Programmering og Modellering (PoM)
Ugeseddel 7 — Uge 43 — Deadline 24/10

Kim Steenstrup Pedersen, Katrine Hommelhoff Jensen, Knud Henriksen,
Mossa Merhi og Hans Jacob T. Stephensen

9. oktober 2014

1 Plan for ugen

I denne uge tager vi fat på et nyt emne, nemlig numeriske metoder og vi starter med at fokusere på numeriske metoder til integration. Numerisk integration er relevant i situationer hvor vi ikke kan evaluerer integralet af en funktion analytisk og derfor må ty til en numerisk metode til at få en tilnærmelse, en såkaldt approksimation, til værdien af integralet. Vi vil kigge på metoder baseret på Riemannsummer samt en metode kaldet Monte Carlo integration.

Dertil vil vi gennemgå relevante emner indenfor computerarkitektur, styresystemer og programlagdeling. Desuden omtales pseudotilfældige tal (biblioteket `random`) og generererering af sådanne i Python.

Til tirsdag:
Læs noter om numerisk integration af Knud Henriksen. Læs også noterne om Pseudotilfældige tal.

Til torsdag:
Læs noter om numerisk integration af Knud Henriksen. Læs også noterne om Pseudotilfældige tal.

1.1 Gruppeopgave

Den 24/10 senest klokken 15:00 skal besvarelse af følgende opgave afleveres elektronisk via Absalon. Opgaven skal besvares i grupper bestående af 1 til 3 personer og skal godkendes, for at gruppedeltagerne kan kvalificere sig til den afsluttende tag-hjem eksamen. Opgavebesvarelsen skal oploades via kursushjemmesiden på Absalon (find underpunktet `ugeseddel7` under punktet `Ugesedler og opgaver`). Kildekodefiler ("script"-filer) skal afleveres som "ren tekst", sådan som den dannes af `emacs`, `gedit`, `Notepad`, `TextEdit` eller hvilket redigeringsprogram man nu bruger (*ikke* PDF eller HTML eller RTF eller MS Word-format). Filen skal navngives *efternavn1.efternavn2.efternavn3*.43.py, mens andre filer skal afleveres som en PDF-fil med navnet *efternavn1.efternavn2.efternavn3*.43.pdf.

7g1 **Integration ved Riemannsummer:** Lad $f : [a, b] \to \mathbb{R}$ være en kontinuert funktion og lad $\int_a^b f(x)dx$ være dens integrale. Denne opgave går ud på at undersøge Riemannsummer, som konvergerer mod integralet.

Først laves en partition af intervallet, dvs. intervallet inddeles i n stykker, så $a = t_0 < t_1 < t_2 \cdots < t_n = b$. Som det næste approksimeres integralet af funktionen f som summen af arealerne på alle rektangler, der fremkommer ved at tegne en streg fra højre til venstre endepunkt for hvert interval $[t_{i-1}, t_i]$, $i = 1, \ldots, n$. Mere præcist, er en Riemannsum givet ved

$$I_f \approx \sum_{i=1}^{n} |t_i - t_{i-1}| \, f(t_i). \tag{1}$$

(a) Skriv en funktion `rInt(f,a,b,n)`, der beregner Riemannsummen givet en funktion `f`, et startpunkt a, et slutpunkt b og et antal knudepunkter `n`. Knudepunkterne skal vælges equidistante, dvs. $t_i - t_{i-1} = t_j - t_{j-1}$ for alle i og j mellem 1 og n.

(b) Afprøv funktionen `rInt(f,a,b,n)` ved at skrive et program som benytter funktionen med følgende testfunktioner:

- $f : [0, 2\pi] \to \mathbb{R}$, $f(x) = \cos(x)$,
- $\phi : [-10, 10] \to \mathbb{R}$, $\phi(x) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{x^2}{2})$,
- $g : [0.001, 10] \to \mathbb{R}$, $g(x) = \sin(1/x)$

ϕ er tæthedsfunktionen for normalfordelingen. Der findes ingen lukket form løsning til $\int \phi$, dog kan man vise, at $\int_{-\infty}^{\infty} \phi(x)\mathrm{d}x = 1$.

(c) Skriv en ny funktion `rIntMid(f,a,b,n)`, der beregner Riemannsummen givet en funktion `f`, et startpunkt a, et slutpunkt b og et antal equidistante knudepunkter `n`. Integralet skal i nu approksimeres ved at beregne

$$I_f \approx \sum_{i=1}^{n} |t_i - t_{i-1}| \, \frac{1}{2}(f(t_i) + f(t_{i-1})). \tag{2}$$

Afprøv funktionen `rIntMid(f,a,b,n)` på testfunktionerne fra 7i1(b).

(d) Udvid dit program til at inkludere, for hver testfunktion i 7i1(b), et grafisk plot som viser resultatet af de to integrationsmetoder `rInt(f,a,b,n)` og `rIntMid(f,a,b,n)` som funktion af antal intervaller n. Dvs. at x-aksen af grafen skal svare til n og y-aksen skal svare til resultatet af integrationsmetoden, og grafen skal inkludere plottene for både `rInt(f,a,b,n)` og `rIntMid(f,a,b,n)`.

Hvor stort n skal der vælges for at få en rimelig approksimation for hver testfunktion? Forbedrer funktionen `rIntMid(f,a,b,n)` dine resultater i forhold til `rInt(f,a,b,n)`? Besvar disse spørgsmål samt kommenter på graferne skriftligt i PDF filen.

(e) Begge funktioner `rInt(f,a,b,n)` og `rIntMid(f,a,b,n)` samt program skal dokumenteres med docstrings og passende kommentarer. Dertil skal der foretages en struktureret afprøvning af funktionerne og dette skal dokumenteres ved at programmet udprinter resultater af afprøvningen samt en kortfattet beskrivelse som afleveres i PDF filen.

[Hint: Ved afprøvningen af de to funktioner, kan det være en god ide at finde nogle funktioner hvor du kender integralet analytisk og let kan udregne resultatet af approksimationen (dvs. enten (1) eller (2)) i integrationsmetoden. Du kan eksempelvis udregne følgende integraler analytisk og i hånden finde værdierne for (1) og (2): $\int_0^1 k\,dx$, $\int_0^1 x^2\,dx$, $\int_1^2 1/x\,dx$]

1.2 Tirsdagsøvelser

Besvarelser af disse opgaver skal ikke afleveres, men opgaverne forventes løst inden torsdag.

7ti1 Skriv en funktion `linspace(a, b, n)` som tager en start a og slut b værdier for et interval og returnere en liste L med n elementer hvor intervallet $[a, b)$ er inddelt i n dele med lige stor afstand i mellem hver del, således at $L_i - L_{i-1} = L_j - L_{j-1}$ for alle i og j mellem 1 og n.

7ti2 Anvend `matplotlib.pyplot` og funktionen `linspace(a, b, n)` fra opgave 7ti1 til at skriv et program som viser grafer af testfunktionerne fra 7i1(b) i de angivne intervaller for x. Grafen skal indeholde passende akse tekster og en titel.

7ti3 Pseudo-tilfældige talgeneratorer, som Pythons indbyggede `random` modul, er ofte deterministiske algoritmer. Det betyder at man ved samme start tilstand for generator algoritmen altid vil få den samme sekvens af tal. Når man arbejder med tilfældige talgeneratorer kan det være svært at afprøve sine programmer, så det kan være en fordel at kunne fastsætte tilstanden for talgenerator algoritmen. Pythons indbyggede `random` modul initialiseres således at `random`

algoritmen har forskellige start tilstande ved hver opstart af Python. Men modulet giver mulighed for at fastsætte start tilstanden, det såkaldte seed, via funktionen `random.seed(x)`. Åben to terminal vinduer og start Python i hver af disse. Importer nu `random` modulet i begge Python instanser med kaldet `import random`. Kald funktionen `random.random()` i begge Python instanser og bemærk at talene er forskellige og tilsyneladende tilfældige. Prøv derpå at sætte seed `x=42` med kaldet `random.seed(42)`. Hvad sker der nu hvis du kalder `random.random()` i begge Python instanser?

7ti4 Brug funktioner fra modulet `random` til at definere en funktion `terningkast(`n`)` af en formel parameter, som simulerer n kast med en fair terning og returnerer listen `[None,` a_1`, ...,` a_6`]`, hvor a_i er antallet af de n kast, som gav i øjne. Den returnerede liste repræsentere en såkaldt frekvenstabel.

1.3 Torsdagsøvelser

Besvarelser af disse opgaver skal ikke afleveres, men opgaverne forventes løst inden tirsdag i efterfølgende uge.

7to1 Et histogram er en grafikfremstilling af en frekvenstabel. Skriv en funktion `hist(frek)` som tager en frekvenstabel, repræsenteret som en liste af tal, som parameter og laver et grafisk histogram af frekvenstabellen. Histogrammet skal vises ved hjælp af `matplotlib.pyplot.hist` ved at plotte punkter (i, a_i), hvor i er den i'te indgang i frekvenstabellen og a_i repræsenterer frekvensen for denne indgang. Prøv at anvend funktionen `hist(frek)` på resultat listen fra funktionen `terningkast(`n`)` fra opgave 7ti4.

7to2 Funktionen `random.gauss(0, 1)` returnere et tilfældig tal som er fordelt efter normalfordelingen. Anvend denne funktion til at skrive et program som genererer en liste af tilfældige normalt fordelte tal og derpå viser et histogram af listen.

Du kan vælge at anvende din løsning til funktionen `hist(frek)` i opgave 7to1 eller at anvende `matplotlib.pyplot.hist`.

7to3 Skriv en funktion `evalfunc(f, xx)` som evaluerer funktionen $f(x)$ i tallene i listen xx. Afprøv funktionen ved at opbygge en liste af tal xx ved at simulerer ligefordelte tilfældige tal med `random.uniform`. Som testfunktioner kan anvendes funktionerne fra opgave 7i1(b). Lav en graf for hver testfunktion som viser parrene $(x_i, f(x_i))$ som punkter.

7to4 **Monte Carlo integration:** Vi forsøger i denne opgave at approksimere et integrale ved hjælp af tilfældige tal. Vi kan få computeren til at simulere et antal af ligefordelte tal i et interval, eksempelvis $[0, 1]$. I Monte Carlo integrationsmetoden betragter vi integralet

$$I_f = \int_0^1 f(x)\mathrm{d}x$$

og approksimere dette ved at udtrække n tilfældige ligefordelte tal $\{x_1, x_2, \ldots, x_n\}$ i intervalet $[0, 1]$ og derpå beregne gennemsnittet af funktionsværdien i de tilfældige tal

$$I_f \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i)$$

Når vi vil beregne værdien af et bestemt integrale

$$I_g = \int_a^b g(x)\mathrm{d}x$$

kan dette gøres ved at substituere $y = (x - a)/(b - a)$, hvilket giver

$$I_g = \int_0^1 (b - a)g(a + (b - a)y)\mathrm{d}y$$

(a) Skriv et program `mcInt(f,a,b,n)`, der beregner integralet I_f ved at anvende Monte Carlo integrationsmetoden, dvs. ved at simulere tilfældige værdier fra en ligefordeling (`random.uniform(0,1)`) og tage gennemsnittet af de simulerede værdier, hvor n betegner antallet af simuleringer.

(b) Afprøv samme testfunktioner som i opgave 7i1(b). Hvor mange punkter skal der bruges ift. opgave 7i1(b)?

7to5 Et uegentligt integrale, er et integrale, hvor vi integrerer op til uendeligt. Dvs.

$$I_f = \int_0^\infty f(x)\mathrm{d}x$$

Sådanne integraler kan vi også beregne ved at substituere $y = 1/(x+1)$ og derved få

$$I_f = \int_0^1 f(1/y - 1)/y^2 \mathrm{d}y$$

(c) Brug nu det uegentlige integrale til at beregne integralet af normalfordelingen ved hjælp af Monte Carlo integrationsmetoden som beskrevet i opgave 7to4. Du må gerne bruge, at ϕ er symmetrisk omkring 0, dvs.

$$\int_{-\infty}^\infty \phi(x)\mathrm{d}x = 2\int_0^\infty \phi(x)\mathrm{d}x.$$