Prosjektoppgave i Matematiske metoder 3 for data

Hans J. Rivertz

5. november 2021

1 Praktisk

1.1 Gruppeinndeling

Samme grupper som før.

1.2 Innlevering

En jupyter-fil eller en zip-fil som inneholder rapport pdf og jupyter-fil med kode. Om det blir vanskelig å skrive matematikk i Jupyter sp kan dere skrive på papir scanne.

Rapporten skal fortelle hva dere har gjort og hvordan dere har gjort det. Den skal være bygd opp med (minst) følgende kapitler.

- **Innledning** (Hvilken oppgave skal løses, og hvorfor)
- **Teori /Metode**(Oppgave 1) Maks 200 ord. (Hvordan er oppgaven løst – metode og teorien bak metoden. Her er det naturlig å ta utgangspunkt i læreboka, men bruk gjerne andre kilder også. Men ikke skriv av/oversett det som står, studer det slik at dere forstår det, og tenk dere deretter at dere skal forklare det til en medstudent. Husk referanser – og referanser til vitenskapelige artikler gjør seg ekstra godt.)
- **Resultater** (Gjerne inkludert tabeller og/eller figurer)
- **Diskusjon/Konklusjon** (Hvorfor blir resultatene slik som de blir? Hva ville dere forventet ut fra teorien, og hva skjer i praksis? Hva har dere lært? Har dere forslag til ting som kunne vært undersøkt videre og/eller ting som kunne vært gjort annerledes/bedre?)
- **Referanseliste** (Fullstendig URI (URL og/eller bok/artikkel-henvisning) til de og bare de referansene dere har nevnt tidligere i rapporten.)
- Alle programmer dere har laget i jupyter note.

2 Innledning

2.1 Beskrivelse

Dere skal ta for dere avsnitt 6.4.3. Lorenz likingene og tilhørende oppgaver 12 og 13. Jeg legger ved en python fil som dere kan importere og bruke importere. Skriv for eksempel **import RKF54 as rkf**.

Viktig bruk numpy. Dere skal sette forskjellige toleranser for utregningen.

$$\begin{aligned} T_1 &= 1.0e-2 \\ T_2 &= 1.0e-3 \\ T_3 &= 1.0e-4 \\ T_4 &= 1.0e-5 \\ T_5 &= 1.0e-6 \\ T_6 &= 1.0e-7 \\ T_7 &= 1.0e-8 \\ T_8 &= 1.0e-9 \\ T_9 &= 1.0e-10 \end{aligned}$$

3 Oppgaver

Oppgave 1 Forklar i korte trekk teorien bak Eulers metode. Fortell også hvordan denne kan forbedres i andre metoder.

Oppgave 2 Test ut Matlab/Python klassen for **Runge-Kutta-Fehlberg** dere finner i filen i BB på et initialverdisystem med minst 2 likninger dere kjenner eksakt løsning på. Ta for eksempel en oppgave fra boka. Finn global feil og sammenlign med akkumulert relativ lokal feil. Har dere konroll på feilen? Bestem forholdet mellom modeltid og bregningstid som en funskjon av toleransen. Hva er optimal toleranse om dere ønsker utregnigner i sanntid?

I resten av oppgaven skal dere bruke **Runge-Kutta-Fehlberg**.

Oppgave 3 La startbetingelsen være som i oppgaven i boka og visualiser løsningen i for toleranse T_4.

Oppgave 4 Gjør oppgave 2 om igjen men tegn forholdet mellom modelltid og beregningstid. Hva er minste toleranse som gir sanntid utregning.

(Modelltid er variabelen det deriveres med hensyn på. Beregningstid er tiden det tar å kjøre algoritmen.)

Oppgave 5 Visualiser tre løsninger av problemet med toleranser T_4, T_5 og T_6 henholdsvis. La tiden gå i 200 sekunder. Vil banene være de samme i hele perioden? Forklar hva du ser.

Referanser

[1] Sauer, T.: Numerical Analysis. Pearson Education Ltd., Harlow, Essex, (2014).