*delta; xk = a; L = []; S = 0
for k in range(n):
    L = L + [(xk, 0)]
    y = f(xk+rdelta)
    S = S + y
    L = L + [(xk,y)]
    xk = xk + delta
    L = L + [(xk, y)]
L = L + [(xk,0)]
G = plot(f(x), (x, a, b), color = 'red', thickness = 1)
G = G + plot(f(x), (x, a, b), color = 'red', thickness = 1, fill = true, fillcolor = 'grey')
G = G + polygon(L, edgecolor = 'black', color = 'lightblue')
G.show(aspect_ratio = 'automatic', figsize=5)
```

(Refer Slide Time: 11:29)



Let us see what I mean by that. Suppose we divide this interval from minus 2 to 2 into let us say 10 equal subintervals and approximate area under this by this rectangle, height is taken as the value of the left endpoint everywhere, the left endpoint is the height of the rectangle, and this is the width.

If you try to approximate the integral using sum of the areas of this rectangle, you will see that these are portions are left out whereas these are extra. You expect that to have some error. Whereas if you increase the number of the interval which you are dividing at present, we have taken 10, let me make it 30.

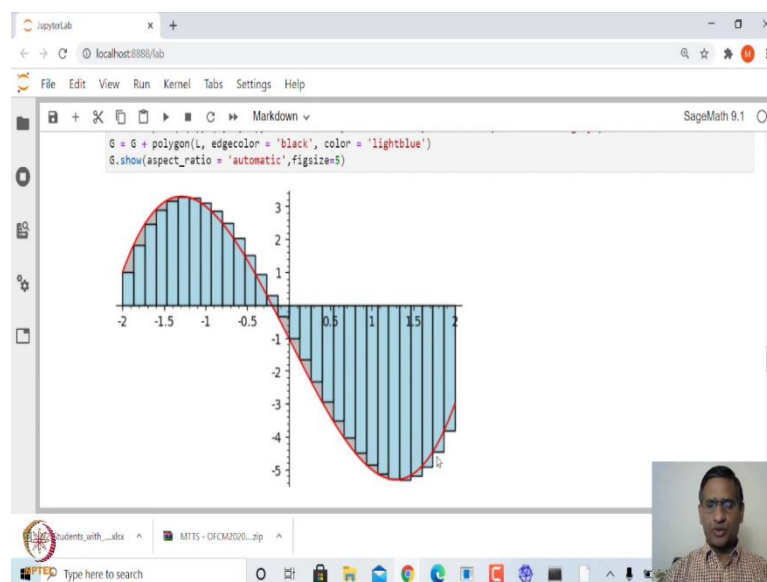
(Refer Slide Time: 12:22)

```

In [1]: f(x) = x^3-5*x-1
a = -2
b = 2
n = 30
## r = 0 for a left Riemann sum, r=1 for a right one, r=0.5 for a middle one
r = 0
delta = (b-a)/n; rdelta = r*delta; xk = a; L = []; S = 0
for k in range(n):
    L = L + [(xk, 0)]
    y = f(xk+rdelta)
    S = S + y
    L = L + [(xk,y)]
    xk = xk + delta
    L = L + [(xk, y)]
L = L + [(xk,0)]
G = plot(f(x), (x, a, b), color = 'red', thickness = 1)
G = G + plot(f(x), (x, a, b), color = 'red', thickness = 1, fill = true, fillcolor = 'grey')
G = G + polygon(L, edgecolor = 'black', color = 'lightblue')
G.show(aspect_ratio = 'automatic', figsize=5)

```

(Refer Slide Time: 12:32)



When I make it 30, then the approximation will be much better. Because now you can see here this the left-out portion and the extra portion are very small.

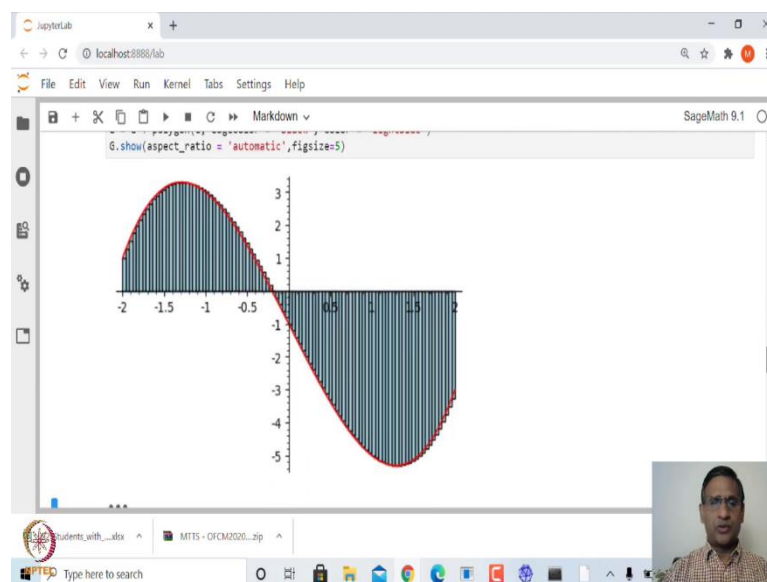
(Refer Slide Time: 12:40)

```

[67]: f(x) = x^3-5*x-1
a = -2
b = 2
n = 10
## r = 0 for a Left Riemann sum, r=1 for a right one, r=0.5 for a middle one
r = 0
delta = (b-a)/n; rdelta = r*delta; xk = a; L = []; S = 0
for k in range(n):
    L = L + [(xk, 0)]
    y = f(xk+rdelta)
    S = S + y
    L = L + [(xk,y)]
    xk = xk + delta
    L = L + [(xk, y)]
L = L + [(xk,0)]
G = plot(f(x), (x, a, b), color = 'red', thickness = 1)
G = G + plot(f(x), (x, a, b), color = 'red', thickness = 1, fill = true, fillcolor = 'grey')
G = G + polygon(L, edgecolor = 'black', color = 'lightblue')
G.show(aspect_ratio = 'automatic', figsize=5)

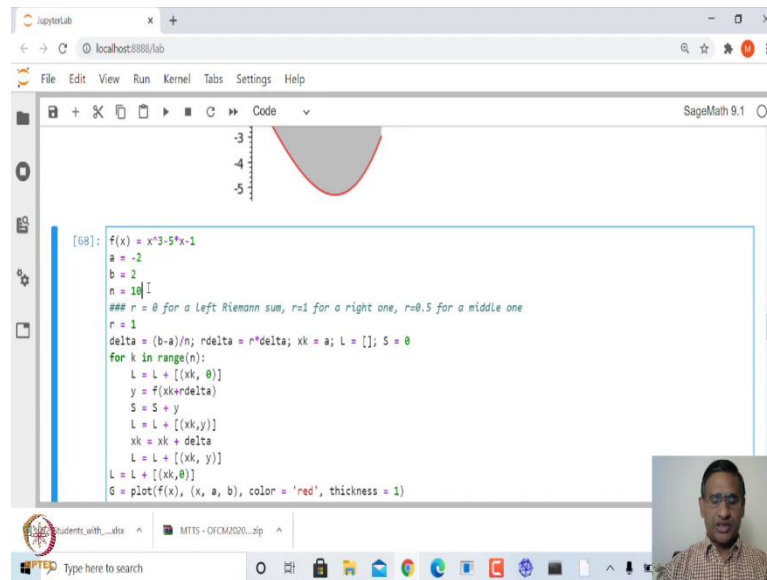
```

(Refer Slide Time: 12:48)



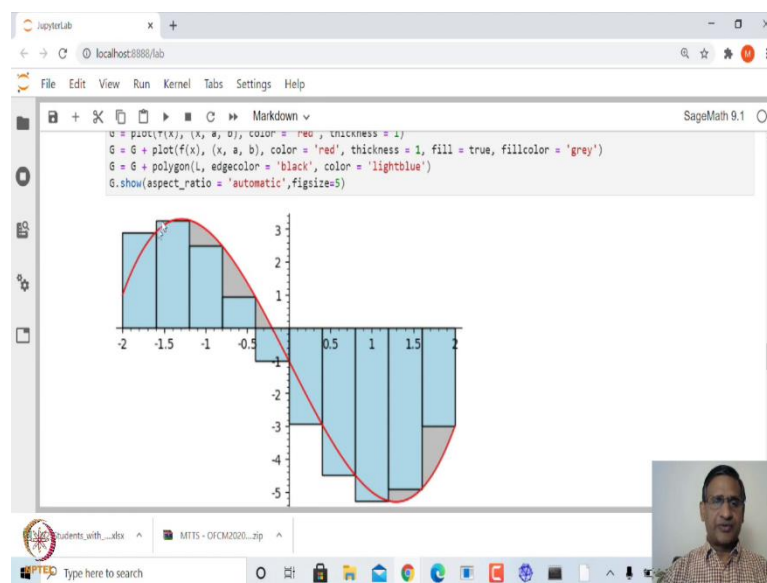
And if you take a greater number of intervals or divisions, let us say if I take 100, then this will be very close to the actual area.

(Refer Slide Time: 13:00)



That is one way to approximate this area. Instead of approximating the area by taking the left endpoint as a height, we can take the right endpoint.

(Refer Slide Time: 13:09)



For example, let me just do it for the 10 subintervals with the right endpoints. For example, here this is the height, the right endpoint, this is the height here, this is the height this is right endpoint so this is the height and so on. One calls this as right Riemann sum.

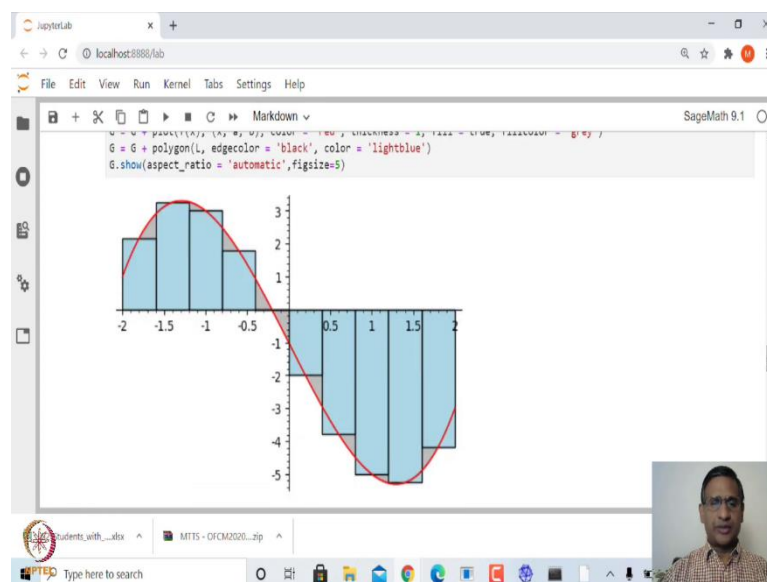
(Refer Slide Time: 13:27)

```

SageMath 9.1
File Edit View Run Kernel Tabs Settings Help
[+] f(x) = x^3-5*x-1
a = -2
b = 2
n = 10
## r = 0 for a Left Riemann sum, r=1 for a right one, r=0.5 for a middle one
r = 0.5
delta = (b-a)/n; rdelta = r*delta; xk = a; L = []; S = 0
for k in range(n):
    L = L + [(xk, 0)]
    y = f(xk+rdelta)
    S = S + y
    L = L + [(xk,y)]
    xk = xk + delta
    L = L + [(xk, y)]
L = L + [(xk,0)]
G = plot(f(x), (x, a, b), color = 'red', thickness = 1)
G = G + plot(f(x), (x, a, b), color = 'red', thickness = 1, fill = true, fillcolor = 'grey')
G = G + polygon(L, edgecolor = 'black', color = 'lightblue')
G.show(aspect_ratio = 'automatic', figsize=5)

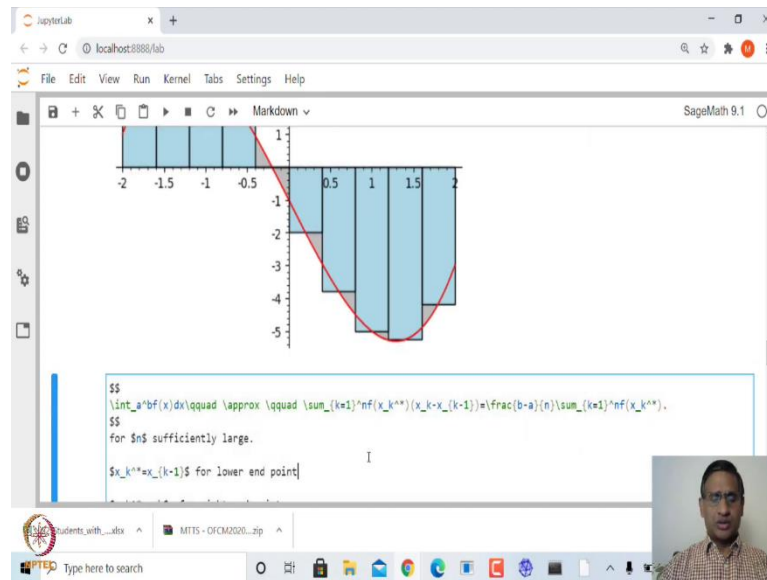
```

(Refer Slide Time: 13:34)

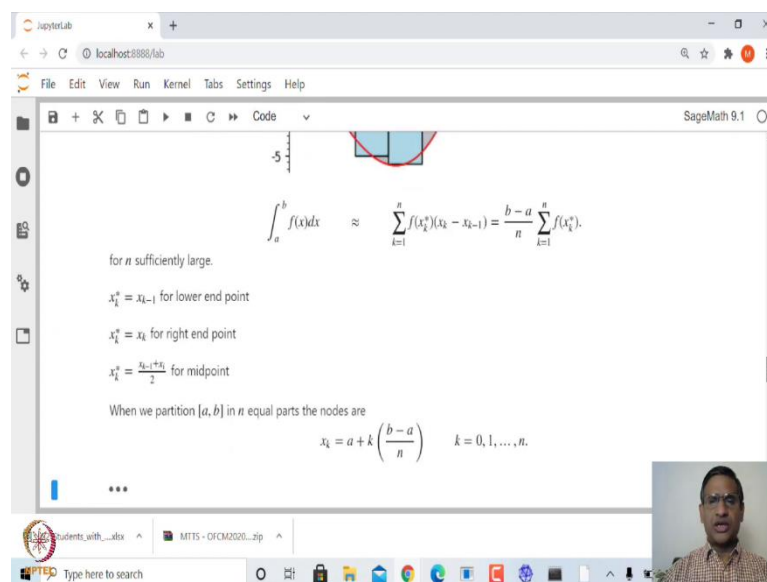


And you can also take a midpoint, for example, if I take  $r$  is equal to 0.5 then it will give you the rectangle whose height is the value of the function at the midpoint of each of this interval. This is what is called the midpoint Riemann sum. And as you increase the number of the subintervals number of points inside this subinterval then you will get close to the area.

(Refer Slide Time: 14:01)



(Refer Slide Time: 14:03)



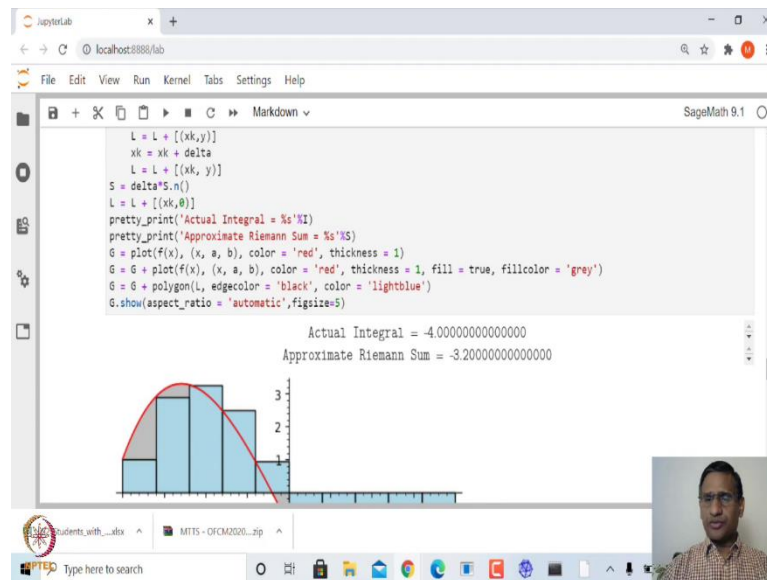
Let us see, you can also in order to define this integral which is called the Riemann integral. What we are doing? We are approximating the integral of  $f(x)$  from  $a$  to  $b$  by means of this sum of the area of these rectangles, where;  $f(x)$  star is a height and this is the width. If you are dividing the interval into  $n$  equal parts, the width is  $b$  minus  $a$  by  $n$  and this is the height so this is the formula.

Now, in the case of the left or lower endpoint left endpoint  $x_k^*$  is  $x_{k-1}$  and in the case of the right endpoint, you have to take  $x_k$  in the case of midpoint you take  $(x_{k-1} + x_k)$  by 2. In principle, one can take any point in between and do this calculation. Generally, in the case of Riemann integral one takes this  $x^*$  as the point at which the minimum occurs.

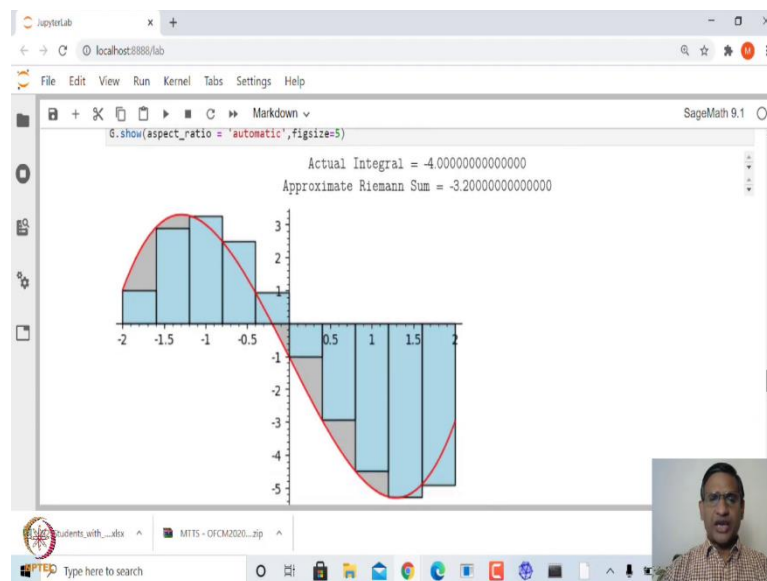
And also, the points at which the maximum occurs, these are called lower Riemann sums and upper Riemann sums and then one has to take what is called supremum and infimum of these 2 sums. Here

$x_k = a + k\left(\frac{b-a}{n}\right)$ , and  $k$  will take value 0 to  $n$ . For example, if  $k$  equal to 0 you are at  $a$ , and if  $k$  is equal to  $n$ , you are at  $b$  and so on.

(Refer Slide Time: 15:32)

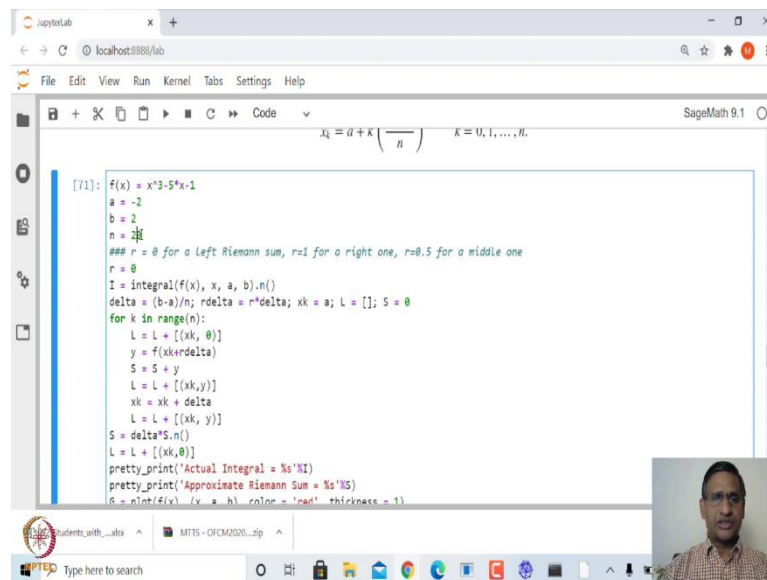


(Refer Slide Time: 15:34)





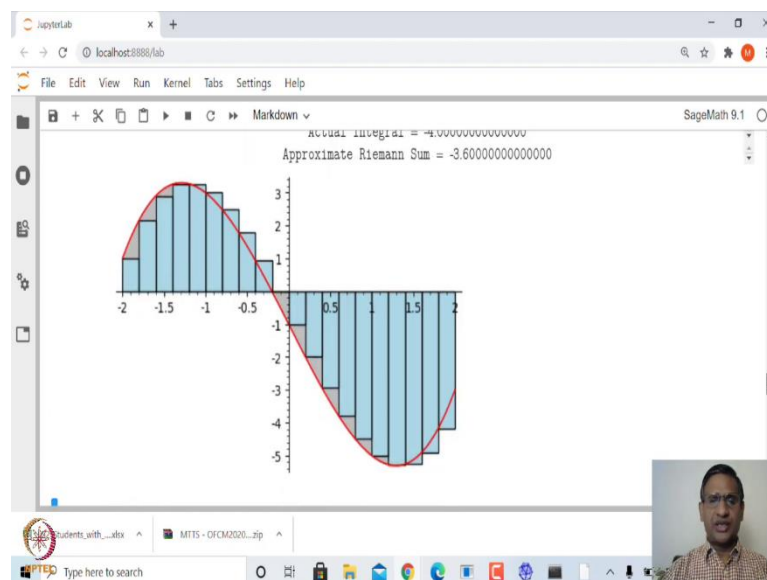
(Refer Slide Time: 15:42)



The image shows a JupyterLab window with a SageMath 9.1 kernel. The code defines a function  $f(x) = x^3 - 5x - 1$  and integrates it from  $a = -2$  to  $b = 2$  with  $n = 20$  subintervals. It calculates the actual integral and an approximate Riemann sum using left endpoints. The formula for the nodes is  $x_k = a + k \left( \frac{b-a}{n} \right)$  for  $k = 0, 1, \dots, n$ .

```
[71]: f(x) = x^3 - 5*x - 1
a = -2
b = 2
n = 20
## r = 0 for a Left Riemann sum, r=1 for a right one, r=0.5 for a middle one
r = 0
I = integral(f(x), x, a, b).n()
delta = (b-a)/n; rdelta = r*delta; xk = a; L = []; S = 0
for k in range(n):
    L = L + [(xk, 0)]
    y = f(xk+rdelta)
    S = S + y
    L = L + [(xk, y)]
    xk = xk + delta
    L = L + [(xk, y)]
S = delta*S.n()
L = L + [(xk, 0)]
pretty_print('Actual Integral = %s'%I)
pretty_print('Approximate Riemann Sum = %s'%S)
P = n*plot(f(x), (x, a, b), color='red', thickness=1)
```

(Refer Slide Time: 15:46)



If you try to see what is the approximate area when you take the left end points, let us take this division as 20, or divides this interval into 12 equal parts. The value of the actual integral which is calculated using sage, it is minus, 4 whereas, this approximate area is minus 3.6.

(Refer Slide Time: 16:08)

```

JupyterLab
localhost:8888/lab

File Edit View Run Kernel Tabs Settings Help

SageMath 9.1

[73]: f(x) = x^3-5*x-1
a = -2
b = 2
n = 50
### r = 0 for a Left Riemann sum, r=1 for a right one, r=0.5 for a middle one
r = 0
I = integral(f(x), x, a, b).n()
delta = (b-a)/n; rdelta = r*delta; xk = a; L = []; S = 0
for k in range(n):
    L = L + [(xk, 0)]
    y = f(xk+rdelta)
    S = S + y
    L = L + [(xk,y)]
    xk = xk + delta
    L = L + [(xk, y)]
S = delta*S.n()
L = L + [(xk,0)]
pretty_print('Actual Integral = %s'%I)
pretty_print('Approximate Riemann Sum = %s'%S)
G = plot(f(x), (x, a, b), color = 'red', thickness = 1)
G = G + plot(f(x), (x, a, b), color = 'red', thickness = 1, fill = true, fillcolor = 'grey')
G = G + polygon(L, edgecolor = 'black', color = 'lightblue')

```

(Refer Slide Time: 16:21)

```

JupyterLab
localhost:8888/lab

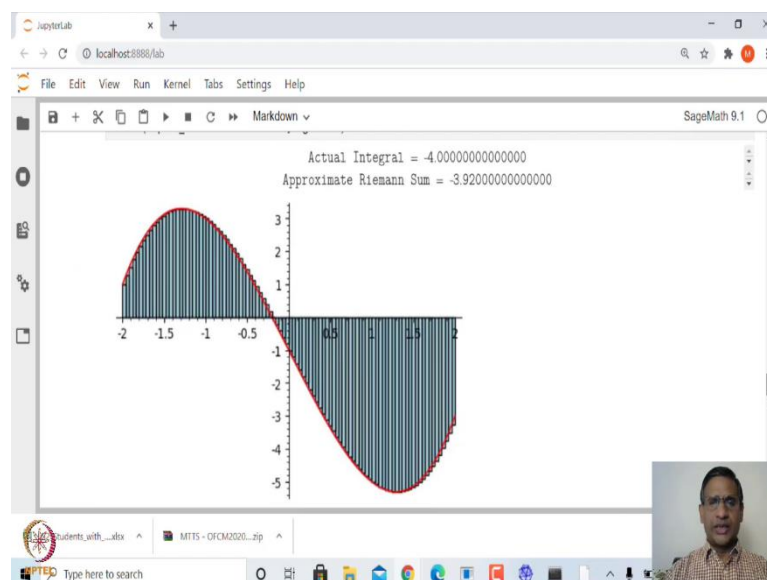
File Edit View Run Kernel Tabs Settings Help

SageMath 9.1

a = -2
b = 2
n = 100
### r = 0 for a Left Riemann sum, r=1 for a right one, r=0.5 for a middle one
r = 0
I = integral(f(x), x, a, b).n()
delta = (b-a)/n; rdelta = r*delta; xk = a; L = []; S = 0
for k in range(n):
    L = L + [(xk, 0)]
    y = f(xk+rdelta)
    S = S + y
    L = L + [(xk,y)]
    xk = xk + delta
    L = L + [(xk, y)]
S = delta*S.n()
L = L + [(xk,0)]
pretty_print('Actual Integral = %s'%I)
pretty_print('Approximate Riemann Sum = %s'%S)
G = plot(f(x), (x, a, b), color = 'red', thickness = 1)
G = G + plot(f(x), (x, a, b), color = 'red', thickness = 1, fill = true, fillcolor = 'grey')
G = G + polygon(L, edgecolor = 'black', color = 'lightblue')
G.show(aspect_ratio = 'automatic', figsize=5)

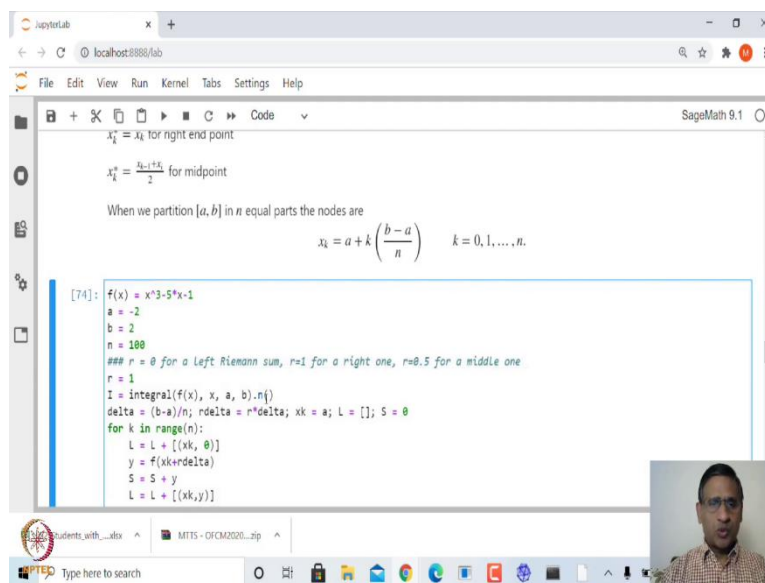
```

(Refer Slide Time: 16:30)



As you increase, this interval, let us say make it 50, then it will be minus 3.84, which is somewhat closer to this. If you increase further, let us say make it 100, then you will get much closer. In, this case, you have this approximate area is minus 3.92 whereas, actually is minus 4.0 and so on.

(Refer Slide Time: 16:45)

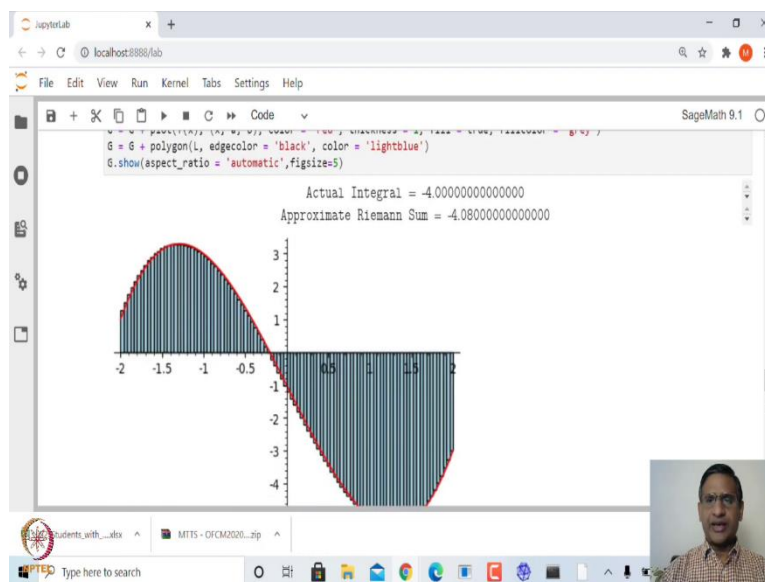


The screenshot shows a JupyterLab interface with a SageMath 9.1 kernel. The code defines a function  $f(x) = x^3 - 5x + 1$  and integrates it from  $a = -2$  to  $b = 2$  using  $n = 100$  intervals. It includes comments for different Riemann sum methods:  $r=0$  for left,  $r=1$  for right, and  $r=0.5$  for midpoint. The code calculates the integral and stores the results in a list  $L$ .

```
[74]: f(x) = x^3-5*x+1
a = -2
b = 2
n = 100
## r = 0 for a Left Riemann sum, r=1 for a right one, r=0.5 for a middle one
r = 1
I = integral(f(x), x, a, b).n()
delta = (b-a)/n; rdelta = r*delta; xk = a; L = []; S = 0
for k in range(n):
    L = L + [(xk, 0)]
    y = f(xk+rdelta)
    S = S + y
    L = L + [(xk,y)]
```

And you could do it with the left endpoint and right endpoint. In this case, if I make  $r$  is equal to 1, this will use the right endpoint.

(Refer Slide Time: 16:53)

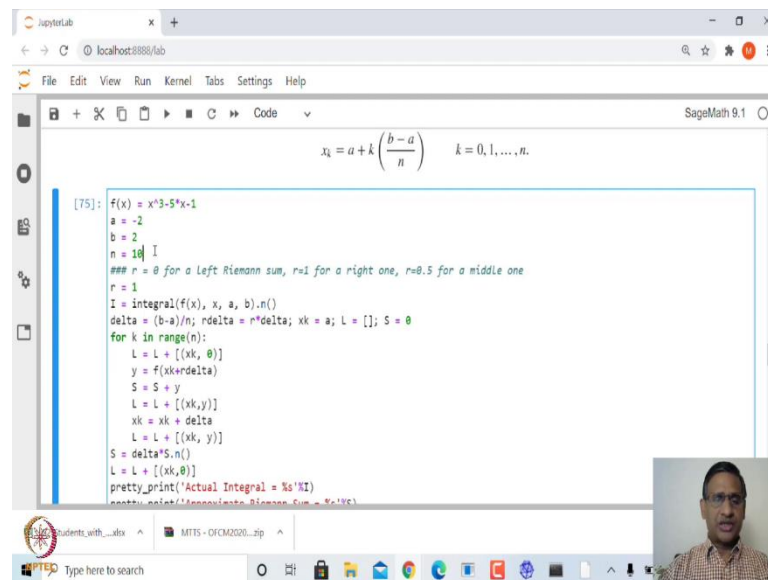


This is the right endpoint; you can see here. If you take the right endpoint, approximate area is somewhat more than the actual area. These codes can see is fairly simple What we are doing is, we are taking the function  $a$  and  $b$  are the endpoints  $n$  is the number of intervals.

$r$  decides whether you want to take the left endpoint or right endpoint or midpoint. Accordingly, you calculate the delta. Delta is the length of each of this interval and  $r \cdot \text{delta}$  is  $r$  times delta. If  $r$  is equal to 0, it will take the left endpoint if  $r$  is equal to 1 it will take the right endpoint if  $r$  is equal to 0.5 it will take the midpoint.

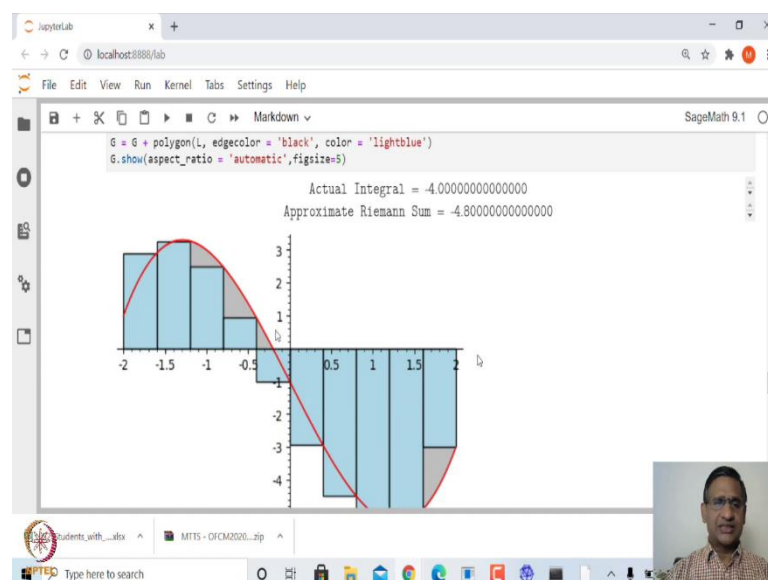
Initially,  $x_k$  is  $a$  and then this is the set of all the points because you need to plot these rectangles. These rectangles, the way you need to store these rectangles.

(Refer Slide Time: 18:03)



```
[75]: f(x) = x^3-5*x+1
a = -2
b = 2
n = 10
### r = 0 for a Left Riemann sum, r=1 for a right one, r=0.5 for a middle one
r = 1
I = integral(f(x), x, a, b).n()
delta = (b-a)/n; rdelta = r*delta; xk = a; L = []; S = 0
for k in range(n):
    L = L + [(xk, 0)]
    y = f(xk+rdelta)
    S = S + y
    L = L + [(xk,y)]
    xk = xk + delta
    L = L + [(xk, y)]
S = delta*S.n()
L = L + [(xk,0)]
pretty_print('Actual Integral = %s'%I)
pretty_print('Approximate Riemann Sum = %s'%S)
```

(Refer Slide Time: 18:07)

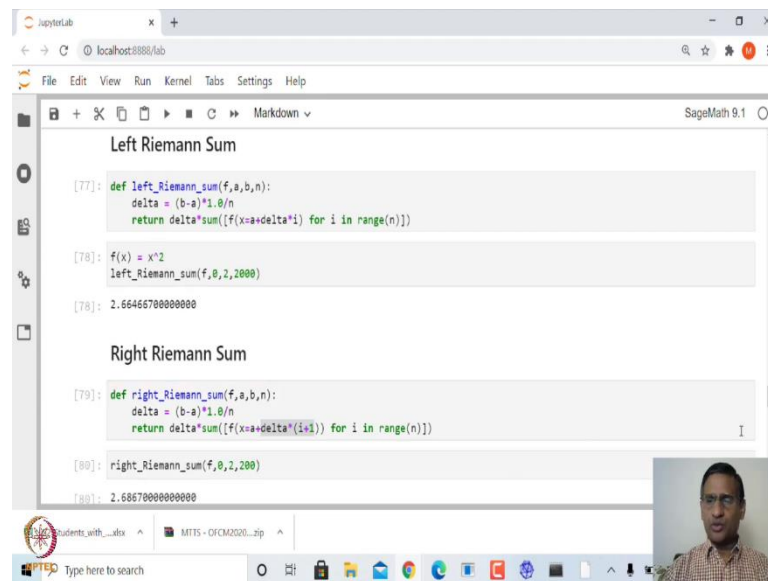


Let me just first make it a small interval, let us say 10, the way you need to plot these rectangles is first your plot, let us say this point, then plot this point, then this point, and then you again go to this point and then you go to next point and like this.

You add all these rectangle's vertices together and then at the end, you have to also add this endpoint, here 2 comma 0, in this case. And then you plot this using what is called a polygon plot. Using polygon plot you plot these and this is the graph of the function and then you are filling this graph between the function and the x-axis and here  $S$  is calculated as the Riemann Sum.

At each point, you can see here  $S$  is equal to  $S$  plus  $y$ ,  $y$  is the value of the function which is the height. And then you keep on adding the height and then at the end,  $S$  is multiplied by the width of each interval.

(Refer Slide Time: 19:18)



```

Left Riemann Sum

[77]: def left_riemann_sum(f,a,b,n):
      delta = (b-a)*1.0/n
      return delta*sum([f(x+a+delta*i) for i in range(n)])

[78]: f(x) = x^2
      left_riemann_sum(f,0,2,2000)

[78]: 2.6646670000000000

Right Riemann Sum

[79]: def right_riemann_sum(f,a,b,n):
      delta = (b-a)*1.0/n
      return delta*sum([f(x+a+delta*(i+1)) for i in range(n)])

[80]: right_riemann_sum(f,0,2,200)

[80]: 2.6867000000000000

```

You can also create animation for this using the sage interact function. You can also create a user-defined function for left Riemann sum, this is quite simple. Similarly for right Riemann sum and midpoint Riemann sum. If it is left Riemann sum, you supply the value of the function  $f$ , the endpoints  $a$  and  $b$  and the number of intervals, and this is the delta. Then for each thing, you are calculating the height and then ultimately at the end you multiply by the length of each interval

If you take the function  $f(x)$  equal to  $x$  square and try to find the left Riemann sum this is 2.66, the integral of  $x$  square will be  $x$  cube by 3. From 0 to 2, that is 8 by 3 which is quite close to 2.6666. Similarly, if you take the lower right Riemann sum, then you just need to define this as  $x$  plus delta into  $i$  plus. And again, you can call, this let us say instead of 2000, let me just have 200 points. This is also very close to this.

(Refer Slide Time: 20:41)

The image shows a JupyterLab window with a SageMath 9.1 kernel. The code defines a function `midpoint_Riemann_sum(f, a, b, n)` that calculates the midpoint Riemann sum for a function `f` over the interval `[a, b]` with `n` subintervals. The function calculates the width of each subinterval `delta = (b-a)/n`, creates a list of midpoints `xs`, and then sums the function values at these midpoints multiplied by `delta`. The code is executed, and the output is `2.666500000000000`. Below the code, there is an interactive widget with input boxes for `f`, `a`, `b`, and `n`, and a dropdown menu for the method (Left, Right, Middle).

```
[81]: def midpoint_Riemann_sum(f,a,b,n):
      delta = (b-a)*1.0/n
      xs=[a+delta*i for i in range(n+1)]
      ysmid=[f(x)=(xs[i]+xs[i+1])/2) for i in range(n)]
      return delta*sum(ysmid)

[82]: midpoint_Riemann_sum(f,0,2,200)

[82]: 2.666500000000000
```

```
[ ]: @interact
      def _ (f=input_box(default=x^3-5*x+2), a=input_box(default=-1), b=input_box(default=2),
            n=input_box(default=15), Method=['Left', 'Right', 'Middle']):
            if Method=='Left':
                t=0
```

Similarly, the midpoint rule, you can call this function again, let me call only with 200 points. The midpoint rule will be much closer to the actual value. This is how you can approximate the integral from  $a$  to  $b$  of  $f(x) dx$ .

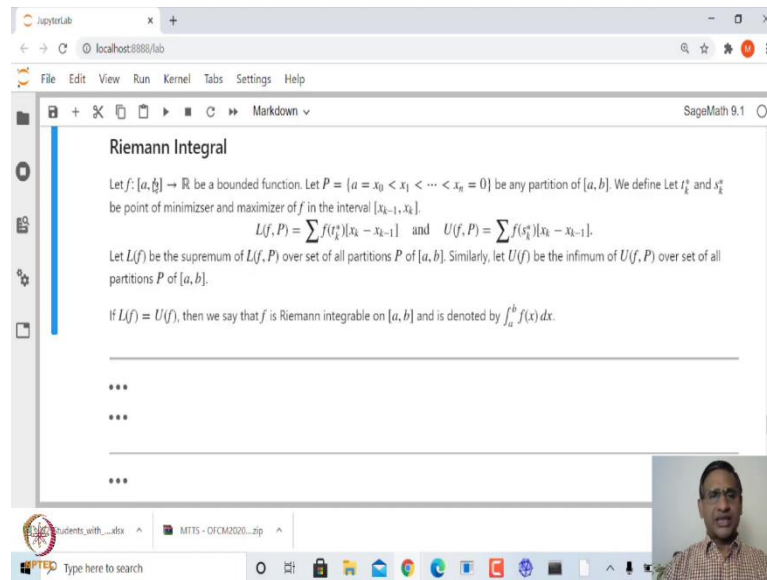
(Refer Slide Time: 21:13)

The image shows a JupyterLab window with a SageMath 9.1 kernel. The code defines a function `f(x)=f` and calculates the integral `I = integral(f(x), x, a, b).n()`. It then calculates the width of each subinterval `delta = (b-a)/n` and the total width `tdelta = t*delta`. It creates a list of midpoints `xs` and a list of function values `ysmid`. It then sums the function values at these midpoints multiplied by `delta` to get the approximate Riemann sum `S`. The code is executed, and the output is `2.666500000000000`. Below the code, there is an interactive widget with input boxes for `f`, `a`, `b`, and `n`, and a dropdown menu for the method (Left, Right, Middle).

```
t=0.5
f(x)=f
I = integral(f(x), x, a, b).n()
delta = (b-a)/n; tdelta = t*delta; xk = a; L = []; S = 0
for k in range(n):
    L = L + [(xk, 0)]
    y = f(xk+tdelta)
    S = S + y
    L = L + [(xk,y)]
    xk = xk + delta
    L = L + [(xk, y)]
S = delta*S.n()
pretty_print('Actual Integral = %s'%I)
pretty_print('Approximate Riemann Sum = %s'%S)
L = L + [(xk,0)]
G = plot(f(x), (x, a-(b-a)/20, b+(b-a)/20), color = 'red', thickness = 1)
G = G + plot(f(x), (x, a, b), color = 'red', thickness = 1, fill = true, fillcolor = 'grey')
G = G + polygon(L, edgecolor = 'black', color='lightblue')
G.show(aspect_ratio = 'automatic')
```

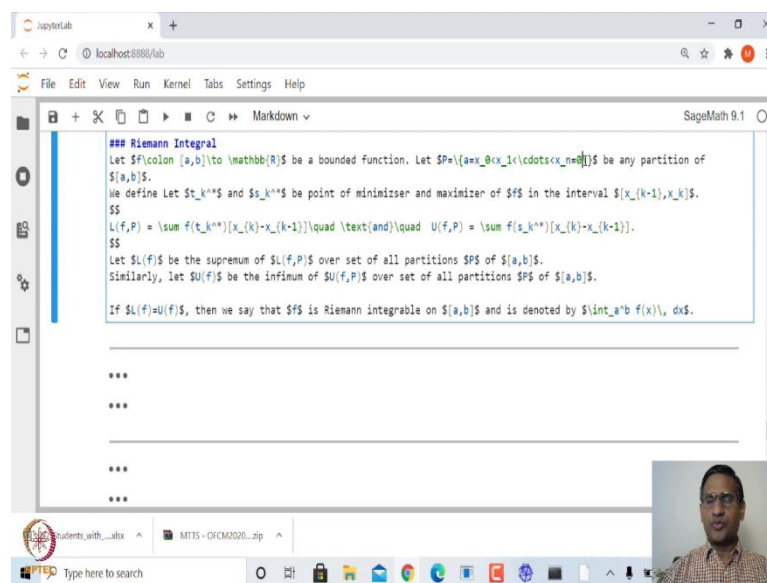
I will skip this because, all you need to do is just put add `@interact` and put all the codes that we have seen earlier. This will create an animation where you can change the function, you can change  $a$  and  $b$ , you can change the number of points. All these things will be interactive.

(Refer Slide Time: 21:34)



The way one defines the Riemann integral is, as I said earlier, instead of left and right endpoints, you just take  $t_k^*$  where  $t_k^*$  is the point at which the function has a minimum in the interval  $x_{k-1}$  to  $x_k$ .

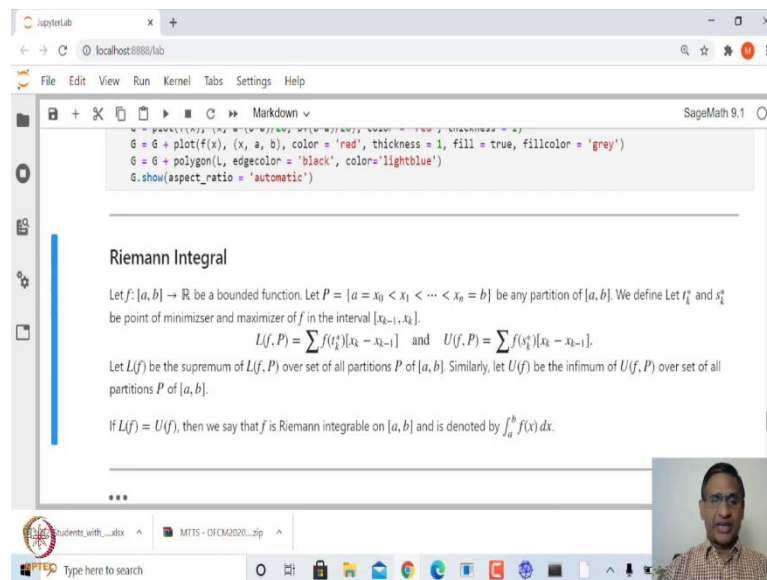
(Refer Slide Time: 21:55)



Where you take a partition P, P is any partition of this interval a to b.

(Refer Slide Time: 22:11)





You take any partition and then define what is called the lower Riemann sum. Similarly, instead of  $t_k^*$  star if we take the point where the maximum of the function occurs. The maximum and minimum will occur because we are considering  $f$  to be bounded. For the existence of integral, we assume that  $f$  is bounded on the interval  $[a, b]$ .

And so you consider this  $L(f, P)$  and  $U(f, P)$  for each of this partition and take all such left Riemann sum and right Riemann sum for all the possible partition. Then you define what is called supremum  $L(f)$  of  $L(f, P)$ .

One can show that this supremum exists because this will be bounded above by  $b$  minus  $a$  into the maximum value of the function in the interval  $[a, b]$ . And similarly, this will be bounded below. It will have supremum of  $L(f, P)$  and infimum of  $U(f, P)$ , you find out the infimum and supremum of these 2. In case these 2 supremum and infimum, which were we are calling as  $L(f)$  and  $U(f)$  agree to each other, they are the same, then we say that  $f$  is Riemann integrable and the value of  $L(f)$  is the Riemann integral is  $\int_a^b f(x) dx$ . That is the notation. Riemann integral this is how we denote. This was given by Riemann which is why it is called Riemann integral.

But the here problem is that to find this  $f(t_k^*)$  and  $f(s_k^*)$  will be a somewhat tedious process. But in case the integral exists, then you can divide this interval into, let us say  $n$  equal parts and you take  $n$  very large. Then the value of this Riemann sum will be close to the actual integral. That is how one defines the Riemann integral of a function.



(Refer Slide Time: 24:21)

The screenshot shows a JupyterLab interface with a SageMath 9.1 kernel. The main content area displays the title "Fundamental Theorem of Calculus" followed by three numbered points:

1. **(First Fundamental Theorem of Calculus)** Let  $f: [a, b] \rightarrow \mathbb{R}$  be differentiable function and that  $f'$  is integrable on  $[a, b]$ . Then
$$\int_a^b f'(x) dx = f(b) - f(a).$$
2. Let  $f: [a, b] \rightarrow \mathbb{R}$  be integrable on  $[a, b]$ . For  $x \in [a, b]$  define
$$F(x) := \int_a^x f(t) dx$$

$F$  is called the indefinite integral of  $f$ .
3. **(Second Fundamental Theorem of Calculus)** If  $f: [a, b] \rightarrow \mathbb{R}$  is integrable function, then the indefinite integral  $F(x)$  defined in 1 is continuous on  $[a, b]$  and is differentiable at  $x$  if  $f$  is continuous at  $x$  with  $F'(x) = f(x)$ .

The interface includes a file explorer on the left, a top menu bar (File, Edit, View, Run, Kernel, Tabs, Settings, Help), and a bottom taskbar with various application icons.

Next, let us look at the fundamental theorem of integral calculus. The integral which we find is based on these two fundamental theorems. It says that, in case the function is differentiable and at the same time  $f'$  is integrable, Riemann integrable, then integral of  $f' \times dx$  from  $a$  to  $b$  will be nothing, but  $f(b)$  minus  $f(a)$ . That is how you find the integral.

To find this integral, you need a function whose derivative is the actual function and then you take the difference of the value function derivative of that function whose derivative is the integrand then you look at the difference.

(Refer Slide Time: 25:19)

This screenshot is identical to the one above, showing the JupyterLab interface with the same content for the Fundamental Theorem of Calculus. It details the First and Second Fundamental Theorems, including the integral formula  $\int_a^b f'(x) dx = f(b) - f(a)$  and the definition of the indefinite integral  $F(x) := \int_a^x f(t) dt$ .

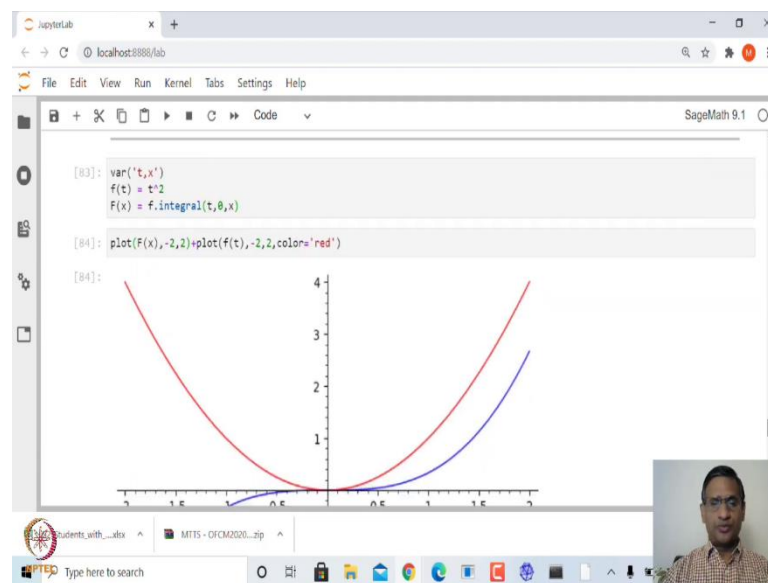
Similarly, if you define, let us say, if the function is integrable, you define integral of  $f(t)$  from  $a$  to  $x$   $dt$ , that is, the variable with respect to which we are integrating. This is the function of  $x$ . What actually we

are doing? We are taking any  $x$  and we are finding the area under this curve  $f(t)$  from  $a$  to  $x$ . This function  $F(x)$  is called antiderivative or indefinite integral of  $f$ .

This will be differentiable, in case  $f$  is continuous at  $x$ . Then capital  $F$  will be differentiable at  $x$  and not only that the derivative of  $F(x)$  is  $f(x)$ . That is what is called the second fundamental theorem of calculus.

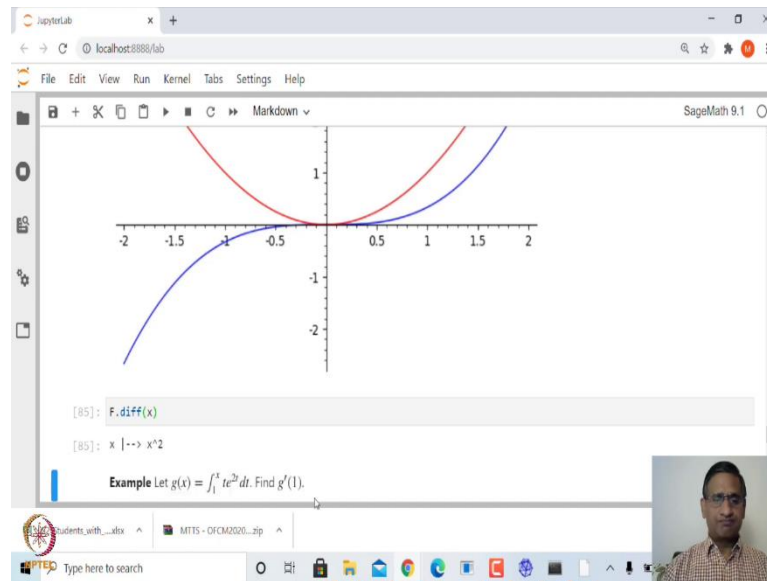
If you define this indefinite integral of  $f$ , take the derivative that is what you get capital  $F$  dash  $x$  is small  $f(x)$ . This is called the second fundamental theorem of integral calculus. This is called the first fundamental theorem of integral calculus. The calculation of this Riemann integral is actually based on these two facts.

(Refer Slide Time: 26:30)



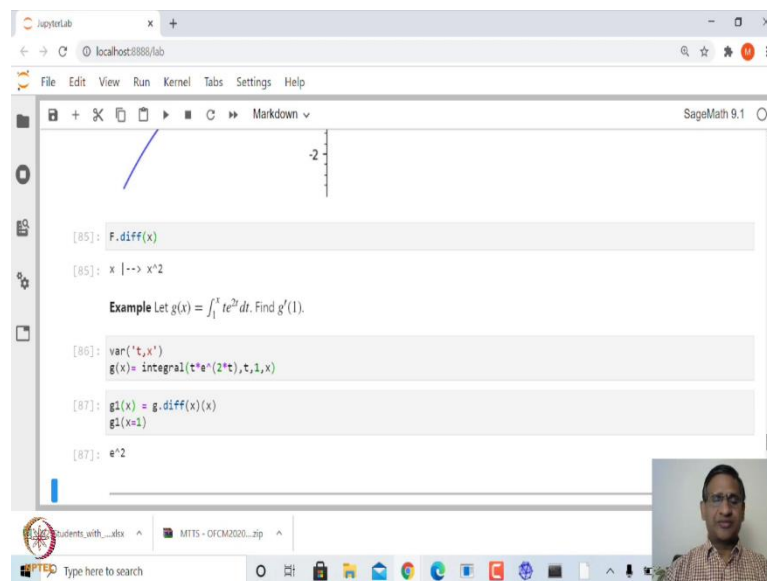
Let us look at a small example. Suppose I have a function  $f(t)$  is equal to  $t$  square and if I find the integral from  $t$  to  $x$ . I call this  $F(x)$  and if I look at what is the graph of this function  $F(x)$  along with the graph of an  $f(x)$ . This is how you see; this red curve is a graph of  $F(x)$ . And the blue one is a graph of derivative, a graph of this indefinite integral of  $f$ . You can see this is the parabola, the graph we have taken as  $f(t)$  equal to  $t$  square and this is you what you are getting is cubic.

(Refer Slide Time: 27:12)



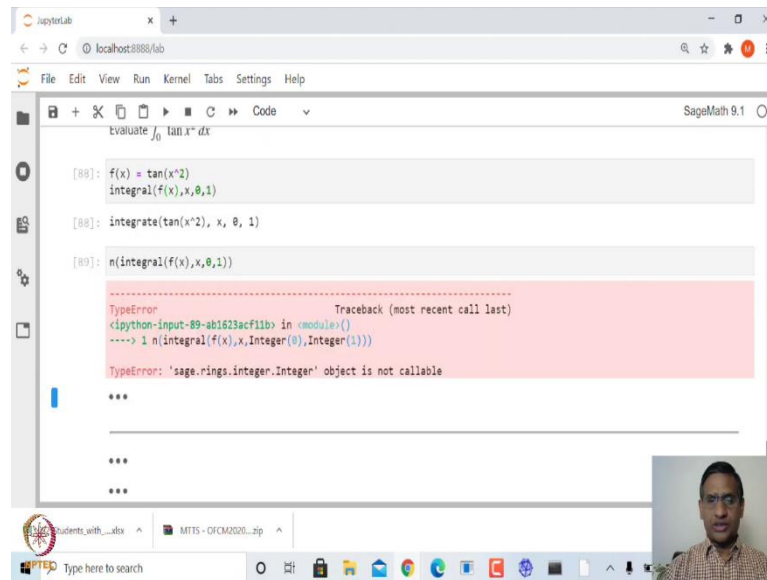
If I try to find the derivative of capital  $F(x)$  with respect to  $x$  you should get  $x$  square. That is what you get. Similarly, you can take another example. This is somewhat more complicated.  $f(x) = xe^{2x}$ , and let us define this indefinite integral as  $g(x)$  from 1 to  $x$  and then find what is derivative of  $g$  at 1.

(Refer Slide Time: 27:38)



This is again very simple. Define  $t$  and  $x$  as variables, define  $g(x)$  as indefinite integral and then find the derivative of  $g$ . Let me call this as  $g1$  and evaluate  $g1$  at 1 which is  $e$  to the power 2. That is the verification of this fundamental theorem of integral calculus.

(Refer Slide Time: 28:02)



```
evaluate  $\int_0^1 \tan x^2 dx$ 

[88]: f(x) = tan(x^2)
      integral(f(x), x, 0, 1)

[89]: integrate(tan(x^2), x, 0, 1)

[90]: n(integral(f(x), x, 0, 1))

-----
TypeError                                 Traceback (most recent call last)
<ipython-input-89-ab1623acf11b> in <module>()
----> 1 n(integral(f(x), x, Integer(0), Integer(1)))

TypeError: 'sage.rings.integer.Integer' object is not callable

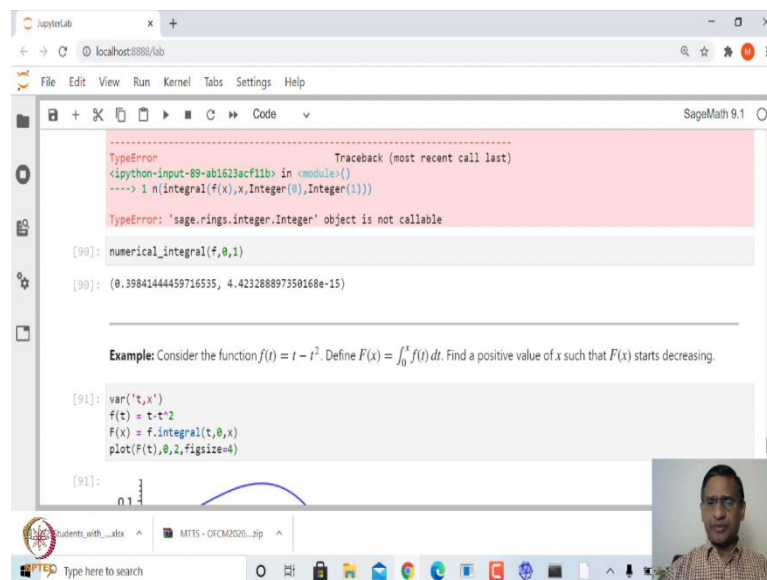
***

***

***
```

Suppose you need to define, for example, the integral of  $\tan(x)$  square from 0 to 1. Again, you can find this integral using `integral`, but in this case, you see that this does not give you the integral. It simply returns, whatever you have written. So that means, for this integral Sage is unable to find this integral in closed form. In that case, what you can do is, you can find the numerical integral.

(Refer Slide Time: 28:42)



```
-----
TypeError                                 Traceback (most recent call last)
<ipython-input-89-ab1623acf11b> in <module>()
----> 1 n(integral(f(x), x, Integer(0), Integer(1)))

TypeError: 'sage.rings.integer.Integer' object is not callable

[90]: numerical_integral(f, 0, 1)

[90]: (0.39841444459716535, 4.423288897359168e-15)

Example: Consider the function  $f(t) = t - t^2$ . Define  $F(x) = \int_0^x f(t) dt$ . Find a positive value of  $x$  such that  $F(x)$  starts decreasing.

[91]: var('t,x')
      f(t) = t-t^2
      F(x) = f.integral(t, 0, x)
      plot(F(t), 0, 2, figsize=4)

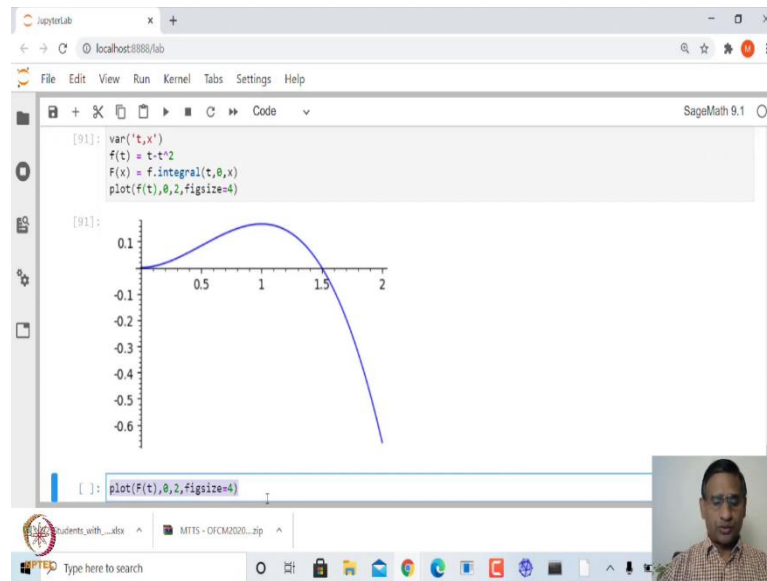
[91]:
```

If you say this `n` will not work. What it says is this is not callable so you need to find this as numerical integral. In this case, you will get the value. In this case, numerical integral is 0.3984 and with the error of the order 10 to the power minus 15. Many times, if you have the definite integral Sage may not be able to find it in closed form and therefore, in that case, you need to find the numerical integral.

Similarly, let us take another example. Suppose you have function  $f(t)$  is equal to  $t$  minus  $t$  square and define this anti-derivative  $F(x)$  is equal to 0 to  $x$   $f(t) dt$  and find the positive value of  $x$  such that  $F(x)$  starts

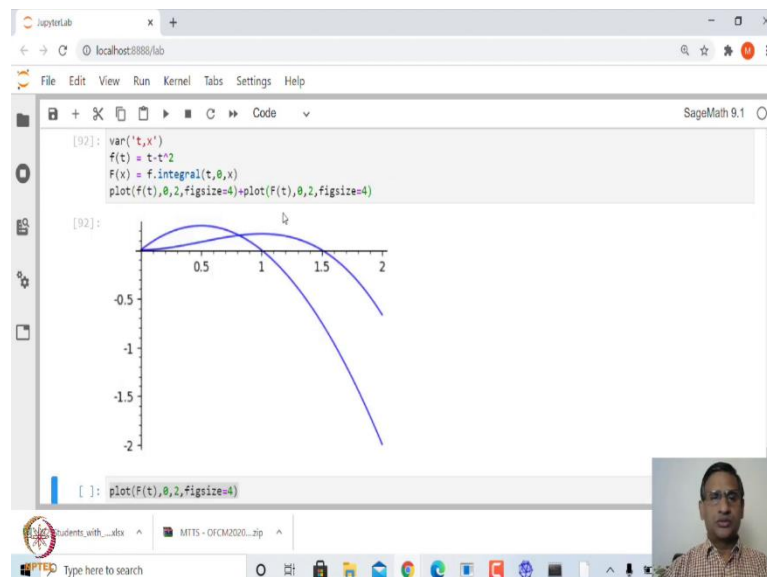
decreasing. That is the problem. It is quite simple. First, let us plot a graph of the function  $f$  and graph of indefinite integral  $F(x)$  and then see what happen.

(Refer Slide Time: 29:37)

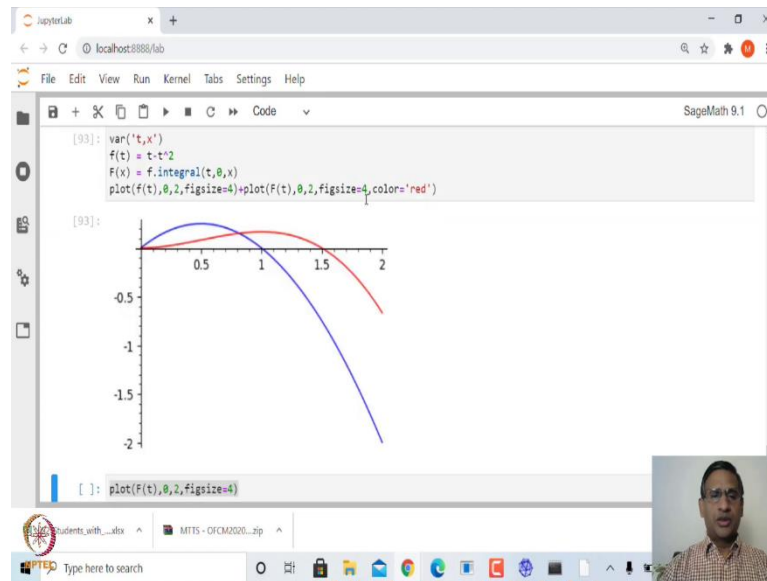


This is the graph of the function  $F$  and you can here when does it start decreasing. You can see here this is the point when it starts decreasing. And suppose we also plot the graph of the function this should be let us plot this  $f$  and  $F$  together so that we can compare.

(Refer Slide Time: 30:21)



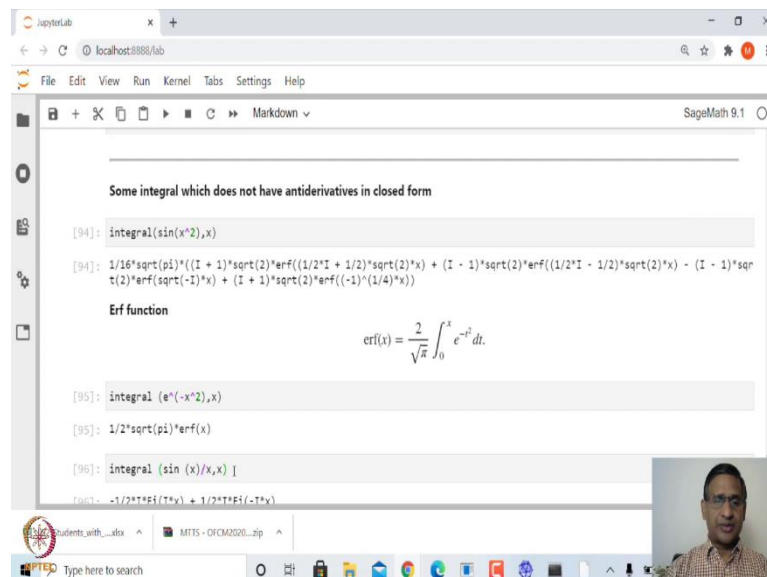
(Refer Slide Time: 30:27)



Let me put this in red, colour equal to red. The blue one is a graph of the function  $f$ , and the red one is a graph of indefinite integral capital  $F(x)$ . You can see here, this at this point from which  $f$  goes to negative. Before that, the indefinite integral, that is, capital  $F(x)$  will go on increasing.

Whereas the moment the function becomes negative, then the capital  $F(x)$  will start decreasing. Here it is at 1, the capital  $F(x)$  will start decreasing.

(Refer Slide Time: 31:26)



We saw that some integral does not have closed-form of antiderivative in that case you need to find numerical integral. Similarly, if you try to find out, for example, the integral of  $\sin x$  square, again this integral of  $\sin x$  square gives you something in terms of erf, that is called error function and this error function, is defined as  $\text{erf } x$  is equal to  $2$  upon square root  $\pi$  integral form of  $e$  to the power minus  $t$  square  $0$  to  $x$  square.

Those who have done a little bit of statistics would be aware of this function. This integral is given in this error form, of course, you can find what is integral of  $e$  to the power minus  $x$  square with respect to  $x$ . It will be error function times square root  $\pi$  by 2.

That this scaling factor is coming here and if you want to find integral of such function. For example, the integral of  $\sin(x)$  upon  $x$  gives you in terms of  $Ei$  of  $x$ . Again, you can look at what is this  $Ei$  function. Often, the integral of some of the standard function is defined in terms of some other function, either error function or  $Ei$   $x$ .

(Refer Slide Time: 32:50)

```
[94]: integral(sin(x^2),x)
[94]: 1/16*sqrt(pi)*((1 + 1)*sqrt(2)*erf((1/2*I + 1/2)*sqrt(2)*x) + (1 - 1)*sqrt(2)*erf((1/2*I - 1/2)*sqrt(2)*x) - (1 - 1)*sqrt(2)*erf(sqrt(-1)*x) + (1 + 1)*sqrt(2)*erf((-1)^(1/4)*x))
Erf function

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt.$$

[95]: integral(e^(-x^2),x)
[95]: 1/2*sqrt(pi)*erf(x)
[97]: integral(sin(x)/x,x,n())
TypeError: Traceback (most recent call last)
/opt/sagemath-9.1/local/lib/python3.7/site-packages/sage/libs/mpmath/utils.pyx in sage.libs.mpmath.utils.sage_to_mpmath
(build/cythonized/sage/libs/mpmath/utils.c:5988)()
327 else:
--> 328 x = RealField(prec)(x)
```

You can find numerical values of these things. For example, here, in this case, I can say dot  $n$  is not giving, we have seen already because this is the indefinite integral, dot not makes sense.

(Refer Slide Time: 33:04)

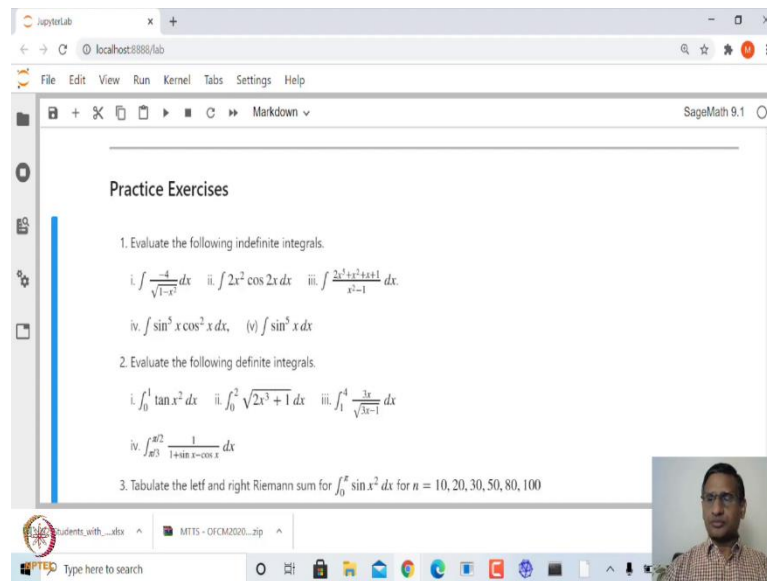
```
[94]: integral(sin(x^2),x)
[94]: 1/16*sqrt(pi)*((1 + 1)*sqrt(2)*erf((1/2*I + 1/2)*sqrt(2)*x) + (1 - 1)*sqrt(2)*erf((1/2*I - 1/2)*sqrt(2)*x) - (1 - 1)*sqrt(2)*erf(sqrt(-1)*x) + (1 + 1)*sqrt(2)*erf((-1)^(1/4)*x))
Erf function

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt.$$

[95]: integral(e^(-x^2),x)
[95]: 1/2*sqrt(pi)*erf(x)
[98]: integral(sin(x)/x,x,0,oo)
[98]: 1/2*pi
```

Whereas if I put this limit 0 to for example, infinity then this integral is pi by 2.

(Refer Slide Time: 33:13)



The screenshot shows a JupyterLab window with a SageMath 9.1 notebook. The notebook is titled "Practice Exercises" and contains three sections of problems:

1. Evaluate the following indefinite integrals.
  - i.  $\int \frac{4}{\sqrt{1-x^2}} dx$
  - ii.  $\int 2x^2 \cos 2x dx$
  - iii.  $\int \frac{2x^3 + x^2 + 1}{x^2 - 1} dx$
  - iv.  $\int \sin^5 x \cos^3 x dx$
  - v.  $\int \sin^3 x dx$
2. Evaluate the following definite integrals.
  - i.  $\int_0^1 \tan x^2 dx$
  - ii.  $\int_0^2 \sqrt{2x^3 + 1} dx$
  - iii.  $\int_1^4 \frac{3x}{\sqrt{5x-1}} dx$
  - iv.  $\int_{\pi/2}^{\pi} \frac{1}{1 + \sin x - \cos x} dx$
3. Tabulate the left and right Riemann sum for  $\int_0^{\pi} \sin x^2 dx$  for  $n = 10, 20, 30, 50, 80, 100$

The interface includes a file explorer on the left, a top menu bar (File, Edit, View, Run, Kernel, Tabs, Settings, Help), and a bottom status bar with a search field and system icons. A small video feed of the presenter is visible in the bottom right corner.

Let me leave you with some easy exercises of computing definite and indefinite integrals. Some of them are quite straightforward, some of them are not so straight forward, when it comes to calculating by hand. The last problem is that if you look at this sin of x square, we saw that this integral is not in closed form. But you just find out left and right Riemann sum. We have already defined what is the function and tabulate this value for n is equal to 10, 20, 30, 50, 80, 100. This is just to demonstrate that this integral will actually converge to the actual integral.

Thank you very much, we will look at some more examples in the next class.