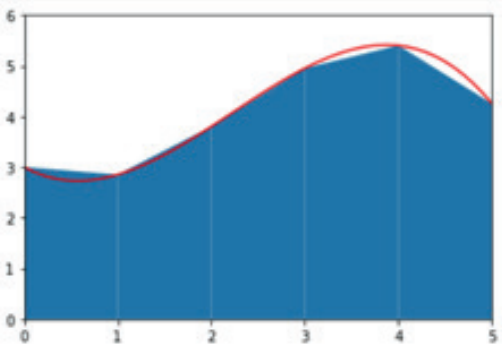
Opplegg 15 – **Numerisk integrasjon med trapesmetoden**

Numerisk integrasjon

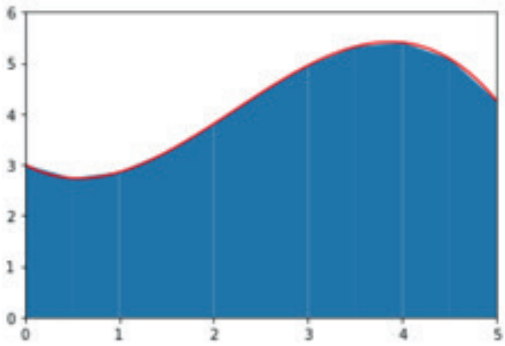
Når vi utfører numerisk integrasjon, treng vi ikkje finne ein anti-derivert slik vi gjer når vi bereknar integral for hand. Det vi treng er ein algoritme som bereknar arealet under grafen med hjelp av kjende figurar som rektangel og trapes.



Figur 1: N = 5

Trapesmetoden

Med bruk av trapesmetoden finn vi arealet under ein graf med å dekke området med trapes. Trapesa blir konstruerte med å trekke linjestykke mellom f(0) og f(1), f(1) og f(2) etc som vist i figurane til høgre. Her har vi brukt respektivt 5 og 10 trapes.



Figur 2: N = 10

Nedanfor ser du delar av koden. Dei manglande linene er plasserte i tilfeldig rekkefølge nedanfor. Klarer du å setje desse i rett rekkefølge?

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

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

Puslespell-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
```

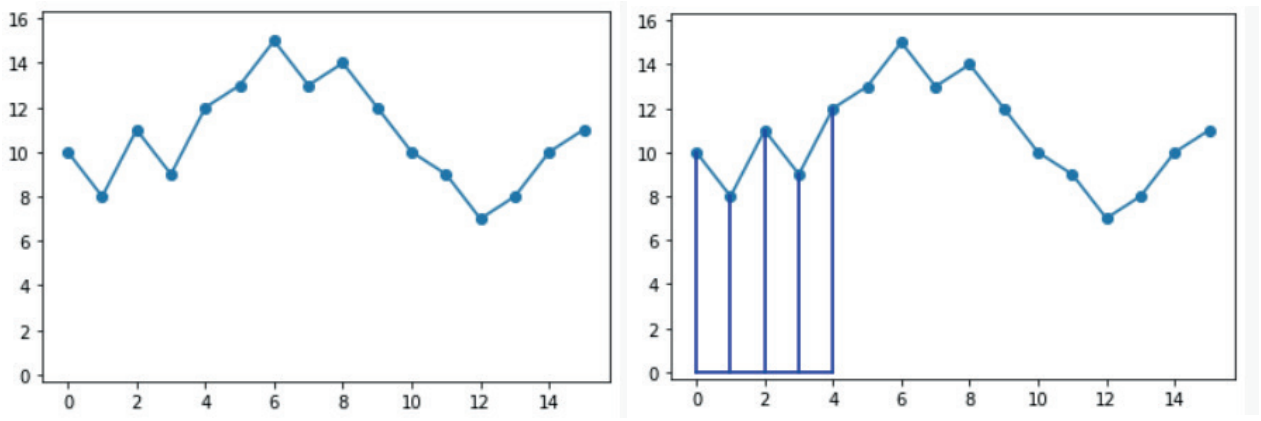
```
h = (b-a)/N
```

Diskusjonsoppgåver

1. Klarer du å utlede uttrykket i lykkja? Berekn t.d. arealet av dei tre første trapesa for hand og legg saman, og sjå om du ser eit mønster.
2. Kva for ein fasong må grafen ha for at trapesmetoden skal gje for stort svar? Eller for lite?
3. Kva er føremonen med trapesmetoden samanlikna med rektangelmetoden?

Numerisk integrasjon av datasett

I døma på førre sida såg vi korleis vi kunne integrere funksjonar numerisk. Vi kan godt integrere datasett numerisk og, dvs. ei mengd med punkt utan funksjonsuttrykk slik som i figuren til venstre. Ein måte å gjere dette på, er å bruke trapesmetoden. Vi kan til dømes teikne inn trapes som vist i figuren til høgre.



Numerisk integrasjon av datasett

Klarer du å setje saman linene til eit program som bereknar arealet av trapesa? Vi forutset at vi har to lister, x og y, med dataene. Einskilde av desse linene står inne i for-lykkja og må stå med innrykk.

Beregner arealet til et trapes

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

Lager en for-løkke

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

```
total += areal
```

Legger arealet av et trapes til totalen

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

Bereknar breidda av trapesa

```
total = 0
```

Definerer ein variabel som skal berekne det totale arealet

```
print(total)
```

Skriv ut totalen

Diskusjonsoppgåver

1. Kvifor trekk vi 1 ifrå lengda til x i rangen til for-lykkja?
2. Finnest det nokon feilkjelder med metoden vist ovanfor? Kva for nokre?
3. nne vi brukt ein annan metode, t.d. rektangelmetoden? Korleis skulle han eventuelt blitt programmert?

Lag ein flaskerakett

- mål tid og dei tre akselerasjonskomponentane med micro:bit

Oppgåve

Lag ein vassrakett eller ein rakett som bruker natron og eddik som drivstoff. Mål tid og akselerasjonskomponentane (x-, y- og z-retning) medan raketten er i lufta, og lagre til fil. Dette datasettet skal de bruke for å berekne raketten sin fart i dei tre retningane med programmering.

Fase 1: Korleis fungerer ein vassrakett eller ein natron/eddik-rakett? Undersøk gjerne.

Fase 2: Det er viktig at de er opne for alle slags idear og ikkje er for kritiske, da kan nyttige framlegg bli kutta ut for tidleg.

1. Tenk sjølv først og teikn gjerne skisser frå ulike vinklar.
2. Forklar ideen din for dei andre på gruppa. Bruk gjerne skissene i forklaringa.
3. Heile gruppa diskuterer dei ulike ideane, og lagar ein 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 kor godt raketten verkar – får de brukbare målingar?

Fase 5: Samanlikn resultata med andre i klassen. Er det nokon løysningar de ønsker i dykkar rakett?

Fase 6: Gå attende til dei andre fasane for å gjere eventuelle forbetringer.

Fase 7: Gjennomfør dei siste målingane og dokumenter prosessen på ein valfri måte.

Programmeringsoppgåve

1. Korleis skal de gjere om akselerasjonsmålingane til rett eining, m/s^2 ?
2. Lag eit program som plottar akselerasjonen i kvar av dei tre retningane som ein funksjon av tida.
3. Bruk dei tre komponentane for akselerasjonen for å finne absoluttverdien av akselerasjonen for raketten, og legg svara inn i ein python-tabell.
4. Kva finn vi dersom vi integrerer akselerasjonskomponentane?
5. tvid programmet til å utføre ein numerisk integrasjon av akselerasjonen (med trapesmetoden)
    a. for absoluttverdien til akselerasjonen.
    b. for dei tre akselerasjonskomponentane kvar for seg.
6. Bruk svaret i 4b) for å finne absoluttverdien for farten. Får de same svaret som i 4a)? Kvifor/kvifor ikkje?
7. Samanlikn svaret med dei andre i klassen.
    a. Kva var spesielt med raketten som fikk den største farten?
    b. Kva 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 modellar. Hadde det ikkje vore kult om vi kunne fått micro:biten til å lage slike filer av målingane sine? Da kan vi bruke micro:biten til å samle inn data over ein viss periode, utan å vere kopla til ein datamaskin. Datasettet blir lagra i micro:biten som ei .txt-fil, og vi kan legge henne over i datamaskinen, før vi plottar resultata eller lager ein matematisk modell med Python.

Det aller første vi må gjere, er å opprette ei tekstfil, det vil seie ei fil som sluttar på .txt, og det er enklast å gjere i Mu. Lagre fila i same mappe som resten av Python Mu-filene dine.

Deretter må den fila du har laga, kopierast over på micro:biten. Dette gjer du på same måten som for sonaren.

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

Opnar fila som heiter filnavn.txt og tildeler den til eit objekt som vi kallar fil. Sidan vi har 'w' (write) i kommandoen gjerest det klar til å skrive inn i fila.

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

Tilsvarande som lina over, men sidan vi har 'r' (read) i kommandoen gjerest det klar til å lese fila. Det er valfritt 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)
```

Gjer om variabelen x til ein streng (tekstverdi).
Må stå med innrykk.

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

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

```
print(z)
```

Skriv variabelen z til skjermen på PC. Må IKKJE 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.

Puslespel-oppgåve

```
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 *
```

Ekstraoppgåve

Lag eit program som bruker ein sensor du koplar på micro:biten for å samle inn data som du lagrar i ei fil på micro:biten. Kva må du gjere annleis?

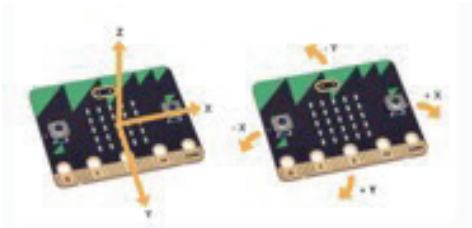
Kva må du forandre i programmet for å lage ei fil med to kolonnar der den første kolonnen er kor lang tid det har gått før målinga er gjort?

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?