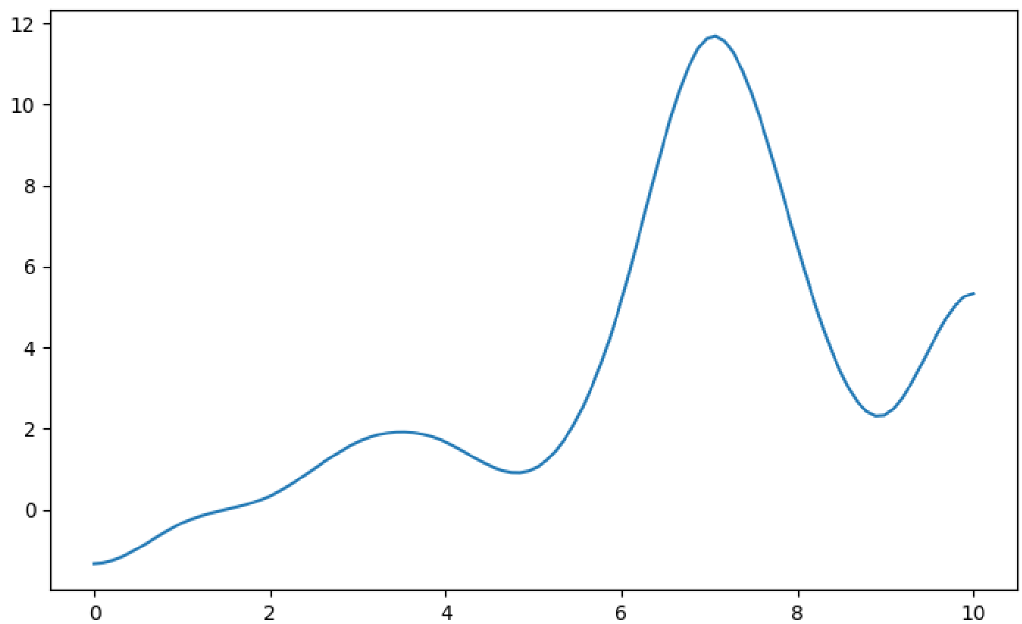
All code runs with python3, and uses numpy, scipy and matplotlib as prerequisites

1

Get a plot by running b-spline.py with python3 with numpy and matplotlib as prerequisites

I hope I understood the question correctly.
I defined `x_i` to be as at the lecture notes: `a + (i * (b - a)) / n`
With `y_i = [-2, 2, 4, 5, 1, 5, 1, 5 , 6, 8]`

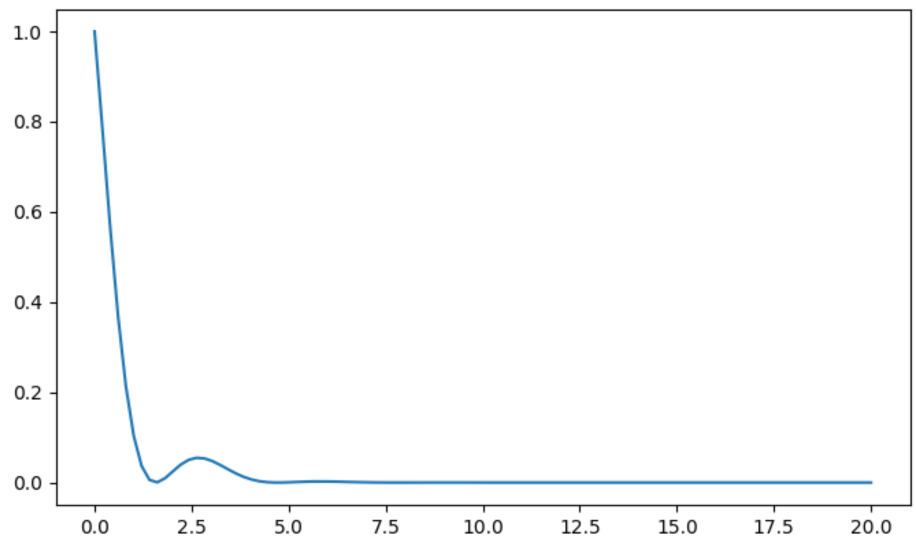
Which gives us the nice plot of



2

Code can be tested at integrate.py using python3 and numpy and scipy as prerequisites

Plotting the function we get:



We are almost sure that after x=10 we don't get a significant addition to the integral, therefore we will truncate at that point.
Then the output of our program for all methods is

```
Expected result using scipy I=(0.6000000000000001,
4.413143579665559e-09)
Simpson's and truncating I=0.6000518449988437
Simpson's and substitution I=0.6008412979670557
Gauss Laguerre n=2 I=0.7284954366282712
Gauss Laguerre n=4 I=0.555108181036087
Gauss Laguerre n=8 I=0.5961158777524371
```

We can see that it starts getting ok results with the Gauss Laguerre method only when we apply 8 points, while both the truncating and sub give good results but indeed it can be smarter to substitute and not truncate the function.

3 a) $\int_0^1 \sin(\sqrt{x})\,dx = \int_0^1 [\sin(\sqrt{x}) - \sqrt{x}]\,dx + \frac{2}{3}$

$$\int_0^1 \sin(\sqrt{x})\,dx = \begin{array}{l} t = \sqrt{x} \\ dt = \frac{1}{2\sqrt{x}}dx \end{array} = 2\int_0^1 t\sin(t)\,dt$$

$$= 2\left(-t\cos t + \int_0^1 \cos(t)\,dt\right) = \sin t - t\cos t = \sin\sqrt{x} - \sqrt{x}\cos\sqrt{x}\,\Big|_0^1$$

$$= \sin 1 - 1\cos 1 - \sin 0 + 0 = \sin 1 - \cos 1 = 2(\sin 1 - \cos 1)$$

$$\int_0^1 [\sin(\sqrt{x}) - \sqrt{x}]\,dx = \begin{array}{l} t = \sqrt{x} \\ dt = \frac{1}{2\sqrt{x}}dx \end{array} = \int_0^1 (\sin t - t)\cdot 2t\,dt$$

$$= 2\int_0^1 \sin t \cdot t\,dt - 2\int_0^1 t^2\,dt = 2(\sin 1 - \cos 1) - 2\left(\frac{t^3}{3}\Big|_0^1\right)$$

$$= 2(\sin 1 - \cos 1) - 2\left(\frac{1}{3}\right) = 2\sin 1 - \cos 1 - \frac{2}{3}$$

then $\int_0^1 [\sin(\sqrt{x}) - \sqrt{x}]\,dx + \frac{2}{3} = 2\sin 1 - \cos = \int_0^1 \sin(\sqrt{x})\,dx$

3

Code can be tested at romberg.py using python3 with np and scipy as prerequisites

The output is:

```
Expected result scipy.integrate.quad I_e=0.6023373578795134
Romberg integration (a) I_a = 0.6019581372889466
Romberg integration (b) I_b = 0.6023371049217722
```

By using n=5 we get a better result with the substituted function f_b.
Which makes sense as the oscillations up to infinity probably still contribute a bit to the result.

$\boxed{4}$ show that $R(1,1)$ of the Romberg method is equal to the Simpson's rule.

Let $R(0,0) = \frac{1}{2}(b-a)\left[f(a)+f(b)\right]$ $\quad h = \frac{b-a}{2^n}$

$$R(1,0) = \frac{1}{2}R(0,0) + \frac{b-a}{2}\sum_{i=1}^{1} f(a+(2i-1)h) =$$

$$= \frac{1}{4}(b-a)\left[f(a)+f(b)\right] + \frac{b-a}{2}f\left(\frac{a+b}{2}\right) = \frac{1}{4}(b-a)\left[f(a)+f(b)+2f\left(\frac{a+b}{2}\right)\right]$$

then $R(1,1) = R(1,0) + \frac{1}{3}\left[R(1,0) - R(0,0)\right] = \frac{4}{3}R(1,0) - \frac{1}{3}R(0,0)$

$$= \frac{1}{3}(b-a)\left[f(a)+f(b)+2f\left(\frac{a+b}{2}\right)\right] - \frac{1}{6}(b-a)\left[f(a)+f(b)\right]$$

$$= \frac{b-a}{6}\cdot\left(2f(a)+2f(b)+4f\left(\frac{a+b}{2}\right) - f(a) - f(b)\right)$$

$$= \frac{b-a}{6}\left(f(a)+f(b)+4f\left(\frac{a+b}{2}\right)\right)$$ which is the simpson rule.