Lab 11

Kate Sperry, Jaylen Reese

Problem 1

A and B represent the limits of the function. N represents how many partitions the function is split up by. Lastly, I represents the index.

In [2]:
```
def f(x): return sqrt(2*x)
a, b = 1,5
n = 4
```

In [3]:
```
print(list(range(n)))
print([i/2 + 1 for i in range(n)])
```

```
[0, 1, 2, 3]
[1, 3/2, 2, 5/2]
```

In [4]:
```
x_n = [a + (b-a)/n*i for i in range(n+1)]
print(x_n)
```

```
[1, 2, 3, 4, 5]
```

In [5]:
```
fx_n = [f(i).n() for i in x_n]
print(fx_n)
```

```
[1.41421356237310, 2.00000000000000, 2.44948974278318, 2.82842712474619,
3.16227766016838]
```

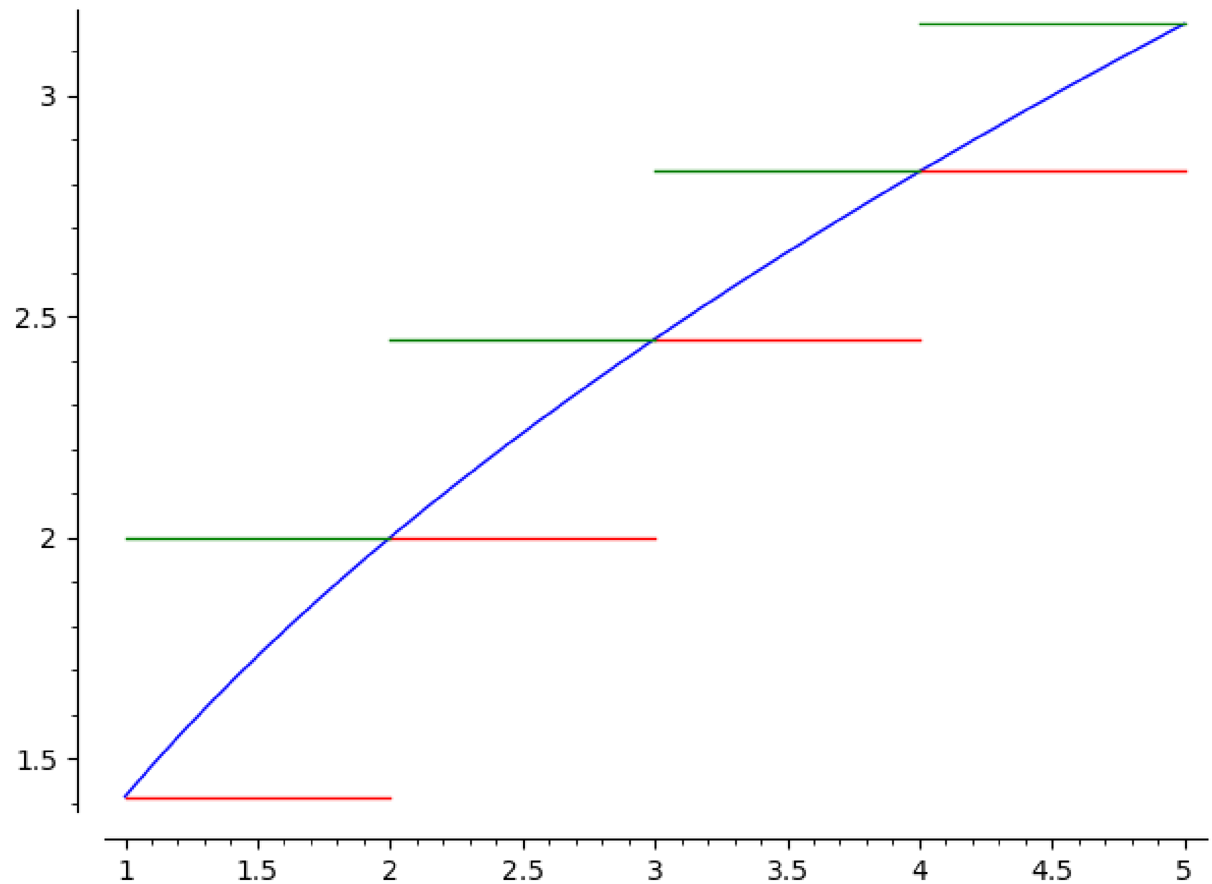
In [6]:
```
print("Left endpoints: ", fx_n[:-1])
print("Right endpoints: ", fx_n[1:])
```

```
('Left endpoints: ', [1.41421356237310, 2.00000000000000, 2.4494897427831
8, 2.82842712474619])
('Right endpoints: ', [2.00000000000000, 2.44948974278318, 2.828427124746
19, 3.16227766016838])
```

In [7]:
```
L = (b-a)/n * sum(fx_n[:-1])
R = (b-a)/n * sum(fx_n[1:])
print("Left Approximation: ", L)
print("Right Approximation: ", R)
```

```
('Left Approximation: ', 8.69213042990246)
('Right Approximation: ', 10.4401945276977)
```

In [8]:
```
P_f = plot(f(x), x, a, b)
P_L = sum([plot(fx_n[i], x, x_n[i], x_n[i+1], color = "red") for i in range
P_R = sum([plot(fx_n[i+1], x, x_n[i], x_n[i+1], color = "green") for i in r
show(P_f + P_L + P_R)
```



Problem 2

In [4]:
```
def f(t): return 100*(1-(e)^(-0.1*t))
a, b = 5,50
n = 10
```

In [5]:
```
print(list(range(n)))
print([i/2 + 1 for i in range(n)])
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
[1, 3/2, 2, 5/2, 3, 7/2, 4, 9/2, 5, 11/2]
```

In [6]:
```
t_n = [a + (b-a)/n*i for i in range(n+1)]
print(t_n)
```

```
[5, 19/2, 14, 37/2, 23, 55/2, 32, 73/2, 41, 91/2, 50]
```

In [7]:
```
ft_n = [f(i).n() for i in t_n]
print(ft_n)
```

```
[39.3469340287367, 61.3258976545499, 75.3403036058394, 84.2762833686372,
89.9741156277196, 93.6072138793292, 95.9237796021634, 97.4008871221245, 9
8.3427324598239, 98.9432795616147, 99.3262053000915]
```

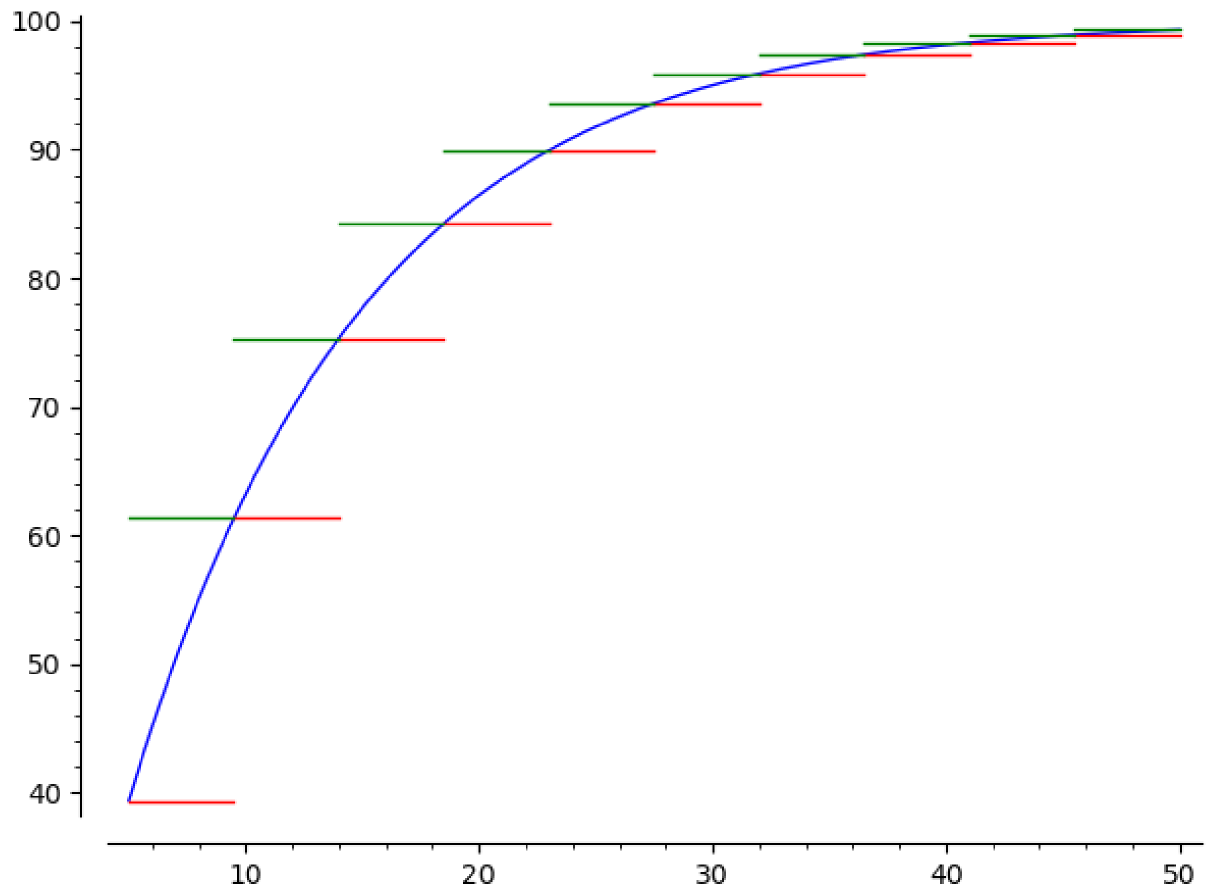
In [8]:
```
print("Left endpoints: ", ft_n[:-1])
print("Right endpoints: ", ft_n[1:])
```

```
('Left endpoints: ', [39.3469340287367, 61.3258976545499, 75.340303605839
4, 84.2762833686372, 89.9741156277196, 93.6072138793292, 95.923779602163
4, 97.4008871221245, 98.3427324598239, 98.9432795616147])
('Right endpoints: ', [61.3258976545499, 75.3403036058394, 84.27628336863
72, 89.9741156277196, 93.6072138793292, 95.9237796021634, 97.400887122124
5, 98.3427324598239, 98.9432795616147, 99.3262053000915])
```

In [9]:
```
L = (b-a)/n * sum(ft_n[:-1])
R = (b-a)/n * sum(ft_n[1:])
print("Left Approximation: ", L)
print("Right Approximation: ", R)
```

```
('Left Approximation: ', 3755.16642109742)
('Right Approximation: ', 4025.07314181852)
```

In [10]:
```
t=var('t')
P_f = plot(f(t), t, a, b)
P_L = sum([plot(ft_n[i], t, t_n[i], t_n[i+1], color = "red") for i in range
P_R = sum([plot(ft_n[i+1], t, t_n[i], t_n[i+1], color = "green") for i in r
show(P_f + P_L + P_R)
```



In [11]:
```
def f(t): return 100*(1-(e)^(-0.1*t))
a, b = 5,50
n = 100
```

In [12]:
```
print(list(range(n)))
print([i/2 + 1 for i in range(n)])
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 2
0, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 3
8, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 5
6, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 7
4, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 9
2, 93, 94, 95, 96, 97, 98, 99]
[1, 3/2, 2, 5/2, 3, 7/2, 4, 9/2, 5, 11/2, 6, 13/2, 7, 15/2, 8, 17/2, 9, 1
9/2, 10, 21/2, 11, 23/2, 12, 25/2, 13, 27/2, 14, 29/2, 15, 31/2, 16, 33/
2, 17, 35/2, 18, 37/2, 19, 39/2, 20, 41/2, 21, 43/2, 22, 45/2, 23, 47/2,
24, 49/2, 25, 51/2, 26, 53/2, 27, 55/2, 28, 57/2, 29, 59/2, 30, 61/2, 31,
63/2, 32, 65/2, 33, 67/2, 34, 69/2, 35, 71/2, 36, 73/2, 37, 75/2, 38, 77/
2, 39, 79/2, 40, 81/2, 41, 83/2, 42, 85/2, 43, 87/2, 44, 89/2, 45, 91/2,
46, 93/2, 47, 95/2, 48, 97/2, 49, 99/2, 50, 101/2]
```

In [14]:
```
t_n = [a + (b-a)/n*i for i in range(n+1)]
print(t_n)
```

```
[5, 109/20, 59/10, 127/20, 34/5, 29/4, 77/10, 163/20, 43/5, 181/20, 19/2,
199/20, 52/5, 217/20, 113/10, 47/4, 61/5, 253/20, 131/10, 271/20, 14, 28
9/20, 149/10, 307/20, 79/5, 65/4, 167/10, 343/20, 88/5, 361/20, 37/2, 37
9/20, 97/5, 397/20, 203/10, 83/4, 106/5, 433/20, 221/10, 451/20, 23, 469/
20, 239/10, 487/20, 124/5, 101/4, 257/10, 523/20, 133/5, 541/20, 55/2, 55
9/20, 142/5, 577/20, 293/10, 119/4, 151/5, 613/20, 311/10, 631/20, 32, 64
9/20, 329/10, 667/20, 169/5, 137/4, 347/10, 703/20, 178/5, 721/20, 73/2,
739/20, 187/5, 757/20, 383/10, 155/4, 196/5, 793/20, 401/10, 811/20, 41,
829/20, 419/10, 847/20, 214/5, 173/4, 437/10, 883/20, 223/5, 901/20, 91/
2, 919/20, 232/5, 937/20, 473/10, 191/4, 241/5, 973/20, 491/10, 991/20, 5
0]
```

In [15]:
```
ft_n = [f(i).n() for i in t_n]
print(ft_n)
```

```
[39.3469340287367, 42.0158216660154, 44.5672715265493, 47.0064511682431,
49.3383007634410, 51.5675431044638, 53.6986931688772, 55.7360672638649, 5
7.6837917682251, 59.5458114896981, 61.3258976545499, 63.0276555455941, 6
4.6545318041220, 66.2098214105287, 67.6966743577747, 69.1181020311980, 7
0.4769833075986, 71.7760703859477, 73.0179943615313, 74.2052705548174, 7
5.3403036058394, 76.4253923444137, 77.4627344460561, 78.4544308830258, 7
9.4024901795117, 80.3088324795806, 81.1752934361253, 82.0036279286888, 8
2.7955136176950, 83.5525543422845, 84.2762833686372, 84.9681664953600, 8
5.6296050222297, 86.2619385883046, 86.8664478851507, 87.4443572506803, 8
7.9968371488543, 88.5250065402721, 89.0299351484489, 89.5126456263713, 8
9.9741156277196, 90.4152797869501, 90.8370316122495, 91.2402252951943, 9
1.6256774407804, 91.9941687213280, 92.3464454576089, 92.6832211304018, 9
3.0051778255345, 93.3129676153406, 93.6072138793292, 93.8885125667412, 9
4.1574334035499, 94.4145210463515, 94.6602961854803, 94.8952565995846, 9
5.1198781637987, 95.3346158135528, 95.5399044659725, 95.7361599007347, 9
5.9237796021634, 96.1031435642715, 96.2746150603784, 96.4385413788628, 9
6.5952545265401, 96.7450719010897, 96.8882969338939, 97.0252197045902, 9
7.1561175285815, 97.2812555186947, 97.4008871221245, 97.5152546337510, 9
7.6245896868695, 97.7291137223269, 97.8290384370151, 97.9245662126300, 9
8.0158905255630, 98.1031963387570, 98.1866604763199, 98.2664519816534, 9
8.3427324598239, 98.4156564048679, 98.4853715126953, 98.5520189802240, 9
8.6157337913520, 98.6766449903459, 98.7348759431995, 98.7905445874922, 9
8.8437636712531, 98.8946409813141, 98.9432795616147, 98.9897779219021, 9
9.0342302372462, 99.0767265387768, 99.1173528960273, 99.1561915912549, 9
9.1933212860900, 99.2288171808537, 99.2627511668632, 99.2951919720368, 9
9.3262053000915]
```

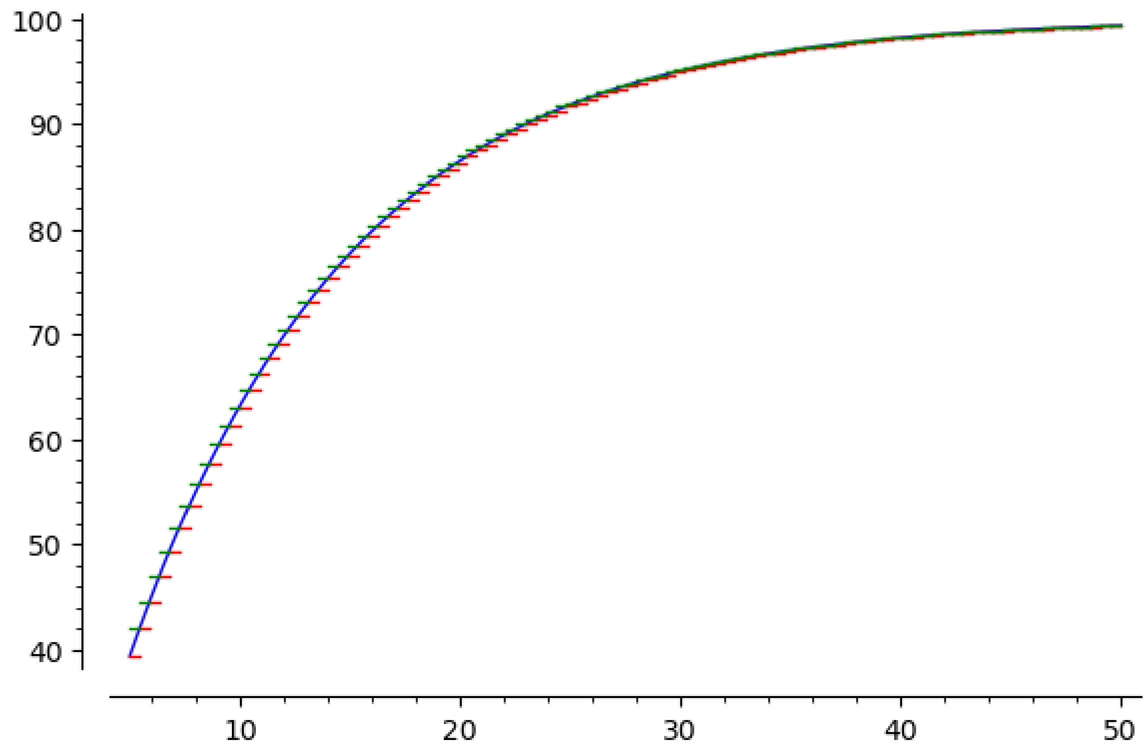
In [16]:
```
print("Left endpoints: ", ft_n[:-1])
print("Right endpoints: ", ft_n[1:])
```

```
('Left endpoints: ', [39.3469340287367, 42.0158216660154, 44.567271526549
3, 47.0064511682431, 49.3383007634410, 51.5675431044638, 53.698693168877
2, 55.7360672638649, 57.6837917682251, 59.5458114896981, 61.325897654549
9, 63.0276555455941, 64.6545318041220, 66.2098214105287, 67.696674357774
7, 69.1181020311980, 70.4769833075986, 71.7760703859477, 73.017994361531
3, 74.2052705548174, 75.3403036058394, 76.4253923444137, 77.462734446056
1, 78.4544308830258, 79.4024901795117, 80.3088324795806, 81.17529343612
53, 82.0036279286888, 82.7955136176950, 83.5525543422845, 84.276283368637
2, 84.9681664953600, 85.6296050222297, 86.2619385883046, 86.866447885150
7, 87.4443572506803, 87.9968371488543, 88.5250065402721, 89.029935148448
9, 89.5126456263713, 89.9741156277196, 90.4152797869501, 90.837031612249
5, 91.2402252951943, 91.6256774407804, 91.9941687213280, 92.346445457608
9, 92.6832211304018, 93.0051778255345, 93.3129676153406, 93.607213879329
2, 93.8885125667412, 94.1574334035499, 94.4145210463515, 94.660296185480
3, 94.8952565995846, 95.1198781637987, 95.3346158135528, 95.539904465972
5, 95.7361599007347, 95.9237796021634, 96.1031435642715, 96.274615060378
4, 96.4385413788628, 96.5952545265401, 96.7450719010897, 96.888296933893
9, 97.0252197045902, 97.1561175285815, 97.2812555186947, 97.400887122124
5, 97.5152546337510, 97.6245896868695, 97.7291137223269, 97.829038437015
1, 97.9245662126300, 98.0158905255630, 98.1031963387570, 98.186660476319
9, 98.2664519816534, 98.3427324598239, 98.4156564048679, 98.485371512695
3, 98.5520189802240, 98.6157337913520, 98.6766449903459, 98.734875943199
5, 98.7905445874922, 98.8437636712531, 98.8946409813141, 98.943279561614
7, 98.9897779219021, 99.0342302372462, 99.0767265387768, 99.117352896027
3, 99.1561915912549, 99.1933212860900, 99.2288171808537, 99.262751166863
2, 99.2951919720368])
('Right endpoints: ', [42.0158216660154, 44.5672715265493, 47.00645116824
31, 49.3383007634410, 51.5675431044638, 53.6986931688772, 55.736067263864
9, 57.6837917682251, 59.5458114896981, 61.3258976545499, 63.027655545594
1, 64.6545318041220, 66.2098214105287, 67.6966743577747, 69.118102031198
0, 70.4769833075986, 71.7760703859477, 73.0179943615313, 74.205270554817
4, 75.3403036058394, 76.4253923444137, 77.4627344460561, 78.454430883025
8, 79.4024901795117, 80.3088324795806, 81.1752934361253, 82.003627928688
8, 82.7955136176950, 83.5525543422845, 84.2762833686372, 84.968166495360
0, 85.6296050222297, 86.2619385883046, 86.8664478851507, 87.444357250680
3, 87.9968371488543, 88.5250065402721, 89.0299351484489, 89.512645626371
3, 89.9741156277196, 90.4152797869501, 90.8370316122495, 91.240225295194
3, 91.6256774407804, 91.9941687213280, 92.3464454576089, 92.683221130401
8, 93.0051778255345, 93.3129676153406, 93.6072138793292, 93.888512566741
2, 94.1574334035499, 94.4145210463515, 94.6602961854803, 94.895256599584
6, 95.1198781637987, 95.3346158135528, 95.5399044659725, 95.736159900734
7, 95.9237796021634, 96.1031435642715, 96.2746150603784, 96.438541378862
8, 96.5952545265401, 96.7450719010897, 96.8882969338939, 97.025219704590
2, 97.1561175285815, 97.2812555186947, 97.4008871221245, 97.515254633751
0, 97.6245896868695, 97.7291137223269, 97.8290384370151, 97.924566212630
0, 98.0158905255630, 98.1031963387570, 98.1866604763199, 98.266451981653
4, 98.3427324598239, 98.4156564048679, 98.4853715126953, 98.552018980224
0, 98.6157337913520, 98.6766449903459, 98.7348759431995, 98.790544587492
2, 98.8437636712531, 98.8946409813141, 98.9432795616147, 98.989777921902
1, 99.0342302372462, 99.0767265387768, 99.1173528960273, 99.156191591254
9, 99.1933212860900, 99.2288171808537, 99.2627511668632, 99.295191972036
8, 99.3262053000915])
```

In [17]:
```
L = (b-a)/n * sum(ft_n[:-1])
R = (b-a)/n * sum(ft_n[1:])
print("Left Approximation: ", L)
print("Right Approximation: ", R)
```

```
('Left Approximation: ', 3886.61073964597)
('Right Approximation: ', 3913.60141171808)
```

In [18]:
```
t=var('t')
P_f = plot(f(t), t, a, b)
P_L = sum([plot(ft_n[i], t, t_n[i], t_n[i+1], color = "red") for i in range
P_R = sum([plot(ft_n[i+1], t, t_n[i], t_n[i+1], color = "green") for i in r
show(P_f + P_L + P_R)
```



In [5]:
```
def f(t): return 100*(1-(e)^(-0.1*t))
a, b = 5,50
n = 1000000
```

In [6]:
```
#print(list(range(n)))
#print([i/2 + 1 for i in range(n)])
```

In [7]:
```
t_n = [a + (b-a)/n*i for i in range(n+1)]
#print(t_n)
```

In [8]:
```
ft_n = [f(i).n() for i in t_n]
#print(ft_n)
```

In [9]:
```
#print("Left endpoints: ", ft_n[:-1])
#print("Right endpoints: ", ft_n[1:])
```

In [10]:
```
L = (b-a)/n * sum(ft_n[:-1])
R = (b-a)/n * sum(ft_n[1:])
print("Left Approximation: ", L)
print("Right Approximation: ", R)
```

```
('Left Approximation: ', 3900.20593775198)
('Right Approximation: ', 3900.20863681919)
```

Problem 3

A We first found that 45 mph is equal to 16.5 feet per second. We used this to find T=1.333. We then plugged this into the equation and we found that the car had went approximently 73.32 feet after the speed was reduced to 30 mph.

B We used to equation 66^2t + 2(90)(-16.5) and with this answer we converted it to Mph and found that after 90 feet the car was moving at 25.384 mph

C We then used the equation 0= 16.5x +66 and solved for x and this told us that the car took 4 seconds to stop

D To solve how long it takes to get to the half way point we set VF^2 equal to 66t^2 + 2(66)(-16.5) and had a total of 46.68908 we then set this equal to 16.5x +66 and found that to reach the halfway point it to the car 1.17 seconds.

Problem 4

(a) (9.8/k^2)e^(-kt)+(9.8/k)t-(9.8/k^2) ¶

(b) k= -0.524675 is what we got but there was an error so we were told to use k=0.01 instead.

(c) Using k=0.01 we got the time to equal 78.246 seconds when the tennis ball is dropped from 30,000 feet.

Problem 5

The power is at its max when resistor 1 is equal to resistor 2.