

Programmering og modellering X

Vurdering uke 49 2025

Innlevering på Teams

Tillatte hjelpemidler: alt

All kode dere leverer skal være skrevet selv

Skriv kommentarer øverst i filen om bruk av KI eller samarbeid med andre, eventuelt over mindre biter av koden dersom mindre bidrag

Sørg for å les oppgavetekstene nøye

I oppgaver hvor det står «implementer» betyr dette at dere skal skrive det om til kode

I oppgaver hvor det står «lag et program» så står dere fritt til å velge på hvilken måte dere vil implementere det oppgaven ber etter, som klasse, som funksjon, osv.

Oppgaver hvor dere skal komme frem til noe, ikke bare skrive kode, kan besvarelsen deres enten være i form av utskrift når koden kjøres, eller svarsetning(er) med bilder i et word-document

Aktuelle kompetansemål:

- Utvikle helhetlige og strukturerte programmer
 - Vurdere og bruke strategier for testing, feilsøking og feilhåndtering
- Bruke, utlede og lage algoritmer som kan brukes til modellering, og vurdere gyldigheten av disse algoritmene
- Utforske, arbeide med og visualisere målte og simulerte data ved hjelp av programmering

Om noe skulle være uklart i oppgavene, eller dere får noen veldig rare feilmeldinger dere ikke klarer å finne ut av på egen hånd, ta kontakt med meg

Oppgave 1 – Richardson ekstrapolasjon

I denne oppgaven skal vi se på en algoritme for å øke nøyaktigheten på numeriske beregninger. Denne algoritmen kommer til å bygge videre på sentraldifferansen, så vi må først lage den.

- a) Lag en funksjon som beregner den deriverte av en funksjon f i et punkt x ved å bruke sentraldifferansen med en steglengde Δx . I denne oppgaven vil denne funksjonen bli referert til som $D_{\Delta x}(x)$, men du velger navn selv.

Her er algoritmen som Richardson ekstrapolasjon går ut på:

1. Velg en steglende Δx
2. Regn ut $D_{\Delta x}(x)$
3. Regn ut $D_{\Delta x/2}(x)$
4. Da får vi en god tilnærming ved å regne ut $R(x) = \frac{4 \cdot D_{\Delta x/2}(x) - D_{\Delta x}(x)}{3}$

- b) Implementer denne algoritmen som en funksjon som beregner den deriverte av en funksjon f i et punkt x , med en steglengde Δx .
- c) Lag tester for algoritmene dine. Bruk ulike funksjoner som du kan derivere for hånd, og sjekk at din algoritme gir approksimasjoner til disse deriverte i ulike punkter. Sjekk for ulike steglengder og ulike punkter, på ulike funksjoner, og prøv å test funksjonene så grundig som du kan.
(Dere som skriver i Spyder har disse testene i en egen fil hvor filnavnet heter det samme som filen til denne oppgaven, men med test foran. Dere som skriver i Spyder har disse testene i cellen nedenfor der hvor dere løste b).)

- d) I denne deloppgaven skal vi bruke $f(x) = xe^x$, som har derivert $f'(x) = (1+x)e^x$. Gjør en feilanalyse for begge algoritmene når $x = 1$.
(Det vil si, for ulike verdier av Δx , bruk begge algoritmene til å regne ut $D_{\Delta x}(1)$ og $R(1)$, og regn ut den prosentvise feilen sammenlignet med $f'(1)$. Dere bør se at $R(1)$ får mindre prosentvis feil enn $D_{\Delta x}(1)$ for mindre Δx , frem til avrundingsfeilen trer i effekt.)

Oppdater algoritmene dine til å ha en standardverdi for Δx som er lik den verdien som gav minst prosentvis feil.

(Disse kan være forskjellige for sentraldifferansen og Richardson ekstrapolasjon.)

- e) Etter å ha gjennomført c) og d) så kan vi med nokså god trygghet stole på at koden for Richardson ekstrapolasjon fungerer som den skal. Bruk den til å estimere $f'(0.35)$ for $f(x) = 5xe^{-2x}$.
(Altså, regn ut $R(0.35)$)

Oppgave 2 – Spillstatistikk

Jonas og Per ønsker å spille videospill som sine profesjonelle karrierer. For å oppnå dette søker de seg inn til en e-sports organisasjon. I denne oppgaven skal vi ta rollen til rekrutteringspersonen for denne organisasjonen. Vi ber hver av spillerne levere inn en oversikt av deres prestasjon i rundene de har spilt. Disse oversiktene er filene JonasPrestasjon.txt og PerPrestasjon.csv som vi skal bruke i denne oppgaven til å vurdere spillerne.

- a) Lag en klasse som heter Spiller som lagrer følgende informasjon som instansvariabler:
- Navn
 - Antall runder spilt
 - Tid i hver runde, Poeng i hver runde, Handler i hver runde og Feil i hver runde, lagret på en hensiktsmessig måte

Klassen skal også ha metoder for å regne ut følgende informasjon

- Totalt antall tid spilt
- Totalt antall poeng oppnådd
- Totalt antall handler tatt
- Totalt antall feil gjort
- Noe vi kan kalle APM som er antall handler tatt i en runde delt på hvor lang tid runden tok, i minutter
- Noe vi kan kalle Feilrate som er antall feil gjort i en runde delt på hvor mange handler som ble tatt i den runden
- Gjennomsnittlig antall poeng per runde
- Gjennomsnittlig antall handler per runde
- Gjennomsnittlig antall feil per runde

- b) Bruk klassen Spiller til å lage en representasjon for Jonas og Per basert på oversiktene over deres prestasjoner. Gjør hensiktsmessige sammenligninger og/eller visualiseringer til å konkludere om du vil rekruttere noen av spillerne.
(Her er det ingen fasitsvar, jeg vil bare at dere skal utforske dataene og vise frem kunnskap i databehandling og visualisering.)

- c) Ettersom du rekrutterer til en ettertraktet organisasjon, kan det være noe motivasjon for å forfalske informasjon om sin prestasjon. For å være den sikre siden ønsker vi å gjøre en undersøkelse på dataene vi fikk fra Jonas og Per for å se om det er noe mistenkelig der. For å gjøre dette skal vi bruke numerisk derivasjon.

Lag en funksjon som tar imot et visst mål (*Tid, Poeng, Handler, Feil, APM, Feilrate*) og estimerer hvordan dette målet endrer seg fra runde til runde.

- d) Visualiser hvordan disse målene endrer seg fra runde til runde for hver spiller, og konkluder om noe er mistenkelig.

(Dere velger selv hvor mye/lite visualisering dere vil presentere i det dere leverer inn.)

Oppgave 3 – Sykdomsspredning i en befolkning

Vi ønsker å se på hvordan spredningen av en sykdom i en befolkning utvikler seg over tid. Vår befolkning består av N antall individer, hvor hvert individ faller innenfor én av tre grupper:

- Smittbart individ S (*kan bli smittet*)
- Infisert individ I (*er smittet*)
- Restituert individ R (*har vært smittet, er nå ikke smittet, og kan ikke igjen bli smittet*)

På enhver dag er det en sannsynlighet p for at et infisert individ smitter et antall smittbare individer k . Et infisert individ blir restituert etter d dager.

- a) Lag et program som kan simulere én dags utvikling.
- b) Bruk det du lagde i a) til å simulere utviklingen over 30 dager med følgende verdier

$$N = 100, \quad p = 0.1, \quad k = 5, \quad d = 7$$

Hvor dag 1 starter med 1 infisert individ.

Lag en visualisering som viser hvordan antall individer i de tre gruppene utvikler seg over de 30 dagene.