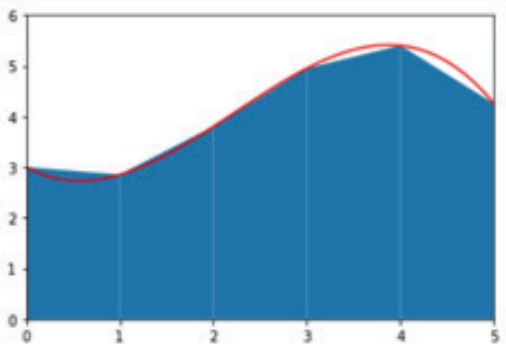
Opplegg 15 – **Numerisk integrasjon med trapesmetoden**

Numerisk integrasjon

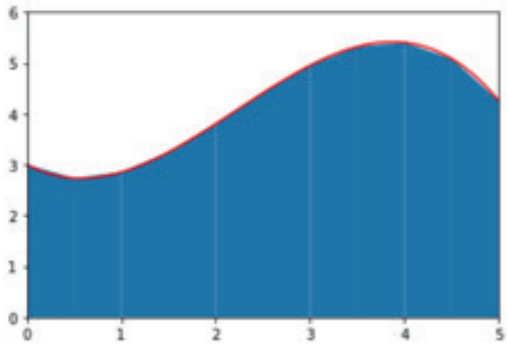
Når vi utfører numerisk integrasjon, trenger vi ikke finne en anti-derivert slik vi gjør når vi beregner integraler for hånd. Det vi trenger er en algoritme som beregner arealet under grafen ved hjelp av kjente figurer som rektangler og trapeser.



Figur 1: N = 5

Trapesmetoden

Ved bruk av trapesmetoden finner vi arealet under en graf ved å dekke området med trapeser. Trapesene konstrueres ved å trekke linjestykker mellom f(0) og f(1), f(1) og f(2) osv. som vist i figurene til høyre. Her har vi brukt hhv. 5 og 10 trapeser.



Figur 2: N = 10

Nedenfor ser du begynnelsen av koden. De manglende linjene er plassert i tilfeldig rekkefølge nedenfor. Klarer du å sette disse i riktig rekkefølge?

```
N = 10
a = 0
b = 5
h = (b-a)/N

def f(x):
    return -0.15*x**3+x**2-x+3
```

Puslespill-programmering for «trapesmetode»-funksjonen

```
def rektangelmetoden(f,a,b,n):
```

```
return Areal
```

```
total =
```

```
print("Trapesmetoden:",round(trapesmetoden(f,a,b,N),5))
```

```
for k in range(0,n):
    total = total + f(a+k*h)
```

```
Areal = h*total
```

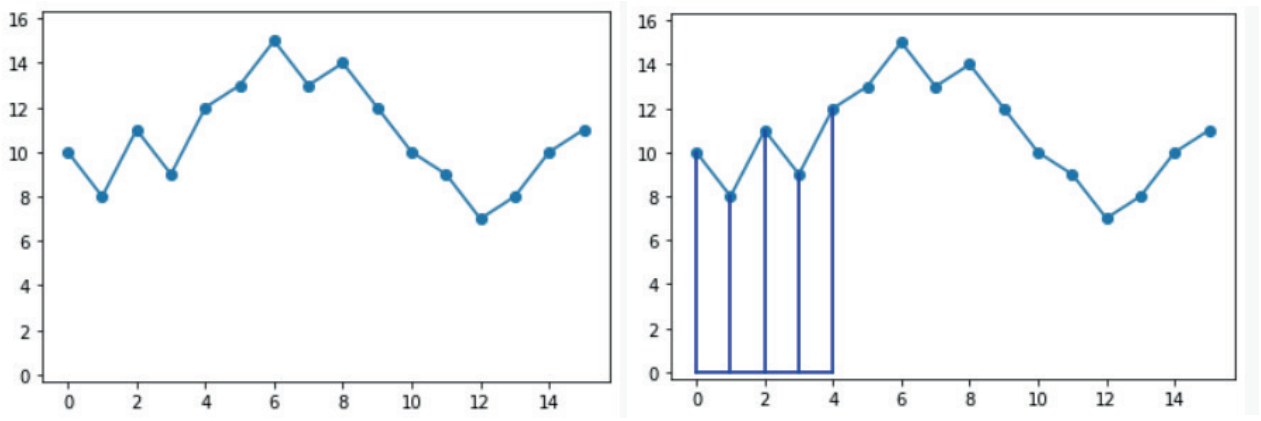
```
h = (b-a)/N
```

Diskusjonsoppgaver

1. Klarer du å utlede uttrykket i løkka? Beregn f.eks. arealet av de tre første trapesene for hånd og legg sammen, og se om du ser et mønster.
2. Hvilken fasong må grafen ha for at trapesmetoden skal gi for stort svar? Eller for lite?
3. Hva er fordelen med trapesmetoden sammenlignet med rektangelmetoden?

Numerisk integrasjon av datasett

I eksemplene på forrige side så vi hvordan vi kunne integrere funksjoner numerisk. Vi kan godt integrere datasett numerisk også, dvs. en mengde med punkter uten funksjonsuttrykk slik som i figuren til venstre. En måte å gjøre dette på, er å bruke trapesmetoden. Vi kan for eksempel tegne inn trapeser som vist i figuren til høyre.



Numerisk integrasjon av datasett

Klarer du å sette sammen linjene til et program som beregner arealet av trapesene? Vi forutsetter at vi har to lister, x og y, med dataene. Enkelte av disse linjene står inne i for-løkka og må stå med innrykk.

Beregner arealet til et trapes

```
areal = (y[i]+y[i+1])/2*h
```

```
h = x[1] - x[0]
```

Beregner bredden av trapesene

Lager en for-løkke

```
for i in range(len(x)-1):
```

```
total = 0
```

Beregner bredden av trapesene

```
total += areal
```

Legger arealet av et trapes til totalen

```
print(total)
```

Skriver ut totalen

Diskusjonsoppgaver

1. Hvorfor trekker vi 1 ifra lengden til x i rangen til for-løkka?
2. Finnes det noen feilkilder ved metoden vist ovenfor? Hvilke?
3. Kunne vi brukt en annen metode, f.eks. rektangelmetoden? Hvordan skulle den evt. programmeres?

Lag en flaskerakett

- mål tid og de tre akselerasjonskomponentene med micro:bit

Oppgave

Lag en vannrakett eller en rakett som bruker natron og eddik som drivstoff. Mål tid og akselerasjonskomponentene (x-, y- og z-retning) mens raketten er i lufta, og lagre til fil. Dette datasettet skal dere bruke for å beregne rakettens fart i de tre retningene ved programmering

Fase 1: Hvordan fungerer en vannrakett eller en natron/eddik-rakett? Undersøk gjerne.

Fase 2: Det er viktig at dere er åpne for alle slags ideer og ikke er for kritiske, da kan nyttige forslag bli avfeid for tidlig.

1. Tenk selv først og tegn gjerne skisser fra forskjellige vinkler.
2. Forklar ideen din for de andre på gruppa. Bruk gjerne skissene i forklaringen.
3. Hele gruppa diskuterer de ulike ideene, og lager en felles plan for bygging.

Fase 3: Gjennomfør planen deres for å lage raketten og programmet for å måle tid og akselerasjonskomponentene som lagres til fil – se neste side for tips.

Fase 4: Test hvor godt raketten virker – får dere brukbare målinger?

Fase 5: Sammenlign resultatene med andre i klassen. Er det noen løsninger dere ønsker i deres rakett?

Fase 6: Gå tilbake til de andre fasene for å gjøre eventuelle forbedringer.

Fase 7: Gjennomfør de siste målingene og dokumenter prosessen på en valgfri måte.

Programmeringsoppgave

1. Hvordan skal dere gjøre om akselerasjonsmålingene til riktig enhet, m/s^2 ?
2. Lag et program som plotter akselerasjonen i hver av de tre retningen som en funksjon av tiden.
3. Bruk de tre komponentene for akselerasjonen for å finne absoluttverdien av akselerasjonen for raketten og legg svarene inn i en python-tabell.
4. Hva finner vi dersom vi integrerer akselerasjonskomponentene?
5. Utvid programmet til å utføre en numerisk integrasjon av akselerasjonen (med trapesmetoden)
    a. for absoluttverdien til akselerasjonen.
    b. for de tre akselerasjonskomponentene hver for seg.
6. Bruk svaret i 4b) for å finne absoluttverdien for farten. Får dere samme svar som i 4a)? Hvorfor/ hvorfor ikke?
7. Sammenlign svaret med de andre i klassen.
    a. Hva var spesielt med raketten som fikk den største farten?
    b. Hva var spesielt med raketten som fikk den minste farten?

Skrive til fil på micro:bit med Python Mu

I slutten av kapittel 9 brukte vi datasett i form av tekstfiler for å lage matematiske modeller. Hadde det ikke vært kult om vi kunne fått micro:biten til å lage slike filer av sine målinger? Da kan vi bruke micro:biten til å samle inn data over en viss periode, uten å være koblet til en datamaskin. Datasettet blir lagret i micro:biten som en .txt-fil, og vi kan legge den over i datamaskinen, før vi plotter resultatene eller lager en matematisk modell med Python.

Det aller første vi må gjøre, er å opprette en tekstfil, det vil i en fil som slutter på .txt, og det er enklest å gjøre i Mu. Lagre den samme i samme mappe som resten av dine Python Mu-filer.

Deretter må den fila du har laget, kopieres over på micro:biten. Dette gjør du på samme måten som for sonaren.

```
with open ('filnavn.txt', 'w') as fil:
```

Åpner fila som heter filnavn.txt og tildeler den til et objekt som vi kaller fil. Siden vi har 'w' (write) i kommandoen gjøres det klar til å skrive inn i fila.

```
with open ('filnavn.txt', 'r') as fil:
```

Tilsvarende som linja over, men siden vi har 'r' (read) i kommandoen gjøres det klar til å lese fila. Det er valgfritt å ta med 'r'.

```
fil.write(y+'\n')
```

Skriver variabelen y til fila vår. +'n' gjør at det blir et linjeskift etter hvert tall, og det må vi ha for at python skal kunne lese filene.
Må stå med innrykk.

```
y = str(x)
```

Gjør om variabelen x til en streng (tekstverdi).
Må stå med innrykk.

```
z = fil.read()
```

Leser det som står i fila vår, og legger det inn i variabel z. Må stå med innrykk.

```
print(z)
```

Skriver variabelen z til skjermen på PC. Må IKKE stå med innrykk.

Da gjenstår det å lage selve programmet. Under er et eksempel der micro:biten måler temperaturen og legger målingen i en .txt-fil ved navn datasett.txt. Det kan være greit å kunne sjekke hvordan datasett.txt-fila ser ut uten å overføre til datamaskinen og åpne den, derfor er dette også med i puslespill-oppgaven.

Puslespill-oppgave

```
print(innhold)
```

```
t = temperature()
```

```
while not button_a.is_pressed():
```

```
sleep(1000)
```

```
fil.write(str(t)+'\n')
```

```
with open('datasett.txt') as fil:
```

```
with open('datasett.txt', 'w') as fil:
```

```
innhold = fil.read()
```

```
from microbit import *
```

Ekstraoppgave

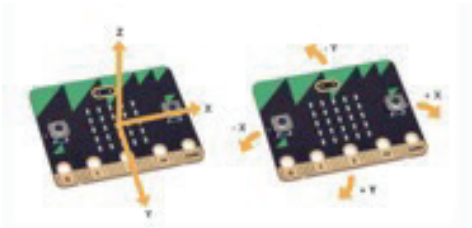
Lag et program som bruker en sensor du kobler på micro:biten for å samle inn data som du lagrer i en fil på micro:biten. Hva må du forandre på? Hva må du forandre i programmet for å lage en fil med to kolonner der den første kolonnen er hvor lang tid det har gått før målingen er gjort?

Akselerasjon og vektorer

Hva er akselerasjon?

Akselerasjon sier noe om endring i farten pr tid. Den raskeste masseproduserte bilen, Porsche 918 Spyder, akselererer fra 0-100km/t på 2,1 sekund. Akselerasjonen er da på 13,2m/s2 som er litt mer enn 1,3G. Til sammenligning er akselerasjonen til romfergene på 2-3 G, mens jagerflypiloter kan utsettes for opp mot 7 G. G tilsvarer tyngdeakselerasjonen på 9,81m/ s2 .

Vektorer; x , y og z:



Bilde hentet fra: https://microbit-challenges.readthedocs.io/en/latest/tutorials/accelerometer.html

En gjenstand kan bevege seg i tre retninger; fram og tilbake, til høyre og venstre, opp eller ned og som regel en kombinasjon av disse.

Vi pleier ofte å bruke et koordinatsystem med tre koordinater for å beskrive slik bevegelse, x y og z. Når noe feks. beveger seg i oppover sier vi at det beveger seg i z-retning har akselerasjon i z-retning, az.
En rakett som skytes opp vil stort sett akselerere i z-retning, men om den ikke er helt i likevekt, vil den også kunne registrere noe akselerasjon i de andre retningene. Vi kan summere bidragene i de tre retningene. For å finne den totale akselerasjonen kan vi bruke denne formelen:

$$a= \sqrt{([\![a\_x]\!]\text{^}2+[\![a\_y]\!]\text{^}2+[\![a\_z]\!]\text{^}2\ )}$$

Akselerasjon med micro:bit

Instrumentet som måler akselerasjon heter akselerometeret og inneholder et lite lodd. Når microbiten ligger i ro på et bord vil tyngdekraften trekke loddet nedover og den vil vise 1000 milli g som tilsvarer 1g som igjen er lik 9,81m/s2. eks:

$$\text{a= } 1300\ milli\ g=13000*9{,}81/1000=12{,}75\ m/s\text{^}2$$

1. Funksjonen accelerometer.get_x(), henter fram x-verdien til akselerasjonen. Lag et program på microbiten som leser av akselerasjonen i z-retning og sender denne til en annen microbit. Husk at az ikke er null når den ligger i ro og at micro:biten måler i milli g.
2. Lag en kode for en microbit som mottar akselerasjonsmåling fra en annen micro:bit og skriver akselerasjonen til skjerm/display.
3. Test microbiten(e) og se om du forstår målingene. Tilpass koden slik at du får fornuftige tall. Har vinkelen/retningen du holder micro:biten noe å si for resultatet?
4. Gjør eventuelt noen endringer i koden for å få bedre målinger. Hva er bedre måling i denne sammenhengen? Tips: vektorregning.
5. Slipp brikken fra en målt høyde og ned på et mykt underlag. Er det bare tyngdekraften som påvirker micro:biten i fallet? Hva skjer med akselerasjonen i det den stopper opp/lander?

Tips

Ta tiden (m/micro:bit) fra du slipper til den treffer bakken. Bruk formelen $at=v-v\_(0\ )$ til å beregne hastigheten, v, like før den treffer bakken. $v_(0\ )$er null når du slipper den.

Høyden på raketten

Dersom du klarer å måle akselerasjon og beregne v kan du bruke formelen $s=v\text{^}2/2g$ til å regne ut hvor høyt raketten går.
Blir dette bedre anslag på høyden enn metoden med å måle tidsforskjell?