

## TILLEGG C

### Tegne integraler og rektangler

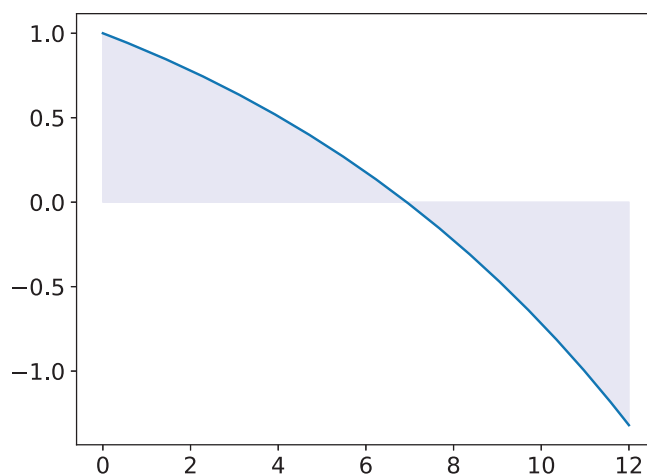
Følgende programmer er *MWE* (Minimal Working Example), slik at eventuell pynt som aksetitler, rutenett og så videre er fjernet. Dette er gjort for at det skal være lettere for leseren å se det minste av kode som er nødvendig for å oppnå ønsket resultat.

#### C.1 Tegne arealet under grafen

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 def f(x):
5     return 2 - np.e**(0.1*x)
6
7 x_verdier = np.linspace(0, 12, 100)
8 y_verdier = f(x_verdier)
9 plt.plot(x_verdier, y_verdier)
10 plt.fill_between(x_verdier, y_verdier, color="lavender")
11 plt.show()
```

Programmet tegner grafen til funksjonen  $f$  og arealet mellom grafen og  $x$ -aksen i intervallet  $[0, 12]$ . Her er  $f$  gitt ved

$$f(x) = 2 - e^{0,1x}$$



Figur C.1: Arealet under grafen.

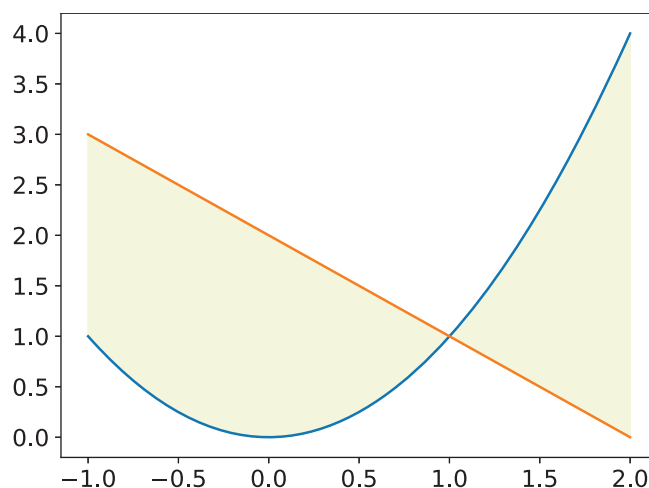
## C.2 Tegne arealet mellom grafer

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 def f(x):
5     return x**2
6
7 def g(x):
8     return 2 - x
9
10 x_verdier = np.linspace(-1, 2, 100)
11 f_verdier = f(x_verdier)
12 g_verdier = g(x_verdier)
13
14 plt.plot(x_verdier, f_verdier)
15 plt.plot(x_verdier, g_verdier)
16
17 plt.fill_between(x_verdier, f_verdier, g_verdier, color="beige")
18 plt.show()
```

Programmet tegner grafene til funksjonene  $f$  og  $g$  gitt ved

$$f(x) = x^2$$
$$g(x) = 2 - x$$

Programmet tegner også arealet mellom grafene i intervallet  $[-1, 2]$ .



Figur C.2: Arealet mellom to grafer.

### C.3 Tegne venstre Riemann-sum

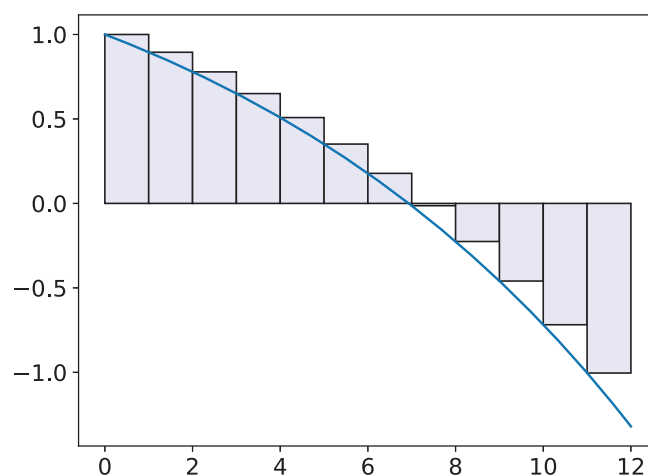
```

1  import numpy as np
2  import matplotlib.pyplot as plt
3
4  def f(x):
5      return 2 - np.e**(0.1*x)
6
7  # Tegne funksjonen
8  x_verdier = np.linspace(0, 12, 100)
9  y_verdier = f(x_verdier)
10 plt.plot(x_verdier, y_verdier)
11
12 # Tegne 12 venstreorienterte rektangler (Riemann-sum)
13 n = 12
14 dx = 12/n
15 x_rekt = np.linspace(0, 12-dx, n)
16 y_rekt = f(x_rekt)
17
18 plt.bar(x_rekt, y_rekt, width=dx, align="edge", color="lavender",
19         ↪ edgecolor="black")
20
21 plt.show()

```

Programmet tegner 12 venstreorienterte rektangler i intervallet  $[0, 12]$ . Funksjonen  $f$  er gitt ved

$$f(x) = 2 - e^{0,1x}$$

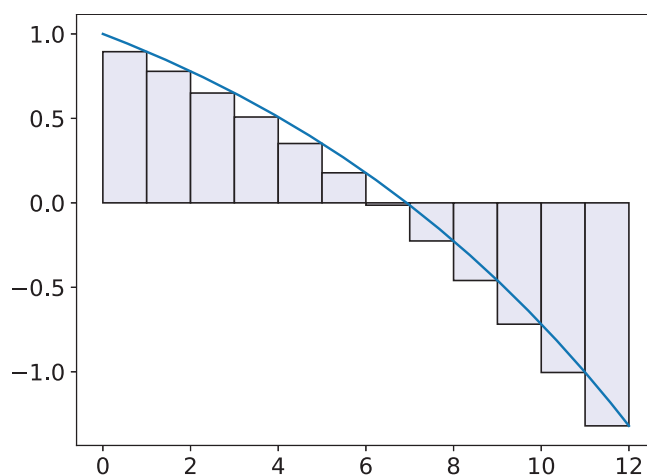


Figur C.3: Venstre Riemann-sum: 12 rektangler.

## C.4 Tegne høyre Riemann-sum

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 def f(x):
5     return 2 - np.e**(0.1*x)
6
7 # Tegne funksjonen
8 x_verdier = np.linspace(0, 12, 100)
9 y_verdier = f(x_verdier)
10 plt.plot(x_verdier, y_verdier)
11
12 # Tegne 12 høyreorienterte rektangler (Riemann-sum)
13 n = 12
14 dx = 12/n
15
16 # Vi starter fra dx og går til 12 for høyreorienterte rektangler
17 x_rekt = np.linspace(dx, 12, n) # Eller np.arange(dx, 12+dx, dx)
18 y_rekt = f(x_rekt)
19 # Merk width=-dx. Dette viser rektanget på høyre side av
20   ↪ x-koordinatet
21 plt.bar(x_rekt, y_rekt, width=-dx, align="edge", color="lavender",
22   ↪ edgecolor="black")
23
24 plt.show()
```

Dette er tilsvarende det forrige eksempelet, men nå med høyreorienterte rektangler.



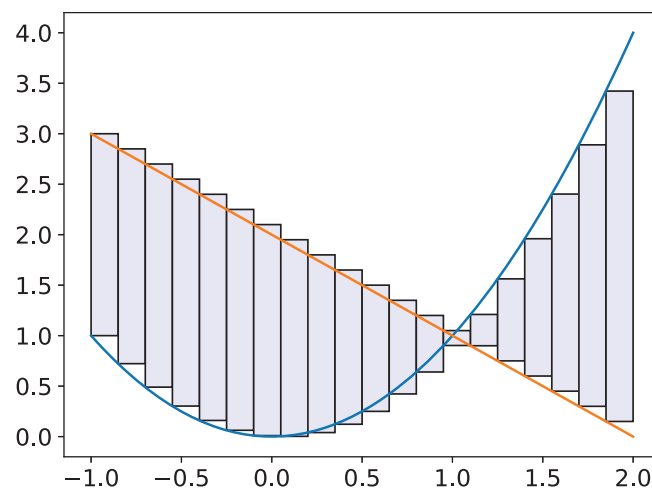
Figur C.4: Høyre Riemann-sum. 12 rektangler.

## C.5 Riemann-sum mellom to grafer

```

1  import numpy as np
2  import matplotlib.pyplot as plt
3
4  def f(x):
5      return x**2
6  def g(x):
7      return 2 - x
8
9  x_verdier = np.linspace(-1, 2, 100)
10 f_verdier = f(x_verdier)
11 g_verdier = g(x_verdier)
12 # Tegne funksjonene
13 plt.plot(x_verdier, f_verdier)
14 plt.plot(x_verdier, g_verdier)
15
16 n = 20 # 20 rektangler
17 a, b = -1, 2 # Intervallet [a, b] = [-1, 2]
18 dx = (b - a) / n
19 x_rekt = np.arange(a, b, dx)
20 y_rekt_f = f(x_rekt)
21 y_rekt_g = g(x_rekt)
22 hoyder = y_rekt_g - y_rekt_f
23 plt.bar(x_rekt, hoyder, bottom=y_rekt_f, width=dx, align='edge',
24        ↪ color="lavender", edgecolor="black")
25 plt.show()

```



Figur C.5: Riemann-sum mellom to grafer. 20 rektangler.

## C.6 Komplette eksempel med pynt

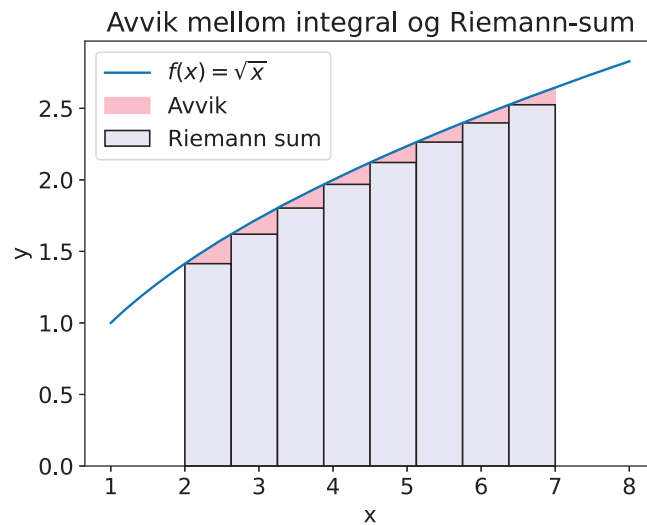
```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 def f(x):
5     return np.sqrt(x)
6
7 # Tegne funksjonen
8 x_verdier = np.linspace(1, 8, 100)
9 y_verdier = f(x_verdier)
10 plt.plot(x_verdier, y_verdier, label=r"$f(x)=\sqrt{x}$")
11
12 # Tegne arealet under grafen fra 2 til 7
13 x_fill = np.linspace(2, 7, 100)
14 y_fill = f(x_fill)
15 plt.fill_between(x_fill, y_fill, color="pink", label="Avvik")
16
17 # Tegner rektanglene
18 n = 8
19 a, b = 2, 7
20 dx = (b - a) / n
21 x_rekt = np.arange(a, b, dx)
22 y_rekt = f(x_rekt)
23 plt.bar(x_rekt, y_rekt, width=dx, align='edge', color="lavender",
24         ↪ edgecolor="black", label="Riemann sum")
25
26 # Legger til etiketter (labels), tittel og aksetitler
27 plt.xlabel("x")
28 plt.ylabel("y")
29 plt.title("Avvik mellom integral og Riemann-sum")
30 plt.legend()
31 plt.show()
```

Dette programmet tegner grafen til funksjonen  $f$  gitt ved

$$f(x) = \sqrt{x}, \quad D_f = [1, 8]$$

Programmet plotter også et rosa område som representerer arealet under kurven fra  $x = 2$  til  $x = 7$ . I tillegg til dette, tegner programmet åtte venstreorienterte Riemann-rektangler (som en tilnærming til integralet) i samme intervall, farget i lavendel. Rektanglene brukes til å tilnærme arealet under kurven, og den rosa regionen viser avviket mellom denne tilnærmingen og det faktiske arealet under funksjonen.

## C. Tegne integraler og rektangler



Figur C.6: Riemann-sum med avvik.

## Oppsummering

I dette kapitlet har vi utforsket hvordan man kan visualisere integrasjon ved hjelp av Python og matplotlib. Vi startet med enkel visualisering av arealet under en kurve ved å fylle mellom kurven og  $x$ -aksen. Deretter utforsket vi hvordan vi kan visualisere arealet mellom to funksjoner ved å fylle mellom deres grafer. Videre så vi på hvordan Riemann-summer kan brukes til å tilnærme integralet av en funksjon, og hvordan man kan visualisere dette ved hjelp av rektangler. Vi utforsket både venstre- og høyreorienterte Riemann-summer. Til slutt så vi på et mer komplett eksempel der flere av disse konseptene ble kombinert, inkludert tilleggsfunksjoner som aksetitler, etiketter og en tittel. Gjennom hele kapitlet har vi lagt vekt på *MWE* for å gjøre det enklere å forstå koden og tilpasse den etter behov.



Figur C.7: Kunst med pensel og kurver.

