

Numeriske metoder

Knut Mørken og Øyvind Ryan
Matematisk institutt
Universitetet i Oslo

Oktober 2024

Forord

Disse notatene utgjør numerisk analyse-delen av pensum i førstesemesterkurset MAT1105 ved Universitetet i Oslo. Stoffet bygger på den lineære algebraen som blir forelest i kurset (kap.1 + kap. 4 i "Flervariabel analyse med lineær algebra").

Notatene inneholder en del programmeringseksempler i Matlab, samt en kort innføring i i noen Matlabkommandoer. Mye av dette vil se ganske likt ut i Python. På kursets hjemmeside vil det meste av den tilsvarende Pythonkodem bli lagt ut. Dere velger selv om dere programmerer i Matlab eller Python, men det anbefales at dere bruker litt tid på begge deler.

Blindern, Oktober 2024

Øyvind Ryan

Innhold

1	Litt mer lineær algebra	1
1.1	Plotting av polynomer	1
1.2	Blokkmatriser	4
2	Interpolasjon med polynomer	7
2.1	Interpolasjon på kanonisk form	8
2.2	Interpolasjon med Newtonformen	14
2.3	Interpolasjonsfeilen	21
3	Numerisk derivasjon	24
3.1	Newtonkvotienten	25
3.2	Alternativer til newtonkvotienten	28
3.3	Trunkeringsfeilen for generelle punktmetoder (*)	35
4	Numerisk integrasjon	39
4.1	Hva er integralet?	39
4.1.1	Definisjonen av integralet	40
4.2	Midtpunktsmetoden	42
4.2.1	Feilestimat for midtpunktsmetoden	43
4.2.2	Estimering av steglengden	46
4.3	Alternativer til midtpunktsmetoden	47
4.3.1	Trapesmetoden ($k = 1$)	49
4.3.2	$k = 2$: Simpsons metode	50
5	Kondisjonstall	56
5.1	Definisjon av kondisjonstallet	56
	Fasit	61
	Løsningsforslag	64

Figurer

2.1	Tre interpolasjonspunkter og det tilsvarende kvadratiske interpolasjonspolynommet.	12
2.2	Interpolasjon av e^x i to punkter med en sekant (a), og i tre punkter med en parabel (b).	15
2.3	Interpolasjon av $f(x) = \sin(x)$ med det interpolerende polynommet p_n for forskjellige n	17
2.4	Funksjonen $f(x) = \sqrt{x}$ (heltrukken linje) og den kubiske interpolanten i de fire punktene 0, 1, 2, og 3 (stiplet linje).	18
2.5	Interpolasjon av funksjonen $f(x) = 1/(1 + x^2)$ på intervallet $[-5, 5]$ med polynomer av grad n . Punktene er uniformt fordelt i intervallet i hvert tilfelle.	18
3.1	Sekanten (rød kuve) til en funksjon (svart kurve) i punktene a og $a + h$, samt tangenten i a (blå kurve).	26
3.2	Trunkeringsfeilen ved bruk av newtonkvotienten for tilnærming til den deriverte til $f(x) = \sin x$ i punktet $x = 0.5$, for $h \in [10^{-6}, 10^0]$. Plottet er et logaritmisk plott.	28
3.3	Sekanten i a (rød kurve) til en funksjon f (svart kurve) basert i $a - h$ og $a + h$, samt tangenten i a (blå kurve). Parabelen som interpolerer f i $a - h$, a , og $a + h$ er også vist (svart kurve)	29
3.4	loglog-plott av estimatene på trunkeringsfeilen for de fire metodene vi ser på, for $f(x) = \sin x$ i $x = 0.5$, og for $h \in [10^{-6}, 10^0]$	32
4.1	Arealet under grafen til en funksjon.	40
4.2	Innskrevne (til venstre) og omskrevne (til høyre) stegfunksjoner for en grov partisjon (øverst) og en fin partisjon (nederst).	41
4.3	Midtpunktsmetoden med ett delintervall (a), og fem delintervaller (b).	43
4.4	Trapesmetoden med ett delintervall (a) og fem delintervaller (b).	49
4.5	Simpsons metode med ett delintervall (a) og tre delintervaller (b).	50
5.1	Kondisjonstall for hilbert- og vandermondematisene	59

Programmer

2.1	Kode for å plote det interpolerende polynomet	10
2.2	Plotting av interpolasjonspolynomet på Newtonform	20
3.1	Sammenligning av trunkeringsfeil	31
5.1	Sammenligning av kondisjonstall	58
2	Utgning av øvre/nedre-integralet	75
3	Midtpunktsmetoden	76
4	Trapecmetoden	77
5	Simpsons metode	78

KAPITTEL 1

Litt mer lineær algebra

I dette kapitlet skal vi ta for oss noen tilleggsaspekter i lineær algebra som vil hjelpe oss i de numeriske metodene i de neste kapitlene. Matlab har en syntaks som er strømlinjeformet med notasjonen som brukes i lineær algebra, og her skal vi se på noen eksempler på dette.

1.1 Plotting av polynomer

La oss se på hvordan vi enkelt kan plote flere polynomer i samme koordinatsystem, og la det første polynomet være

$$p_1(x) = c_0 + c_1 x + \cdots + c_n x^n.$$

La også plottepunktene være $x_1, \dots, x_m$. Vi skriver

$$p_1(x_1) = c_0 + x_1 c_1 + \cdots + x_1^n c_n$$

$$p_1(x_2) = c_0 + x_2 c_1 + \cdots + x_2^n c_n$$

$$\vdots$$

$$p_1(x_m) = c_0 + x_m c_1 + \cdots + x_m^n c_n.$$

Ved definisjonen av matrise-vektor-multiplikasjon, kan dette skrives

$$\begin{pmatrix} 1 & x_1 & \cdots & x_1^n \\ 1 & x_2 & \cdots & x_2^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & \cdots & x_m^n \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} p_1(x_1) \\ p_1(x_2) \\ \vdots \\ p_1(x_m) \end{pmatrix}. \quad (1.1)$$

Hvis vi nå har enda et polynom, $p_2(x) = d_0 + d_1 x + \cdots + d_n x^n$, kan vi på samme måte skrive

$$\begin{pmatrix} 1 & x_1 & \cdots & x_1^n \\ 1 & x_2 & \cdots & x_2^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & \cdots & x_m^n \end{pmatrix} \begin{pmatrix} d_0 \\ d_1 \\ \vdots \\ d_n \end{pmatrix} = \begin{pmatrix} p_2(x_1) \\ p_2(x_2) \\ \vdots \\ p_2(x_m) \end{pmatrix}. \quad (1.2)$$

Ved definisjonen av matrisemultiplikasjon kan (1.1) og (1.2) slås sammen til

$$\begin{pmatrix} 1 & x_1 & \dots & x_1^n \\ 1 & x_2 & \dots & x_2^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & \dots & x_m^n \end{pmatrix} \begin{pmatrix} c_0 & d_0 \\ c_1 & d_1 \\ \vdots & \vdots \\ c_n & d_n \end{pmatrix} = \begin{pmatrix} p_1(x_1) & p_2(x_1) \\ p_1(x_2) & p_2(x_2) \\ \vdots & \vdots \\ p_1(x_m) & p_2(x_m) \end{pmatrix}, \quad (1.3)$$

og på samme måte hvis vi har enda flere polynomer. Matrisen i (1.2) og (1.3) med komponenter x_i^j kalles for en *vandermondematrix*, og punktene $x_1, \dots, x_m$ er *generatorene* til vandermondematriksen. Som vi ser trenger ikke vandermondematriser være kvadratiske — antall generatorer (m) kan være forskjellig fra antall potenser ($n + 1$). Disse matrisene vil dukke opp senere i forbindelsene med interpolasjon, samt i forbindelse med numerisk derivasjon og -integrasjon.

La oss nå se på hvordan dette kan programmeres. Litt ny Matlab-syntaks vil forenkle dette vesentlig. Vi er vant til at komponentvise operasjoner i Matlab kun er tillatt for input av samme dimensjoner, og at output også har samme dimensjon. For eksempel, hvis x og p begge er radvektorer av samme lengde, så vil $x \cdot p$ i stedet gi radvektoren $(x_1^{p_1}, x_2^{p_2}, \dots, x_m^{p_m})$. Hvis x og p hadde forskjellig lengde ville ikke dette kunne defineres. Matlab tillater en type komponentvis operasjoner som avviker litt fra dette. Hvis x er en søylevektor, og p en radvektor, så gir $x \cdot p$ matrisen

$$\begin{pmatrix} x_1^{p_1} & x_1^{p_2} & \dots & x_1^{p_n} \\ x_2^{p_1} & x_2^{p_2} & \dots & x_2^{p_n} \\ \vdots & \vdots & \ddots & \vdots \\ x_m^{p_1} & x_m^{p_2} & \dots & x_m^{p_n} \end{pmatrix}$$

i Matlab.

Det andre eksemplet vi skal ta for oss er å legge sammen en søylevektor og en radvektor. Generelt er ikke dette definert, men med x og p som over regner Matlab ut $x+p$ som

$$\begin{pmatrix} x_1 + p_1 & x_1 + p_2 & \dots & x_1 + p_n \\ x_2 + p_1 & x_2 + p_2 & \dots & x_2 + p_n \\ \vdots & \vdots & \ddots & \vdots \\ x_m + p_1 & x_m + p_2 & \dots & x_m + p_n \end{pmatrix},$$

det vil si matrisen med element (i, j) lik $x_i + p_j$.

La oss nå bruke syntaksen for komponentvis potens slik vi beskrev den. I (1.3) brukte vi potenser $0, \dots, n$. vandermondematriksen på venstre side der kan man altså få ved hjelp av koden

```
x_powers = x.^(0:n);
```

der søylevektoren x inneholder plottepunktene $x_1, \dots, x_m$. Hvis vi skal plote over $[0, 1]$ så kan vi for eksempel definere x ved å skrive

```
x = linspace(0,1,100)'
```

Hvis vi har tre polynomer p_1 , p_2 , og p_3 , der koeffisientene er samlet i vektorene p_1 , p_2 , og p_3 , så sier (1.3) at

```
Y = x_powers*[p1 p2 p3]
```

gir en matrise Y med tre søyler, der søyle 1 inneholder punktverdiene for p_1 , søyle 2 for p_2 , og søyle 3 for p_3 . Polynomene kan nå plottes i samme koordinat-system ved å skrive

```
plot(x,Y)
legend('p_1','p_2','p_3')
```

Den vanlige syntaksen for `plot` tar to argumenter x og y , der vektoren x inneholder x -koordinatene til plottepunktene, og y y -koordinatene. Over så vi en litt annen syntaks der

- første argument var søylevektoren med plottepunkter, og
- andre argument var en matrise med like mange rader som i første argument, og tre søyler.

Hvis x er en søylevektor og Y en matrise med like mange rader som x og tre søyler, så er `plot(x, Y)` det samme som å skrive

```
plot(x, Y(:,1))
hold on
plot(x, Y(:,2))
plot(x, Y(:,3))
```

som også er det samme som

```
plot(x, Y(:,1), x, Y(:,2), x, Y(:,3))
```

Dette gjelder også mer generelt hvis Y -matrisen har flere søyler: Syntaksen `plot(x, Y)` plotter alle søyler i Y mot samme vektor x .

En annen nyttig måte å plote på er å bruke såkalte `loglog`-plott. I de numeriske metodene vi skal se på er feilen ofte gitt ved en potens, la oss si $e = Kh^k$, der h er en presisjon i metoden, og konstanten K og potensen k er ukjent. Plotter vi h mot e så er det ikke så lett å se hva K og k er ut fra plottet, siden kurven skyter rett til værs. Et triks kan være å ta logaritmer, noe som gir

$$\log e = \log K + k \log h.$$

Med andre ord, hvis vi plottet logaritmen til feilen mot logaritmen til h , så får vi en rett linje der

- stigningstallet til kurven gir oss potensen k , og

- skjæringen med y -aksen er i $\log K$, slik at vi kan finne K fra denne skjæringen.

Vi skriver `loglog(h,e)` hvis vi skal plote $\log h$ mot $\log e$ i Matlab. Denne kommandoen følger omtrent samme syntaks som `plot`-kommandoen.

Oppgaver for Seksjon 1.1

1. Skriv kode som bruker kommandoen `plot(x,Y)` som beskrevet over til å plote de tre polynomene $p_1(x) = x$, $p_2(x) = x^2$, og $p_3(x) = x^3$ i samme koordinatsystem, og over intervallet $[0, 1]$.

2. Vi så at det er mulig å oppdage sammenhenger på formen $y = Kx^k$ fra et `loglog` plott. Vi sier da at y er *polynomiell* i x , med konstant K og potens k . Tilsvarende, hvis $y = Ke^{kx}$, så sier vi at y er *eksponensiell* i x , med konstant K og eksponent k .

(a) Kommandoen `semilogx(x,y)` plottet $\log x$ mot y i Matlab. Anta at `semilogx(x,y)` viser en linje. Vis at da må x være eksponensiell i y . Hvordan kan vi lese ut konstanten K og eksponenten k ?

(b) Kommandoen `semilogy(x,y)` plottet på samme måte x mot $\log y$. Anta at `semilogy(x,y)` viser en linje. Vis at da må y være eksponensiell i x . Hvordan kan vi lese ut konstanten og eksponenten nå?

3. Anta X og Y er to matriser av samme størrelse. Forklar hva som plottes hvis man skriver `plot(X, Y)`. Her kan du konsultere dokumentasjonen for Matlab

1.2 Blokkmatriser

Av og til er det nyttig å bruke matriser som byggeklosser for større matriser. Hvis vi har fire matriser A, B, C, D , så skriver vi

$$\left(\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right) \quad (1.4)$$

for matrisen der elementene i disse fire matrisene er plassert over, under, til venstre, eller til høyre for hverandre. I Matlab blir dette `[A B;C D]`. Vi kaller dette for en *blokkmatrise*. Her brukte vi streker til å adskille blokkene. Vanligvis vil vi ikke ta med disse strekene.

Uttrykket (1.4) gir bare mening hvis

- A og B har like mange rader,
- C og D har like mange rader.

Da kan nemlig disse settes ved siden av hverandre. På samme måte må A og C ha like mange søyler, B og D ha like mange søyler. Som et eksempel vil koden

```
A = [1 1 1; 1 1 1];
B = [2 2; 2 2];
C = [3 3 3; 3 3 3; 3 3 3];
D = [4 4; 4 4; 4 4];
[A B; C D]
```

returnere matrisen

$$\begin{pmatrix} 1 & 1 & 1 & 2 & 2 \\ 1 & 1 & 1 & 2 & 2 \\ 3 & 3 & 3 & 4 & 4 \\ 3 & 3 & 3 & 4 & 4 \\ 3 & 3 & 3 & 4 & 4 \end{pmatrix}.$$

Videre gir multiplikasjon av blokkmatriser mening: Vi har at

$$\left(\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right) \left(\begin{array}{c|c} E & F \\ \hline G & H \end{array} \right) = \left(\begin{array}{c|c} AE+BG & AF+BH \\ \hline CE+DG & CF+DH \end{array} \right) \quad (1.5)$$

hvis i tillegg

- antall søyler i A er lik antall rader i E ,
- antall søyler i B er lik antall rader i G .

Dette virker kanskje opplagt, men la oss ta med et bevis. La oss for enkelthets skyld anta at alle de åtte matrisene har dimensjon $n \times n$, og la $1 \leq i, j \leq n$. La oss sammenligne element ij på venstre og høyre side i (1.5):

- På venstre side får vi $\sum_{k=1}^n a_{ik}e_{kj} + \sum_{k=1}^n b_{ik}g_{kj}$, som også kan skrives $(AE)_{ij} + BG_{ij} = (AE + BG)_{ij}$.
- På høyre side får vi $(AE + BG)_{ij}$.

Siden dette er riktig for alle $1 \leq i, j \leq n$, så må øvre venstre blokk i venstresiden i (1.5) være $AE + BG$. På samme måte kan vi vise at det er samsvar mellom de andre tre blokkene også, og (1.5) følger.

En blokkmatrise kan også ha flere enn fire blokker — det som står over lar seg greit generalisere. Spesielt kan de ha to blokker (vertikalt eller horisontalt), og vi har at

$$\begin{pmatrix} A & B \end{pmatrix} \begin{pmatrix} C \\ D \end{pmatrix} = AC + BD$$

så lenge dimensjonene matcher.

Oppgaver for Seksjon 1.2

1. Anta A og B er kvadratiske matriser, og la oss se på den kvadratiske blokk-matrisen $C = \begin{pmatrix} A & \mathbf{0} \\ \mathbf{0} & B \end{pmatrix}$.

(a) Vis at alle egenverdier for A også er egenverdier for C . Kan du finne en tilhørende egenvektor? Gjør tilsvarende analyse hvis λ er en egenverdi for B .

(b) Forklar at en egenverdi for C må være en egenverdi for enten A eller B . Fra (a) følger det derfor at egenverdiene til C er lik unionen av egenverdiene til A , og egenverdiene til B .

KAPITTEL 2

Interpolasjon med polynomer

En grunnleggende ide i matematikken er å tilnærme noe komplisert med noe enklere, i håp om at tilnærmingen kan fange opp noe av den viktigste informasjonen fra det kompliserte. Dette er den grunnleggende ideen bak approksimasjon med polynomer, som har vært sentral i matematikk siden oppdagelsen av Kalkulus.

Ideen med approksimasjon er ofte nyttig for å studere funksjoner. De fleste matematiske funksjoner kan ikke uttrykkes som kombinasjoner av de elementære funksjonene vi lærte om på skolen. Likevel kan så å si alle funksjoner av praktisk interesse tilnærmes vilkårlig godt med enkle funksjoner som polynomer, trigonometriske funksjoner, eller eksponensialfunksjoner. Innen beregningsvitenskap er særlig polynomer mye brukt.

Du skal allerede ha lært om approksimasjon med *Taylorpolynomer*, som gir en enkel tilnærming basert på informasjon om funksjonen i kun ett punkt, nærmere bestemt verdien av f og dens deriverte i dette punktet: Hvis $f: \mathbb{R} \rightarrow \mathbb{R}$ er en gitt funksjon som er n ganger deriverbar i et punkt $a \in \mathbb{R}$, så vet vi fra Seksjon 11.1 i *Kalkulus* at

$$p(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \cdots + \frac{f^{(n)}(a)}{n!}(x-a)^n$$

er det entydige polynomet av grad n med de samme n første deriverte som f i a . Én måte å vise dette på er ved først å observere at ethvert polynom p av grad n kan skrives unikt som

$$p(x) = c_0 + c_1(x-a) + c_2(x-a)^2 + \cdots + c_n(x-a)^n.$$

Deriverer vi denne funksjonen k ganger i a får vi at $p^{(k)}(a) = c_k k!$, og om dette skal være lik $f^{(k)}(a)$, må vi ha $c_k = f^{(k)}(a)/k!$, som var det vi skulle vise. Merk at

likningene $p^{(k)}(a) = c_k k!$ kan skrives som det enkle diagonale systemet

$$\begin{pmatrix} 0! & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1! & 0 & 0 & \cdots & 0 \\ 0 & 0 & 2! & 0 & \cdots & 0 \\ 0 & 0 & 0 & 3! & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & n! \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} f(a) \\ f'(a) \\ f''(a) \\ f'''(a) \\ \vdots \\ f^{(n)}(a) \end{pmatrix}.$$

Vi kommer tilbake til dette. *Taylors formel med restledd* sier at hvis f er $n+1$ ganger kontinuerlig deriverbar, så finnes det for enhver x , et tall $\xi \in [a, x]$, s.a.

$$f(x) = p(x) + \frac{(x-a)^{n+1}}{(n+1)!} f^{(n+1)}(\xi). \quad (2.1)$$

Dersom x er nær a , og den deriverte $f^{(n+1)}$ ikke er veldig stor, vil restleddet være lite, og p er dermed nær f . Med andre ord: Om de deriverte til f ikke er veldig store, vil taylorpolynomet p være en god approksimasjon av f , i alle fall nær a .

I dette kapitlet skal vi se på *interpolasjon*, som i stedet gir tilnærminger basert på et sett funksjonsverdier. Vi sier at funksjonen $p: [a, b] \rightarrow \mathbb{R}$ *interpolerer* f i punktene $x_0, x_1, \dots, x_n \in [a, b]$ (som vi antar er forskjellige) hvis

$$f(x_k) = p(x_k) \quad \text{for } k = 0, 1, \dots, n. \quad (2.2)$$

$x_0, x_1, \dots, x_n$ kalles *interpolasjonspunktene*. Funksjonen f kan, men behøver ikke, være kjent: Alt vi trenger fra denne er funksjonsverdiene $f(x_i)$. *polynominterpolasjon*, dreier seg om å finne et polynom p som interpolerer f . Siden betingelsen (2.2) består av $n+1$ ligninger, og et n -tegradspolynom har $n+1$ koeffisienter som kan velges fritt, gir det mening å la interpolasjonspolynomet p være av orden n .

Vi skal se på to måter for å finne interpolasjonspolynomet: Ved hjelp av kanonisk form (Seksjon 2.1), og ved hjelp av Newtonformen (Seksjon 2.2). Vi skal også se at feilen ved interpolasjon kan uttrykkes på en måte som ligner på restleddsformelen (2.1), se Seksjon 2.3. I de neste kapitlene skal vi bruke interpolasjon til å utlede numeriske metoder for de deriverte, og integraler, av funksjoner.

2.1 Interpolasjon på kanonisk form

Et generelt n -tegradspolynom kan skrives på *kanonisk form*

$$p_n(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_n x^n. \quad (2.3)$$

Interpolasjonsbetingelsene tar formen

$$\begin{aligned} f(x_0) &= c_0 + c_1 x_0 + c_2 x_0^2 + \cdots + c_n x_0^n \\ f(x_1) &= c_0 + c_1 x_1 + c_2 x_1^2 + \cdots + c_n x_1^n \\ f(x_2) &= c_0 + c_1 x_2 + c_2 x_2^2 + \cdots + c_n x_2^n \\ &\vdots \\ f(x_n) &= c_0 + c_1 x_n + c_2 x_n^2 + \cdots + c_n x_n^n. \end{aligned}$$

På matriseform kan dette skrives

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} f(x_0) \\ f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{pmatrix} \quad (2.4)$$

Merk at matrisen i (2.4) er en kvadratisk vandermondematrise. Dersom vi klarer å vise at slike matriser er inverterbare, vil (2.4) ha en unik løsning, og vi kan konkludere at det finnes et unikt interpolerende polynom.

Teorem 2.1. *Vandermondematrisen i (2.4) er inverterbar hvis (og bare hvis) interpolasjonspunktene $x_0, x_1, \dots, x_n$ er forskjellige.*

Bevis. La oss kalle vandermondematrisen for V_n . For å vise “bare hvis”-delen av teoremet må vi bevise at dersom minst to av interpolasjonspunktene er like, vil V_n ikke være inverterbar. Men hvis to av punktene er like, vil to av radene i V_n være like, og da er ikke matrisen inverterbar.

Det gjenstår å vise “hvis”-delen av teoremet, som vi gjør ved å vise at determinanten til V_n er ulik 0. Dette skal vi vise ved å gjøre elementære radoperasjoner; siden slike operasjoner ikke forandrer invertibiliteten til en matrise, holder det å vise at den radreduserte matrisen er inverterbar. Og siden matrisen og dens transponerte har samme determinant, er det nok å se på

$$V_n^T = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ x_0 & x_1 & \cdots & x_n \\ x_0^2 & x_1^2 & \cdots & x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_0^n & x_1^n & \cdots & x_n^n \end{pmatrix}.$$

Vi starter med radoperasjoner som nuller ut alt under eneren i øvre venstre hjørne. Trekker vi x_0 ganget med rad $i - 1$ fra rad i , får vi den nye matrisen

$$V_n^T \sim W_n = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 0 & x_1 - x_0 & \cdots & x_n - x_0 \\ 0 & x_1(x_1 - x_0) & \cdots & x_n(x_n - x_0) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & x_1^{n-1}(x_1 - x_0) & \cdots & x_n^{n-1}(x_n - x_0) \end{pmatrix}.$$

Siden addisjon av rader ikke endrer determinanten, vil denne nye matrisen ha samme determinant som V_n . Ekspanderer vi determinanten til W_n langs første

søyle får vi

$$\det(V_n) = \det(W_n) = \begin{vmatrix} x_1 - x_0 & \cdots & x_n - x_0 \\ x_1(x_1 - x_0) & \cdots & x_n(x_n - x_0) \\ \vdots & \ddots & \vdots \\ x_1^{n-1}(x_1 - x_0) & \cdots & x_n^{n-1}(x_n - x_0) \end{vmatrix}$$

Ved å trekke ut faktoren $x_1 - x_0$ fra første søyle, $x_2 - x_0$ fra andre søyle, og så videre, kan dette skrives om til

$$\det(V_n) = (x_1 - x_0)(x_2 - x_0) \cdots (x_n - x_0) \begin{vmatrix} 1 & \cdots & 1 \\ x_1 & \cdots & x_n \\ \vdots & \ddots & \vdots \\ x_1^{n-1} & \cdots & x_n^{n-1} \end{vmatrix}.$$

Her fikk vi altså $(x_1 - x_0)(x_2 - x_0) \cdots (x_n - x_0)$ (som er ulik 0, siden interpolasjonspunktene er ulike!), ganget med en determinant av en vandermondematrise av én dimensjon mindre. Siden en determinant av en 1×1 vandermondematrise er 1, så følger det ved induksjon at alle determinanter av vandermondematriser er ulik 0. ■

Fra beviset over følger det også at determinanten til en vandermondematrise er

$$\det(V_n) = \prod_{i>j} (x_i - x_j).$$

Dette produktet kalles også for *diskriminanten*.

Ved først å finne koeffisientene, så kan vi gjenta det vi gjorde i Seksjon 1.1 for å plote det interpolerende polynomet som følger. Vi kan finne og plote det unike interpolerende polynomet slik:

```
n = length(x)-1;
V = x.^(0:n);
c = V\f;

x_plot = linspace(x(1)-0.1, x(end)+0.1, 100)';
x_plot_powers = x_plot.^(0:n);
plot(x_plot, x_plot_powers*c)
```

Program 2.1. Kode for å plote det interpolerende polynomet

Noen kommentarer om denne koden er på sin plass:

- Koden antar at vektoren x inneholder interpolasjonspunktene, og at vektoren f inneholder de tilhørende interpolasjonsverdiene. Vektorene antas å ha lik lengde.

- På den andre linja brukes den komponentvise potenssyntaksen vi forklarte i Seksjon 1.1. Resultatet blir vandermondematrisen

$$V = \begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix}.$$

- Den tredje linja finner koeffisientene c til det interpolerende polynomet ved å løse Likning (2.4).
- De siste tre linjene sørger for selve plottingen. Først lages en søylevektor `x_plot` med plottepunkter, og som dekker interpolasjonspunktene. `x_plot` er typisk mye lengre enn vektoren `x`.
- En matrise `x_plot_powers` regnes så ut ved å opphøye søylevektor `x_plot` til de samme potensene som på den første linja. Vi får da en ny vandermondematrise, med mange flere rader.
- På siste linje plottes det interpolerende polynomet. Koden `x_plot_powers*c` gir en søylevektor der hvert element representerer en verdi av polynomet.

Koden over prioriterer oversiktighet fremfor effektivitet og minnebruk. Utregningen av vandermondematrisen V kan gjøres ved hjelp av ca. n^2 regneoperasjoner på datamaskinen (se Oppgave 2). I Oppgave 3 blir du bedt om å vise at det krever $\approx \frac{2}{3}n^3$ operasjoner å løse et generelt system med n likninger og n ukjente. Med andre ord: Det er mer krevende å løse systemet enn å regne ut matrisen for systemet.

Eksempel 2.2. Anta vi vil finne et polynom som går gjennom de tre punktene $(0, 1)$, $(1, 3)$, and $(2, 2)$. Med andre ord, vi vil finne et polynom p slik at

$$p(0) = 1, \quad p(1) = 3, \quad p(2) = 2. \quad (2.5)$$

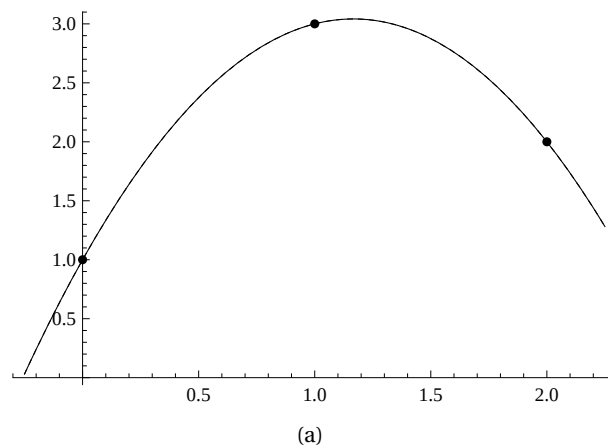
Siden det er tre punkter er det naturlig å anta et kvadratisk polynom $p(x) = c_0 + c_1x + c_2x^2$. Setter vi inn i (2.5) så får vi de tre likningene

$$\begin{aligned} 1 &= p(0) = c_0, \\ 3 &= p(1) = c_0 + c_1 + c_2, \\ 2 &= p(2) = c_0 + 2c_1 + 4c_2. \end{aligned}$$

Vi løser disse og finner $c_0 = 1$, $c_1 = 7/2$, and $c_2 = -3/2$, slik at p er gitt ved

$$p(x) = 1 + \frac{7}{2}x - \frac{3}{2}x^2.$$

Et plott av dette polynomet og interpolasjonspunktene er vist i Figur 2.1. Plottet får du ved å kjøre koden i Program 2.1 etter å ha definert



Figur 2.1. Tre interpolasjonspunkter og det tilsvarende kvadratiske interpolasjonspolynom.

```
x = [0; 1; 2];
f = [1; 3; 2];
```

Oppgaver for Seksjon 2.1

1. Marker følgende påstander som sanne eller usanne.

(a) Anta polynomet p_n interpolerer f i $n + 1$ punkter, og at polynomet q_n interpolerer g i de samme punktene. Det interpolerende polynomet til $f + g$ i de samme punktene er da lik $p_n + q_n$.

(b) Anta polynomet p_n interpolerer f i $n + 1$ punkter, og at polynomet q_n interpolerer g i de samme punktene. Det interpolerende polynomet til $f g$ i de samme punktene er da lik $p_n q_n$.

(c) Det finnes mange polynomer av grad n som interpolerer en gitt funksjon f i $n + 1$ punkter.

2. I denne oppgaven skal vi se på kode som regner ut matrisen vi trenger for å finne den kanoniske formen for interpolasjonspolynom. I Program 2.1 regnet vi ut vandermondematrisen med koden

```
A = x.^(0:n)
```

Sammenlign dette med følgende alternative kode:

```
A(:,0) = ones(n+1,1);
for k = 2:(n+1)
    A(:,k) = A(:,k-1)*x;
end
```

Hvilken av de to alternativene er å foretrekke? Begrunn svaret, og hvilke antagelser du gjør.

3. Forklar at det krever omtrent $\frac{2}{3}n^3$ regneoperasjoner (addisjoner og multiplikasjoner) å løse et generelt system med n likninger og n ukjente ved hjelp av Gausseliminering.

Hint: her kan du få bruk for at $\sum_{k=1}^n k^2 = \frac{n(n+1)(2n+1)}{6}$. Ganger du ut dette uttrykket blir det dominerende n^3 -leddet lik $n^3/3$. Meningen her er at du skal prøve å telle regneoperasjoner som du gjør i Gausseliminering, søyle for søyle. Når du gjør dette vil du se at leddene i summen $\sum_{k=1}^n k^2$ sier noe om hvor mange regneoperasjoner som kreves for å nulle ut elementene under diagonalen ved Gausseliminering, for en konkret søyle.

4. I denne oppgaven skal vi se på et alternativt bevis for at interpolasjonspolynom er unikt.

(a) Anta at det finnes to kvadratiske polynomer p_1 og p_2 som interpolerer funksjonen f i de tre punktene x_0 , x_1 og x_2 . Se på differansen $p = p_2 - p_1$. Hva er verdiene til p i interpolasjonspunktene?

(b) Bruk observasjonen i (a) til å vise at p_1 og p_2 må være like.

(c) Generaliser resultater fra (a) og (b) til polynomer av grad n .

5. En rett-fram utregning av (2.3) i et punkt x krever n multiplikasjoner og n addisjoner i tillegg til utregning av potensene ($n-1$ multiplikasjoner). Totalt blir dette $3n-1$ operasjoner. La oss skrive om (2.3) til

$$p_n(x) = c_0 + x(c_1 + c_2x + c_3x^2 + \cdots c_nx^{n-1}). \quad (2.6)$$

Inne i parentesene på høyre side her er det et polynom av grad $n-1$ (som vi kaller p_{n-1}), som på samme måte kan uttrykkes med et polynom av grad $n-2$ (p_{n-2}), o.s.v.. Vi har at

$$\begin{aligned} p_n(x) &= c_0 + xp_{n-1}(x) \\ p_{n-1}(x) &= c_1 + xp_{n-2}(x) \\ p_{n-2}(x) &= c_2 + xp_{n-3}(x) \\ &\vdots \\ p_1(x) &= c_{n-1} + xp_0(x) \\ p_0(x) &= c_n \end{aligned}$$

For å regne ut $p_n(x)$ starter vi fra siste likning, og fortsetter oppover til første likning. Denne algoritmen kalles også for *Horner's regel*.

(a) Lag en implementasjon av Horner's regel.

(b) Hvor mange aritmetiske operasjoner trenger Horner's regel?

2.2 Interpolasjon med Newtonformen

La igjen $x_0, x_1, \dots, x_n$ være interpolasjonspunktene. Vi sier at polynomet p_n er på *newtonform* dersom det er skrevet på formen

$$p_n(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + \dots + c_n(x - x_0)(x - x_1) \cdots (x - x_{n-1}). \quad (2.7)$$

Newtonformen har en del fordeler sammenlignet med den kanoniske. Interpolasjonsbetingelsene tar nå formen

$$\begin{aligned} f(x_0) &= c_0 \\ f(x_1) &= c_0 + c_1(x_1 - x_0) \\ &\vdots \\ f(x_n) &= c_0 + c_1(x_n - x_0) + \dots + c_n(x_n - x_0) \cdots (x_n - x_{n-1}), \end{aligned} \quad (2.8)$$

eller på matriseform:

$$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & x_1 - x_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n - x_0 & \cdots & (x_n - x_0) \cdots (x_n - x_{n-1}) \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix}. \quad (2.9)$$

Dette er et *triangulært system*, og hvis alle interpolasjonspunktene er forskjellige vil alle elementene langs diagonalen være ulik 0. Dermed er også determinanten til matrisen ulik 0, så matrisen er inverterbar. Matriseligningen (2.9) har dermed en unik løsning $(c_0, c_1, \dots, c_n)$. Med andre ord, vi har en unik Newtonform.

Triangulære systemer er enklere å løse enn generelle systemer. For å se dette, kall den triangulære matrisen for A , la $A = (a_{ij})_{i,j=0}^n$ være elementene i A , og la $A\mathbf{c} = \mathbf{f}$ være systemet vi skal løse. Det er klart at vi kan finne c_0 fra den første likningen, siden denne bare sier $c_0 = f_0$ (husk at $a_{00} = 1$). Anta så at vi har klart å finne $c_0, c_1, \dots, c_{k-1}$ fra de k første likningene. Likning $k + 1$ sier at

$$a_{k0}c_0 + a_{k1}c_1 + \dots + a_{k(k-1)}c_{k-1} + a_{kk}c_k = f_k.$$

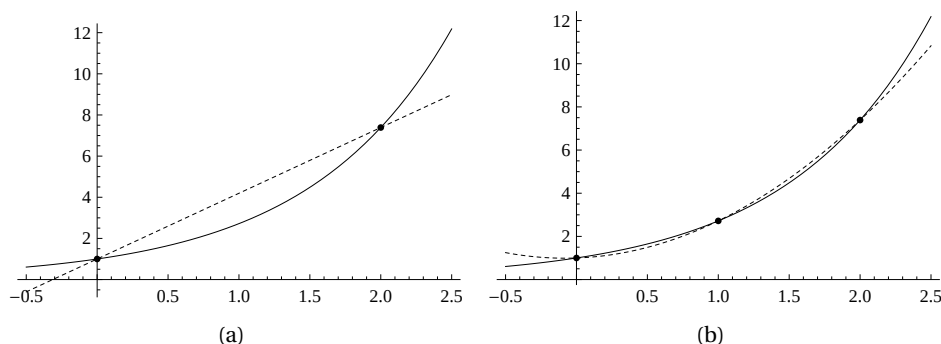
Vi kan skrive dette som

$$c_k = \frac{1}{a_{kk}} (f_k - a_{k0}c_0 - a_{k1}c_1 - \dots - a_{k(k-1)}c_{k-1}) \quad (2.10)$$

(diagonalelementet $a_{kk} = x_k - x_0$ er ulik 0, så dette uttrykket gir mening). Dermed kan vi finne c_k ut fra likning $k + 1$ i systemet. Denne metoden for å løse et triangulært system kalles *fremoversubstitusjon*. Vi finner først c_0 , bruker denne til å finne c_1 , bruker deretter disse til å finne c_2 , og så videre.

Formelen (2.10) for å finne c_k krever k subtraksjoner og k multiplikasjoner. Summerer vi over alle k får vi

$$2 + 4 + 6 + \dots + 2(n-1) + 2n = 2(1 + 2 + \dots + n) = n(n+1) \approx n^2$$



Figur 2.2. Interpolasjon av e^x i to punkter med en sekant (a), og i tre punkter med en parabel (b).

operasjoner (vi tilnærmer med n^2 siden n^2 er det dominerende leddet her). Dette er en stor reduksjon når vi sammenligner med $\approx \frac{2}{3}n^3$ operasjoner for det kanoniske systemet. Ved å tilpasse formen på interpolasjonspolynomet, oppnår vi dermed en regnemessig reduksjon i metoden vår. Den viktige lærdommen fra newtonformen er derfor at det kan være lurt å se på andre måter å representere polynomer på også.

Den triangulære matrisen kan også regnes ut med n^2 operasjoner (se Oppgave 5, som også viser kode for å plote interpolasjonspolynomet via newtonformen, tilsvarende Program 2.1).

Eksempel 2.3 (Newtonformen for $n = 0$). Hvis vi bare har ett interpolasjonspunkt x_0 så blir newtonformen $p_0(x) = c_0$. Vi får at $c_0 = f(x_0)$, slik at

$$p_0(x) = f(x_0).$$

Eksempel 2.4 (Newtonformen for $n = 1$). Med to punkter x_0 og x_1 blir Newtonformen $p_1(x) = c_0 + c_1(x - x_0)$. Vi får likningene

$$\begin{aligned} f(x_0) &= c_0 \\ f(x_1) &= c_0 + c_1(x_1 - x_0) \end{aligned}$$

som har løsning

$$c_0 = f(x_0), \quad c_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}. \quad (2.11)$$

Dette gir

$$p_1(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0),$$

som ikke er noe annet enn sekanten til f gjennom x_0 og x_1 (se Figur 2.2(a)). Til sammenligning ga det førsteordens taylorpolynomet til f tangenten i punktet.

Eksempel 2.5 (Newtonformen for $n = 2$). Vi legger til enda et punkt x_2 og setter opp et kvadratisk interpolasjonspolynom:

$$p_2(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1)$$

Setter vi inn $x = x_0$ og $x = x_1$ får vi likningene

$$\begin{aligned} f(x_0) &= c_0, \\ f(x_1) &= c_0 + c_1(x_1 - x_0), \end{aligned}$$

Dette er de samme likningene vi løste for $n = 1$, så c_0 og c_1 må være gitt ved (2.11) også når $n = 2$. Setter vi inn $x = x_2$ får vi

$$f(x_2) = c_0 + c_1(x_2 - x_0) + c_2(x_2 - x_0)(x_2 - x_1),$$

som gir

$$c_2 = \frac{f(x_2) - f(x_0) - \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x_2 - x_0)}{(x_2 - x_0)(x_2 - x_1)}.$$

Etter en del regning kan dette også skrives

$$c_2 = \frac{\frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0}. \quad (2.12)$$

Interpolasjonspolynomet blir nå en parabel, se Figur 2.2(b).

Mer generelt ser vi fra det triangulære systemet (2.8) at likningen som stammer fra innsettingen $x = x_k$ bare involverer punktene $(x_0, f(x_0)), (x_1, f(x_1)), \dots, (x_k, f(x_k))$. Spesielt vil c_k bare avhenge av disse punktene. Det er derfor vanlig å skrive

$$c_k = f[x_0, \dots, x_k] \quad (2.13)$$

for $k = 0, 1, \dots, n$, slik at newtonformen

$$p_n(x) = c_0 + c_1(x - x_0) + \dots + c_n(x - x_0)(x - x_1) \cdots (x - x_{n-1}) \quad (2.14)$$

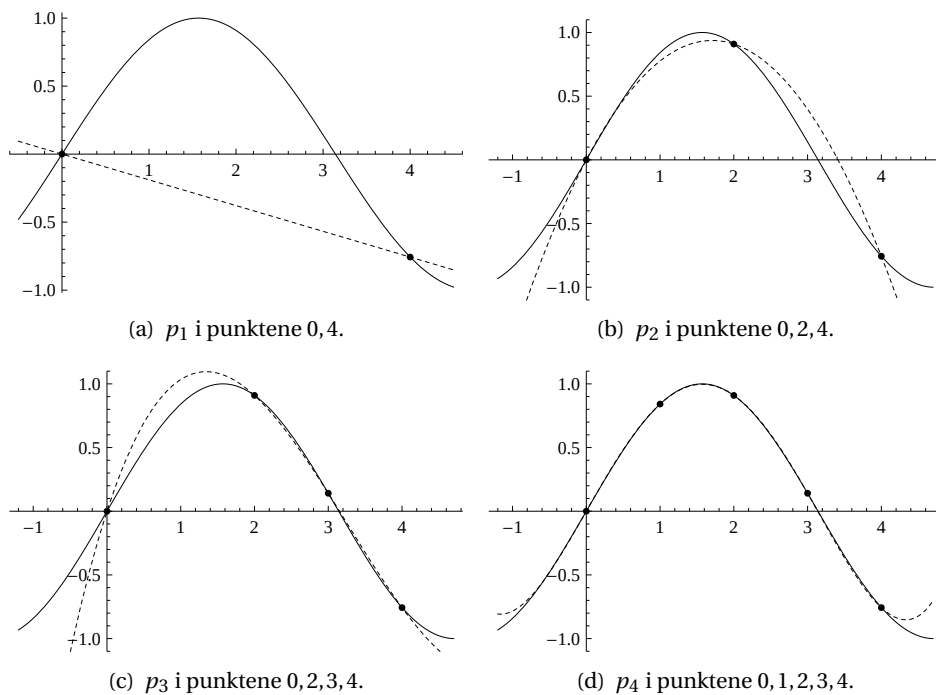
kan skrives

$$\begin{aligned} p_n(x) &= f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \\ &\quad + \dots + f[x_0, x_1, \dots, x_n](x - x_0)(x - x_1) \cdots (x - x_{n-1}). \end{aligned} \quad (2.15)$$

Merk at interpolasjonspolynomene p_n og p_{n-1} oppfyller

$$p_n(x) = p_{n-1}(x) + f[x_0, \dots, x_n](x - x_0) \cdots (x - x_{n-1}).$$

Figur 2.3 viser det interpolerende polynomet p_n for forskjellige verdier av n for funksjonen $f(x) = \sin(x)$. Vi ser at tilnærmingen blir bedre etterhvert som vi øker graden n .



Figur 2.3. Interpolasjon av $f(x) = \sin(x)$ med det interpolerende polynomet p_n for forskjellige n .

Eksempel 2.6. Vi interpolerer funksjonen $f(x) = \sqrt{x}$ i punktene $x_i = i$, for $i = 0, 1, 2, 3$. Newtonformen blir

$$p_3(x) = c_0 + c_1x + c_2x(x-1) + c_3x(x-1)(x-2),$$

og interpolasjonsbetingelsene gir

$$0 = c_0,$$

$$1 = c_0 + c_1,$$

$$\sqrt{2} = c_0 + 2c_1 + 2c_2,$$

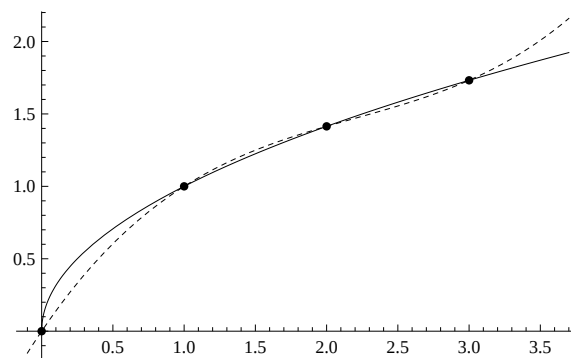
$$\sqrt{3} = c_0 + 3c_1 + 6c_2 + 6c_3.$$

Løser vi dette ligningssystemet får vi

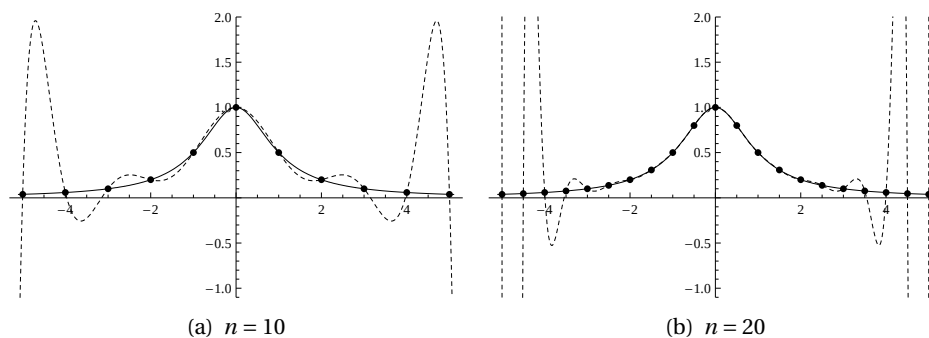
$$c_0 = 0, \quad c_1 = 1, \quad c_2 = -(1 - \sqrt{2}/2), \quad c_3 = (3 + \sqrt{3} - 3\sqrt{2})/6.$$

Figur 2.4 viser det interpolerende polynomet.

Hvis interpolasjon brukes riktig, vil den vanligvis gi en god tilnærming til den underliggende funksjonen. Etterhvert som avstanden mellom punktene avtar (enten ved å øke antall punkter eller flytte punktene nærmere hverandre), kan man forvente at tilnærmingen blir bedre. Men vi vet at det finnes funksjoner



Figur 2.4. Funksjonen $f(x) = \sqrt{x}$ (heltrukken linje) og den kubiske interpolanten i de fire punktene 0, 1, 2, og 3 (stiplet linje).



Figur 2.5. Interpolasjon av funksjonen $f(x) = 1/(1+x^2)$ på intervallet $[-5, 5]$ med polynomer av grad n . Punktene er uniformt fordelt i intervallet i hvert tilfelle.

der taylorpolynomer er dårlige tilnærminger, og det samme kan skje for interpolasjon. Dette problemet kan vi få når den deriverte til funksjonen blir veldig stor. Under er et viktig eksempel på dette.

Eksempel 2.7. *Rungefunksjonen* er definert ved $f(x) = 1/(1+x^2)$. La oss se på denne på intervallet $[-5, 5]$. Figur 2.5 viser oss interpolanter av grad 10 og 20. Midt på intervallet blir feilen mindre når vi øker graden, men nær endepunktene på intervallet blir feilen større når vi øker graden. Vi skal undersøke hva som går galt i Seksjon 2.3.

Newtonformen og den kanoniske formen er bare to måter å skrive det interpolerende polynomet på — det finnes mange andre måter også. En av disse er *Lagrangeformen*, som vi ser på i Oppgave 2.

Oppgaver for Seksjon 2.2

1. Vi interpolerer funksjonen $f(x) = x^2$ med et polynom p_3 av grad 3, i punktene 0, 1, 2, og 3. Hva blir $p_3(4)$?
2. Vi har dataene

x	0	1	3	4
$f(x)$	1	0	2	1

(a) Skriv det kubiske polynomet som interpolerer i disse punktene som

$$p_3(x) = c_0(x-1)(x-3)(x-4) + c_1x(x-3)(x-4) + c_2x(x-1)(x-4) + c_3x(x-1)(x-3),$$

og finn koeffisientene fra interpolasjonsbetingelsene. Dette kalles *Lagrange-formen* til det interpolerende polynomet.

(b) Finn newtonformen til det interpolerende polynomet.

(c) Verifiser at løsningene du fant i (a) og (b) er like.

3. Interpolasjon av et sett med data.

(a) Vi har dataene

x	0	1	2
$f(x)$	2	1	0

Disse er punkter på den rette linja $y = 2 - x$. Finn newtonformen til det kvadratiske interpolerende polynomet og sammenlign. Hva er forskjellen?

(b) Anta at vi interpolerer f med polynomer av grad n i punktene $x_0, \dots, x_n$. Vis at hvis f er et polynom av grad n , så vil interpolanten p_n være identisk med f . Hvorfor forklarer dette resultatet i (a)?

4. Anta at vi har dataene

$$(0, y_0), \quad (1, y_1), \quad (2, y_2), \quad (3, y_3) \quad (2.16)$$

der $y_i = f(i)$ er funksjonsverdier av en ukjent funksjon f . I denne oppgaven skal vi finne uttrykk for tilnærminger til f i forskjellige punkter ved å bruke kubisk interpolasjon.

(a) Finn den rette linjen p_1 som interpolerer de to midterste punktene i (2.16), og bruk $p_1(3/2)$ som en tilnærming til $f(3/2)$. Vis at

$$p_1(3/2) = \frac{1}{2}(y_1 + y_2).$$

(b) Finn det kubiske polynomet p_3 som interpolerer dataene (2.16) og bruk $p_3(3/2)$ som en tilnærming til $f(3/2)$. Vis at

$$p_3(3/2) = \frac{-y_0 + 9y_1 + 9y_2 - y_3}{16}.$$

5. I denne oppgaven skal vi tilpasse plotting av interpolasjonspolynomet (Program 2.1) til newtonformen. Se på følgende kode:

```
n = size(x,1) - 1;
A = zeros(n+1);
A(:,1) = 1;
for c = 2:(n+1)
    A(c:end,c) = A(c:end,c-1).*(x(c:end)-x(c-1));
end

c = zeros(n+1,1);
for k = 1:(n+1)
    c(k) = (f(k) - A(k,1:k)*c(1:k))/A(k,k);
end

x_plot = linspace(x(1)-0.1,x(end)+0.1,100)';
x_plot_powers = ones(size(x_plot,1), n+1);
for c = 2:(n+1)
    x_plot_powers(:,c) = x_plot_powers(:,c-1).*(x_plot-x(c-1));
end
plot(x_plot, x_plot_powers*c)
```

Program 2.2. Plotting av interpolasjonspolynomet på Newtonform

Koden antar som før at interpolasjonspunktene x , med tilhørende verdier f , allerede er definert. Forklar hva som skjer i de tre løkkene. Forklar spesielt den delen som er ansvarlig for å gjøre fremoversubstitusjon. Verifiser at koden trenger omtrent $2n^2$ operasjoner for å finne newtonformen (n^2 operasjoner for å finne matrisen, og ytterligere n^2 operasjoner for å løse systemet med fremoversubstitusjon).

6. I noen sammenhenger kjenner vi kun til noen punktverdier, og noen verdier av de deriverte til en funksjon.

(a) Anta at f er deriverbar i x_1 , og at x_0 , x_1 , og x_2 er forskjellige. Avgjør om det finnes et unikt kubisk polynom p_3 slik at

$$p_3(x_0) = f(x_0), \quad p_3(x_1) = f(x_1), \quad p'_3(x_1) = f'(x_1), \quad p_3(x_2) = f(x_2).$$

(b) Kan du generalisere dette resultatet til tilfeller der noen av de deriverte i interpolasjonspunktene også er kjent? Kan du sette dette i sammenheng med taylorpolynomer?

2.3 Interpolasjonsfeilen

I forrige seksjon så vi eksempler der det interpolerende polynomet p_n blir en bedre og bedre tilnærming til den underliggende funksjonen når n øker, og eksempler der det motsatte skjer. Vi skal nå gi en matematisk begrunnelse for hvorfor dette skjer.

Teorem 2.8. La $x_0, x_1, \dots, x_n \in [a, b]$ være distinkte interpolasjonspunkter, og anta at funksjonen $f: [a, b] \rightarrow \mathbb{R}$ er (minst) $n + 1$ ganger kontinuerlig deriverbar. La p_n være det unike n -teordens interpolerende polynomet til f i punktene $x_0, x_1, \dots, x_n$. Da vil

$$|f(x) - p_n(x)| \leq \frac{M_{n+1}(b-a)^{n+1}}{(n+1)!} \quad (2.17)$$

for alle $x \in [a, b]$, der $M_{n+1} = \max_{[a,b]} |f^{(n+1)}|$.

Bevis. La $x \in [a, b]$. Vi påstår først at det finnes et punkt $y \in [a, b]$ slik at

$$f(x) - p_n(x) = \frac{f^{(n+1)}(y)}{(n+1)!} Q_n(x), \quad (2.18)$$

der Q_n er $(n+1)$ -tegradspolynomet

$$Q_n(x) = (x - x_0)(x - x_1) \cdots (x - x_n).$$

Dersom x er et av punktene $x_0, x_1, \dots, x_n$, er begge sider av (2.18) lik 0, så identiteten er automatisk sann. Anta så at x ikke er et av punktene $x_0, x_1, \dots, x_n$, og definer funksjonen

$$\varphi(t) = f(t) - p_n(t) - \frac{f(x) - p_n(x)}{Q_n(x)} Q_n(t) \quad \text{for } t \in [a, b].$$

Da har φ nullpunkter i de $n+2$ distinkte punktene $x, x_0, x_1, \dots, x_n$ (dette er Oppgave 2). Av middelverdisetningen følger det dermed at det finnes $n+1$ punkter i intervallet (a, b) der φ' er lik 0. Med samme argumentasjon ser vi at det må finnes n punkter der $(\varphi')' = \varphi''$ er lik 0. Fortsetter vi slik, finner vi at funksjonen $\varphi^{(n+1)}$ har minst ett nullpunkt $y \in (a, b)$. Siden $p_n^{(n+1)} \equiv 0$ og $Q^{(n+1)} \equiv (n+1)!$ (dette er Oppgave 3), får vi

$$0 = \varphi^{(n+1)}(y) = f^{(n+1)}(y) - \frac{f(x) - p_n(x)}{Q_n(x)} (n+1)!.$$

Om vi stokker om på denne ligningen får vi nettopp (2.18).

Om vi erstatter $|f^{(n+1)}(y)|$ på høyresiden i (2.18) med dets eget maksimum, får vi

$$|f(x) - p_n(x)| \leq \frac{M_{n+1}}{(n+1)!} |Q_n(x)|.$$

Siden hvert av leddene $(x-x_0), \dots, (x-x_n)$ kan være høyst lik $b-a$ (fordi punktene $x, x_0, x_1, \dots, x_n$ alle ligger i $[a, b]$), ser vi at $|Q_n(x)| \leq (b-a)^{n+1}$ for alle x . Dette beviser (2.17). ■

Siden faktulteten $(n+1)!$ blir veldig stor når n vokser, ser vi at interpolasjonsfeilen minsker når n vokser, *så lenge M_{n+1} ikke vokser for fort!*

Eksempel 2.9. La oss interpolere funksjonen $f(x) = \sin(x)$ på intervallet $[a, b] = [0, 2\pi]$ (se Figur 2.3 for en illustrasjon). Vi velger $n+1$ uniformt fordelte interpolasjonspunkter $x_0, x_1, \dots, x_n$ og får n -tegradspolynomet p_n . Vi vet at de deriverte av f er enten $\pm \cos(x)$ eller $\pm \sin(x)$, og ingen av disse funksjonene oppnår funksjonsverdier større enn 1. Derfor er $M_{n+1} \leq 1$ for alle n . Vi får dermed at

$$|f(x) - p_n(x)| \leq \frac{(2\pi)^{n+1}}{(n+1)!}.$$

Siden $(n+1)!$ vokser mye raskere enn $(2\pi)^{n+1}$ når n vokser, vil feilen gå raskt mot 0 når n vokser.

Eksempel 2.10. La f være *rungefunksjonen* fra Eksempel 2.7, gitt ved $f(x) = 1/(1+x^2)$ for $x \in [-5, 5]$. En litt teknisk utregning viser at $M_{n+1} \approx (n+1)!$ på intervallet $[-5, 5]$. Feilestimatet (2.17) sier dermed at

$$|f(x) - p_n(x)| \leq \frac{M_{n+1}(b-a)^{n+1}}{(n+1)!} \approx (b-a)^{n+1} = 10^{n+1},$$

som er et veldig stort tall når n er stor. Feilestimatet garanterer altså *ikke* at interpolasjonsfeilen er liten når n er stor, og som vi så i Eksempel 2.7, er feilen faktisk ganske stor.

Oppgaver for Seksjon 2.3

1. Vi tilnærmer $f(x) = e^x$ med et interpolerende polynom p_n i punktene

$$0, 1/n, 2/n, \dots, (n-1)/n, 1.$$

Hvor stor må n være for at feilen $|f(x) - p_n(x)|$ skal være mindre enn 0.001 for alle $x \in [0, 1]$?

2. Vis at $\varphi(t)$ i beviset for Teorem 2.17 oppfyller

$$\varphi(x) = \varphi(x_0) = \varphi(x_1) = \dots = \varphi(x_n) = 0$$

3. Vi har sett at

$$Q_n(x) = (x-x_0)(x-x_1)\cdots(x-x_n)$$

er sentral i Newtonformen, og i beviset for Teorem 2.17. I de neste kapitlene skal vi derivere og integrere det interpolerende polynomet, og vi vil derfor ha behov for å derivere og integrere Q_n .

(a) Vis at $Q^{(n+1)}/x = (n+1)!$

(b) Et spesielt tilfelle vi vil se på er når x_i 'ene er en uniform partisjon, det vil si at $x_i - x_{i-1} = h$ for alle i . Regn ut $\int_{x_0}^{x_2} Q_1(x) dx$ og $\int_{x_0}^{x_3} Q_2(x) dx$.

Hint: Delvis integrasjon kan være til hjelp her.

KAPITTEL 3

Numerisk derivasjon

Derivasjon er en grunnleggende operasjon i matematikk, med mange anvendelser. Derfor er det viktig å ha gode metoder for å beregne og bruke de deriverte. Du har sannsynligvis lært de grunnleggende reglene for derivasjon i skolen — symbolske metoder som passer for utregninger på papir. Disse er av begrenset verdi på datamaskiner siden mange programmeringsspråk ikke støtter symbolske beregninger.

Et annet punkt er at funksjonen vår ofte bare er kjent i et sett isolerte punkter, slik som målinger av posisjonen til en bil med en GPS-enhet. Anta at vi vil beregne bilens fart. Hvis posisjonen til bilen er kjent for alle tidspunkter (som en matematisk funksjon) så kan dette gjøres ved symbolsk derivasjon, men når posisjonen til bilen er bare kjent for enkelte tidspunkter så må vi ty til andre metoder.

Løsningen er numeriske metoder som tilnærmer den deriverte med passende kombinasjoner av kjente funksjonsverdier. La oss starte med å formulere akkurat dette som en generell strategi for numerisk derivasjon.

Strategi 3.1. Anta at f er kjent i punktene

$$\dots, a - 2h, a - h, a, a + h, a + 2h, \dots$$

(så, h er “oppløsningen” for kjente verdier av f).

1. Plukk en delmengde av $n + 1$ av disse punktene nær a , og skriv disse som

$$a + m_0h, a + m_1h, a + m_2h, \dots, a + m_nh, \quad (3.1)$$

der $m_0, \dots, m_n$ er heltallige.

2. Finn polynomet $p_n(x)$ som interpolerer f i de $n + 1$ punktene (3.1).
3. Bruk $p'_n(a)$ som en tilnærming til $f'(a)$.

Vi skal også kalle dette for *punktmetoden* basert på punktene (3.1). Punktmetoder kan brukes for høyere ordens deriverte også, ved å bruke $p_n^{(k)}(a)$ som en tilnærming til $f^{(k)}(a)$. Vi definerer *trunkeringsfeilen* i denne tilnærmingen som

$$E_k(f; a, h) = f^{(k)}(a) - p_n^{(k)}(a) \quad (3.2)$$

Dette er feilen som følger av at vi deriverer det interpolerende polynomet i stedet for funksjonen f . Målet er å få denne feilen så liten som mulig.

I det aller enkleste tilfellet baserer punktmetoder seg på kun to kjente funksjonsverdier, og vi skal i Seksjon 3.1 se hvordan dette naturlig henger sammen med definisjonen av den deriverte. I Seksjon 3.2 skal vi så se hvordan punktmetoder med flere punkter kan bli enda mer presise. Metodene for numerisk integrasjon i Kapittel 4 vil bli utledet og analysert på den samme måten som metodene for derivasjon i dette kapitlet.

I dette kapitlet vil vi skrive $M_k = \max_x |f^{(k)}(x)|$ (når f er k ganger deriverbar), der maksimum tas over et intervall som går klart fram fra konteksten. I notasjonen p_n og $E_k(f; a, h)$ vil det gå klart ut fra konteksten hva tallene m_i er, selv om de ikke er en del av notasjonen.

3.1 Newtonkvotienten

Den deriverte av f i et punkt a er definert som

$$f'(a) = \lim_{h \rightarrow 0} \frac{f(a+h) - f(a)}{h}. \quad (3.3)$$

Vi vil anta at f er deriverbar i a , det vil si at denne grenseverdien eksisterer. For en fiksert $h \neq 0$ kalles tilnærmingen

$$\frac{f(a+h) - f(a)}{h} \quad (3.4)$$

til $f'(a)$ for *Newtons differenskvotient*, eller bare *newtonkvotienten*.

Bruker vi punktmetoden basert på punktene a og $a+h$ får vi den rette linjen p_1 gjennom $(a, f(a))$ og $(a+h, f(a+h))$, eller sekanten gjennom disse. Denne kan skrives

$$p_1(x) = f(a) + \frac{f(a+h) - f(a)}{h}(x-a).$$

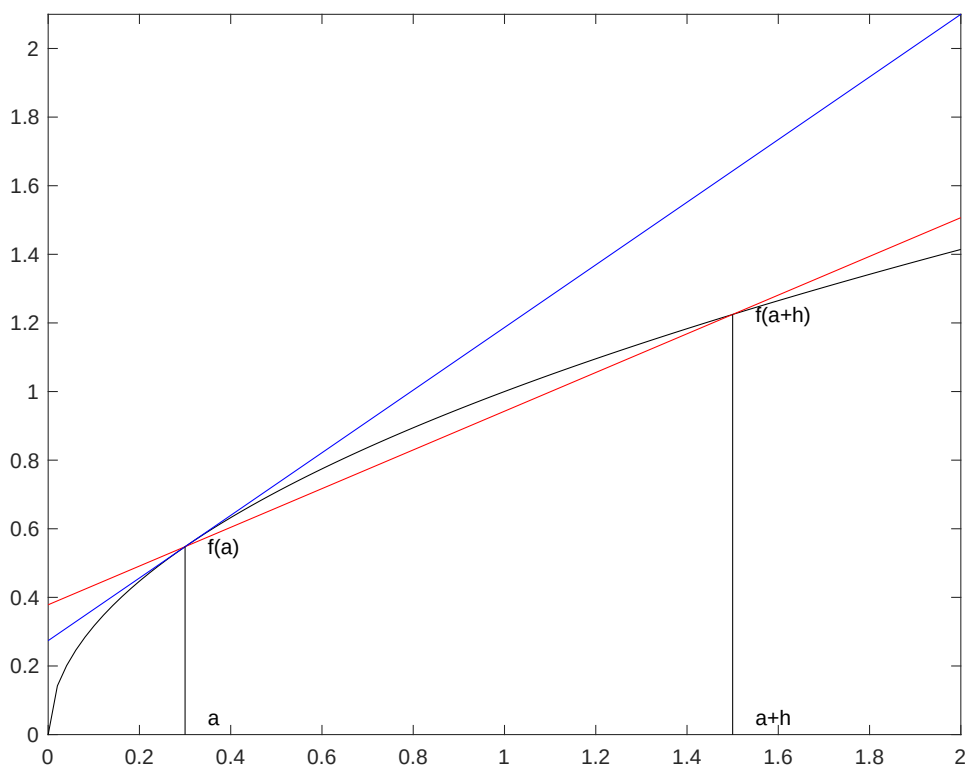
Den deriverte av p_1 , sekantens stigningstall, er nøyaktig høyresiden i (3.4). Newtonkvotienten faller altså sammen med punktmetoden basert på a og $a+h$ (se Figur 3.1).

Trunkeringsfeilen for newtonkvotienten er

$$E_1(f; a, h) = f'(a) - \frac{f(a+h) - f(a)}{h}.$$

La oss se nærmere på denne. Fra Taylors formel med restledd (2.1), får vi at

$$f(a+h) = f(a) + hf'(a) + \frac{h^2}{2}f''(\xi_h), \quad (3.5)$$



Figur 3.1. Sekanten (rød kuve) til en funksjon (svart kurve) i punktene a og $a + h$, samt tangenten i a (blå kurve).

der $\xi_h \in [a, a + h]$. Stokker vi om på dette uttrykket får vi følgende uttrykk for trunkeringsfeilen:

$$f'(a) - \frac{f(a+h) - f(a)}{h} = -\frac{h}{2} f''(\xi_h). \quad (3.6)$$

Hvis den andrederiverte er kontinuerlig, så sier ekstremalverdisetningen at funksjonen $|f''|$ har et maksimum på intervallet $[a, a + h]$, som vi kaller M_2 . Fra dette får vi følgende feilestimat.

Proposisjon 3.2. Anta f har kontinuerlige andrederiverte nær a . Da oppfylder trunkeringsfeilen for netwonkvotienten

$$|E_1(f; a, h)| \leq M_2 \frac{h}{2}, \quad (3.7)$$

der $M_2 = \max_x |f''(x)|$.

Eksempel 3.3. La oss se på funksjonen $f(x) = \sin(x)$ i punktet $a = 0.5$. Siden den deriverte er $f'(x) = \cos(x)$, og $f'(a) \approx 0.8775825619$, er det lett å sjekke nøyaktig-

heten i Newtons kvotient. Prøver vi med noen verdier av h får vi

h	$(f(a+h) - f(a))/h$	$E_1(f; a, h)$
10^{-1}	0.8521693479	2.5×10^{-2}
10^{-2}	0.8751708279	2.4×10^{-3}
10^{-3}	0.8773427029	2.4×10^{-4}
10^{-4}	0.8775585892	2.4×10^{-5}
10^{-5}	0.8775801647	2.4×10^{-6}
10^{-6}	0.8775823222	2.4×10^{-7}

Vi ser at tilnærmingen blir bedre med avtagende h , som forventet. Mer presist ser vi at når h blir redusert med en faktor 10, så blir feilen redusert med samme faktor.

La oss sjekke at estimatet for feilen (3.7) stemmer overens med verdiene i tabellen. Vi har $f''(x) = -\sin(x)$, og siden $|\sin(x)| \leq 1$ for alle x , kan vi velge $M_2 = 1$. Dermed er høyre side i (3.7) lik $\frac{1}{2}h$. Tabellen tilsier at den egentlige feilen er $|E_1(f; a, h)| \approx 0.24h$, så estimatet $|E_1(f; a, h)| \leq 0.5h$ overestimerer feilen med en faktor 2. I slike feilestimer er det veldig vanlig å bomme litt, og “overdrive” hvor stor feilen kan bli. Det sentrale her er at vi har estimert noe ukjent (feilen E_1) med noe som er beregnbart (høyresiden i (3.7)).

Figur 3.2 viser et loglog-plott av h mot trunkeeringsfeilen, sammen med den øvre grensen for feilen $M_2 h/2$. Her satt vi $M_2 = 1$, slik at vi gjør den nevnte overestimeringen. Et bedre estimat ville vi fått ved å regne ut $\max_x |f''(x)|$ over et lite intervall i nærheten av a i stedet, men gjør du dette vil du se at de to kurvene blir liggende oppå hverandre. Plottet ser ut som en linje med derivert lik 1. Siden dette er et logaritmisk plott, er dette i overensstemmelse med tilnærmingen $|E_1(f; a, h)| \approx \frac{h}{2}|f''(a)|$ for trunkeeringsfeilen, siden vi ved å ta logaritmer i denne får

$$\begin{aligned}\log|E_1(f; a, h)| &\approx \log h + \log(0.5) + \log(\sin(0.5)) \\ &\approx \log h - 0.6203.\end{aligned}$$

Dette samsvarer bra med der linja treffer y -aksen.

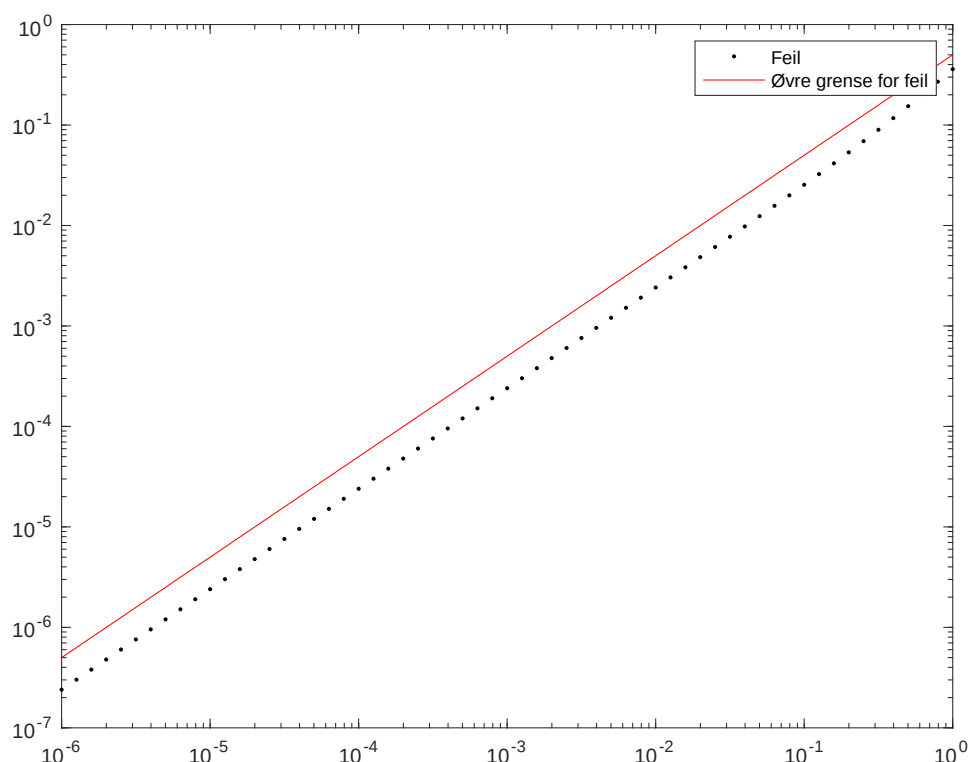
Oppgaver for Seksjon 3.1

1. Et alternativ til newtonkvotienten (3.4) er den venstresidige varianten

$$f'(a) \approx \frac{f(a) - f(a-h)}{h}.$$

Vis at vi denne faller sammen med punktmetoden basert på a og $a-h$.

2. Skriv et program som regner ut newtonkvotienten for funksjonen $f(x) = e^x$ i $a = 1$, og bruker $h = 10^{-k}$ for $k = 1, 2, \dots, 14$. Regn også ut feilen. Oppfør tilnærmingen og dens feil seg slik som Proposisjon 3.2 sier?



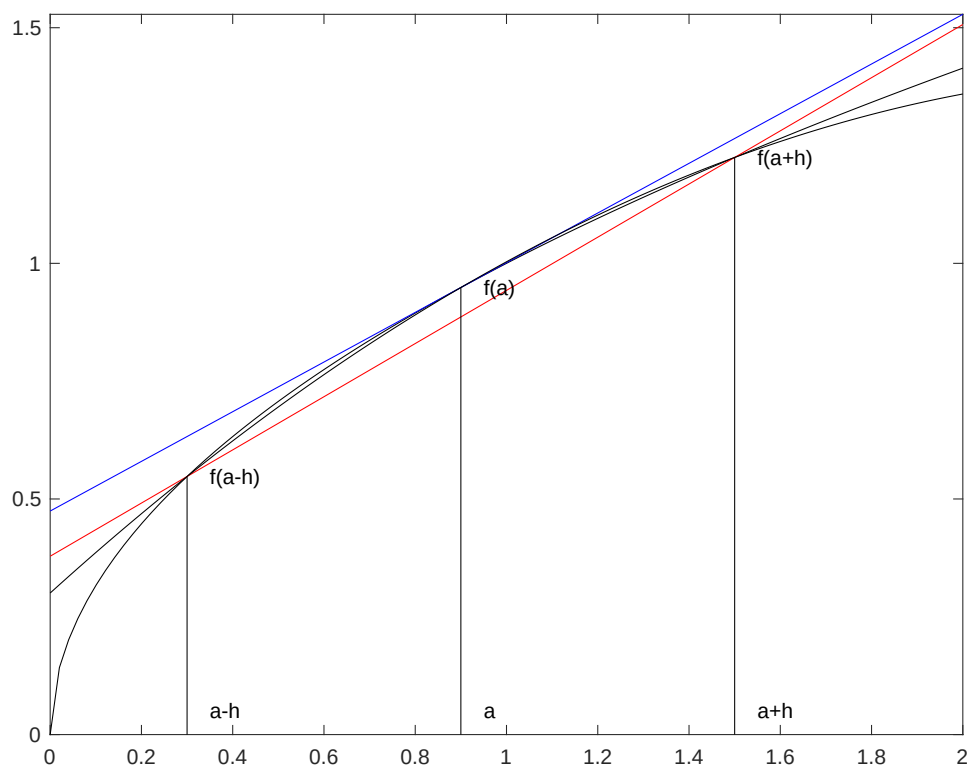
Figur 3.2. Trunkeringsfeilen ved bruk av newtonkvotienten for tilnærming til den deriverte til $f(x) = \sin x$ i punktet $x = 0.5$, for $h \in [10^{-6}, 10^0]$. Plottet er et logaritmisk plott.

3. Da vi utledet trunkeringsfeilen (3.7), var er det ikke opplagt hva graden til taylorpolynomet i (3.5) skulle være. I denne oppgaven skal du se hva som skjer om man senker graden på taylorpolynomet.

- (a) Gjør taylorutviklingen i (3.5) på nytt, men bruk et taylorpolynom av grad 2 i stedet. Prøv fra dette å utlede en formel for feilen som er tilsvarende (3.6).
- (b) Gjør (a) på nytt, men bruk et taylorpolynom av grad 0, altså bare en konstant funksjon.
- (c) Hvorfor kan du konkludere at det førsteordens taylorpolynomet og formelen for feilen i (3.6) er best?

3.2 Alternativer til newtonkvotienten

Vi startet dette kapitlet med newtonkvotienten, punktmetoden basert på $f(a)$ og $f(a+h)$. Hvis vi har flere punkter, byr det fort på mye regning å finne og derivere det interpolerende polynomet. Som et eksempel, se på punktmetoden basert



Figur 3.3. Sekanten i a (rød kurve) til en funksjon f (svart kurve) basert i $a-h$ og $a+h$, samt tangenten i a (blå kurve). Parabelen som interpolerer f i $a-h$, a , og $a+h$ er også vist (svart kurve)

på $f(a)$, $f(a-h)$, og $f(a+h)$. Bruker vi newtonformen til det interpolerende polynomet får vi fra (2.11) og (2.12) at

$$p_2(x) = f(a-h) + \frac{f(a) - f(a-h)}{h}(x - (a-h)) + \frac{f(a+h) - 2f(a) + f(a-h)}{2h^2}(x - (a-h))(x - a). \quad (3.8)$$

Etter litt regning finner vi at

$$p_2'(a) = \frac{f(a+h) - f(a-h)}{2h} \quad (3.9)$$

Detaljene i (3.8) og (3.9) overlater vi til Oppgave 3. Uttrykket (3.9) er stignings-tallet til sekanten gjennom punktene $a-h$ og $a+h$, og kalles den *symmetriske newtonkvotienten*. Denne er illustrert i Figur 3.3,

Trunkeringsfeilen for den symmetriske Newtonkvotienten kan analyseres på samme måte som Newtonkvotienten. Vi evaluerer nå taylorpolynomet i to punk-

ter i stedet, og tar med andregradsledd også:

$$\begin{aligned}f(a+h) &= f(a) + f'(a)h + \frac{1}{2}f''(a)h^2 + \frac{1}{6}f'''(c_1)h^3 \\f(a-h) &= f(a) - f'(a)h + \frac{1}{2}f''(a)h^2 - \frac{1}{6}f'''(c_2)h^3,\end{aligned}$$

der $c_1 \in [a, a+h]$, $c_2 \in [a-h, a]$. Trekkes disse fra hverandre får vi

$$\frac{f(a+h) - f(a-h)}{2h} - f'(a) = \frac{1}{12}h^2(f'''(c_1) + f'''(c_2)).$$

Tar vi absoluttverdi av begge sider og bruker trekantulikheten får vi at

$$|E_1(f; a, h)| = \left| \frac{1}{12}h^2(f'''(c_1) + f'''(c_2)) \right| \leq \frac{h^2}{12}M_3 + \frac{h^2}{12}M_3 = M_3 \frac{h^2}{6}.$$

Spesielt vil tilnærmingen være eksakt for andregradspolynomer, siden disse har tredjederivert lik 0 (slik at $M_3 = 0$). Dermed blir trunkeringsfeilen 0 for disse. Tilnærmingen er ikke eksakt for tredjegradspolynomer. Tester vi den symmetriske Newtonkvotienten med samme verdier for h som før får vi

h	$(f(a+h) - f(a-h))/(2h)$	$E_1(f; a, h)$
10^{-1}	0.8761206554	1.5×10^{-3}
10^{-2}	0.8775679356	1.5×10^{-5}
10^{-3}	0.8775824156	1.5×10^{-7}
10^{-4}	0.8775825604	1.5×10^{-9}
10^{-5}	0.8775825619	1.8×10^{-11}
10^{-6}	0.8775825619	-7.5×10^{-12}

Når h reduseres med faktor 10, ser vi at feilen blir redusert med omtrent faktor 100. Tilsvarende faktor fra tabellen for newtonkvotienten var 10.

La oss se på et eksempel til, denne gang en punktmetode for den andredriverte. Vi bruker samme punkter som den symmetriske Newtonkvotienten, slik at vi får samme interpolerende polynom som i (3.8). Etter litt mer regning finner vi at

$$p_2''(a) = \frac{f(a+h) - 2f(a) + f(a-h)}{h^2} \quad (3.10)$$

Som over kan vi finne trunkeringsfeilen ved å evaluere Taylorpolynomet i de samme punktene, men nå tar vi også med et tredjegradsledd,

$$\begin{aligned}f(a+h) &= f(a) + f'(a)h + \frac{1}{2}f''(a)h^2 + \frac{1}{6}f'''(a)h^3 + \frac{1}{24}f^{(iv)}(c_1)h^4 \\f(a-h) &= f(a) - f'(a)h + \frac{1}{2}f''(a)h^2 - \frac{1}{6}f'''(a)h^3 - \frac{1}{24}f^{(iv)}(c_2)h^4,\end{aligned}$$

og i stedet for å trekke fra hverandre legger vi de nå sammen. Da får vi

$$f(a-h) - 2f(a) + f(a+h) - f''(a)h^2 = \frac{1}{24}f^{(iv)}(c_1)h^4 - \frac{1}{24}f^{(iv)}(c_2)h^4.$$

Deler vi med h^2 på begge sider får vi trunkeringsfeilen

$$|E_2 f(; a, h)| = \left| \frac{f(a-h) - 2f(a) + f(a+h)}{h^2} - f''(a) \right|$$

på venstre side, og på høyre side får vi

$$M_4 \frac{h^2}{24} + M_4 \frac{h^2}{24} = M_4 \frac{h^2}{12}.$$

For punktmetoder med flere punkter kan man også gjøre som over - kombinere Taylorrekken i de kjente punktene slik at vi på venstre side får et uttrykk for trunkeringsfeilen. I Seksjon 3.3 skal vi utdype dette, og utlede en punktmetode basert på fire punkter som har trunkeringsfeil $|E_1(f; a, h)| \leq \frac{M_5}{18} h^4$. h^4 -faktoren gjør at trunkeringsfeilen går enda raskere mot 0 når h avtar.

Vi har nå estimater på trunkeringsfeilen for fire metoder

1. For Newtonkvotienten: $M_2 \frac{h}{2}$,
2. For den symmetriske Newtonkvotienten: $M_3 \frac{h^2}{6}$,
3. For 4-punkts metoden fra Seksjon 3.3: $\frac{M_5}{18} h^4$,
4. For den andrederiverte: $M_4 \frac{h^2}{12}$.

Vi ser at 4-punkts metoden har lavere trunkeringsfeil enn de to første metodene. Hvis vi bruker funksjonen $f(x) = \sin x$, samt at $M_k \approx (\sin a)^{(k)}$ nær a , så kan vi plote trunkeringsfeilestimatene mot hverandre som følger

```
a=0.5;
h = 10.^(-(0:0.1:6)');
loglog(h, abs(sin(a)*h/2), ...
        h, abs(cos(a)*h.^2/6), ...
        h, abs(18*cos(a)*h.^4), ...
        h, abs(sin(a)*h.^2/12))
legend('Newton', 'Symmetrisk Newton', 'Fire punkter', 'Andrederiverte')
```

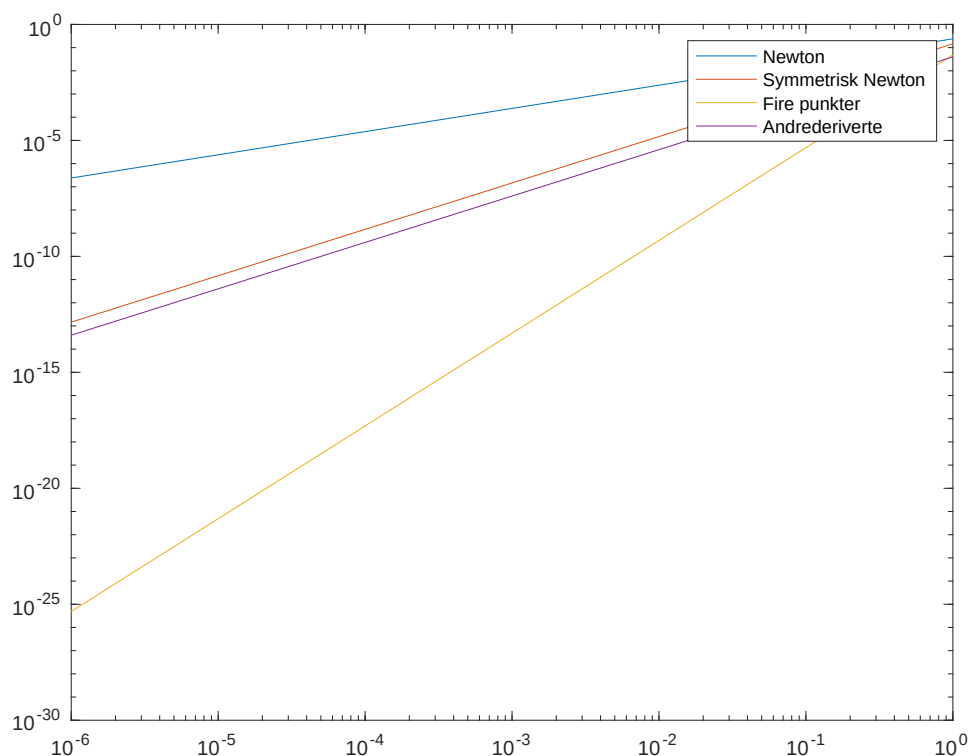
Program 3.1. Sammenligning av trunkeringsfeil

Plottene er vist i Figur 3.4. Dette er et loglog-plott. Merk at, hvis $|E_1(f; a, h)| \approx Ch^p$, så har vi at

$$\log|E_1(f; a, h)| \approx p \log h + \log C.$$

Siden våre metoder har trunkeringsfeil proporsjonal med h , h^2 , og h^4 , så ser vi i loglog-plottet lineære kurver med stigningstall 1, 2, og 4.

Hvis vi baserer oss på enda flere punkter blir alt enda mer komplisert, og vi trenger derfor et resultat som gir oss system i kaoset. Det neste teoremet sier oss, ved hjelp av lineær algebra, hva punktmetoden basert på $f(a + m_0 h)$, $f(a + m_1 h)$, ..., $f(a + m_n h)$ blir. Vi skal bevise dette først i Seksjon 3.3.



Figur 3.4. loglog-plott av estimatene på trunkeringsfeilen for de fire metodene vi ser på, for $f(x) = \sin x$ i $x = 0.5$, og for $h \in [10^{-6}, 10^0]$.

Teorem 3.4. Anta at vi vet $n+1$ funksjonsverdier $\{f(a+m_i h)\}_{i=0}^n$, der $m_0 = 0$ og $m_0, m_1, \dots, m_n$ er forskjellige heltall (vi antar altså spesielt at $f(a)$ skal bidra). La p_n være det interpolerende polynomet av grad $\leq n$ i punktene $\{a+m_i h\}_{i=0}^n$, og la

$$V = \begin{pmatrix} m_1 & m_2 & \cdots & m_n \\ m_1^2 & m_2^2 & \cdots & m_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ m_1^n & m_2^n & \cdots & m_n^n \end{pmatrix}. \quad (3.11)$$

Da har vi at

$$\begin{pmatrix} h^1 p_n^{(1)}(a)/1! \\ h^2 p_n^{(2)}(a)/2! \\ \vdots \\ h^n p_n^{(n)}(a)/n! \end{pmatrix} = (-V^{-T} \mathbf{e} \quad V^{-T}) \begin{pmatrix} f(a) \\ f(a+m_1 h) \\ \vdots \\ f(a+m_n h) \end{pmatrix}. \quad (3.12)$$

(Her er V^{-T} en kort notasjon for $(V^{-1})^T$, og $\mathbf{e}$ er en søylevektor med bare enere.)

Her er $(-V^{-T} \mathbf{e} \quad V^{-T})$ blokkmatrisen der V^{-T} plasseres til høyre for $-V^{-T} \mathbf{e}$

(se Seksjon 1.2).

Det er ikke meningen at du skal huske dette resultatet! Poenget er at vi fra dette faktisk kan lese ut, og lage kode for å finne, tilnærmingene $p_n^{(k)}(a)$ til alle de deriverte! For å utdype dette, la oss se spesielt på tilfellet med punktemetoder basert på de tre punktene $f(a)$, $f(a-h)$, og $f(a+h)$. Dette svarer til at vi setter $n = 2$, $m_1 = -1$, $m_2 = 1$, i teoremet. Dette gir $V = \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix}$, som har invers $V^{-1} = \frac{1}{2} \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix}$. Setter vi inn i (3.12) får vi

$$\begin{aligned} \begin{pmatrix} hp'_n(a) \\ h^2 p''_n(a)/2 \end{pmatrix} &= (-V^{-T} \mathbf{e} \quad V^{-T}) \begin{pmatrix} f(a) \\ f(a-h) \\ f(a+h) \end{pmatrix} \\ &= \frac{1}{2} \begin{pmatrix} 0 & -1 & 1 \\ -2 & 1 & 1 \end{pmatrix} \begin{pmatrix} f(a) \\ f(a-h) \\ f(a+h) \end{pmatrix} \\ &= \frac{1}{2} \begin{pmatrix} f(a+h) - f(a-h) \\ f(a-h) - 2f(a) + f(a+h) \end{pmatrix} \end{aligned}$$

Dette kan vi skrive om til

$$\begin{aligned} p'_n(a) &= \frac{f(a+h) - f(a-h)}{2h} \\ p''_n(a) &= \frac{f(a-h) - 2f(a) + f(a+h)}{h^2} \end{aligned}$$

Vi ser at dette faller sammen med punktmetodene fra (3.9) og (3.10).

Oppgaver for Seksjon 3.2

1. Vi tilnærmer den andrederiverte til en funksjon $f(x)$ i $x = 0$ med

$$D_2 f(0) = \frac{f(h) - 2f(0) + f(-h)}{h^2}$$

Vi antar at f er deriverbar uendelig mange ganger. Finn det korrekte alternativet i følgende flervalgsoppgaver.

(a) Tilnærmingen $D_2 f(0)$ vil alltid være eksakt hvis $f(x)$ er

- ☐ en trigonometrisk funksjon
- ☐ en logaritmisk funksjon
- ☐ et polynom av grad 4
- ☐ et polynom av grad 3

(b) Feilen

$$|f''(0) - D_2 f(0)|$$

er begrenset av

- ☐ $\frac{h^2}{12} \max_{x \in [-h, h]} |f''(x)|$
- ☐ $\frac{h^2}{48} \max_{x \in [-h, h]} |f^{(4)}(x)|$
- ☐ $\frac{h}{4} \max_{x \in [-h, h]} |f''(x)|$
- ☐ $\frac{h^2}{12} \max_{x \in [-h, h]} |f^{(4)}(x)|$

2. Marker hver av følgende påstander som sanne eller gale

(a) Den symmetriske newtonkvotienten er eksakt for polynomer av grad 2 eller lavere.

(b) Selv om den symmetriske newtonkvotienten gir større presisjon, så er det en trade-off siden den krever mye mer regning enn newtonkvotienten.

(c) firepunktsmetoden med steglengde $h = 0.2$ vil vanligvis ha mindre feil enn den symmetriske newtonkvotienten med $h = 0.1$.

(d) firepunktsmetoden er eksakt for alle fjerdegradspolynomer, men ikke for alle femtegradspolynomer.

3. Vis at formlene (3.8) (3.9) er riktige.

4. Vis at den symmetriske newtonkvotienten i $x = a$

(a) er lik $f'(a)$ når f er en kvadratisk funksjon $f(x) = bx^2 + cx + d$.

(b) er lik gjennomsnittet av de to asymmetriske kvotientene

$$f_r(a) = \frac{f(a+h) - f(a)}{h} \quad f_l(a) = \frac{f(a) - f(a-h)}{h}. \quad (3.13)$$

5. I denne oppgaven ser vi på noen andre mulige tilnærminger til den deriverte.

(a) Finn en tilnærming til $f'(a)$ som bare bruker verdiene $f(a-h)$, $f(a)$, og $f(a+2h)$.

(b) Finn en tilnærming til $f'(a)$ som bare bruker verdiene $f(a)$, $f(a+h)$, og $f(a+2h)$.

6. Denne oppgaven illustrerer en annen måte som kan brukes til å utlede metoder for numerisk derivasjon.

(a) Anta vi vil utlede en metode for å tilnærme $f'(a)$ som er på formen

$$f'(a) \approx c_1 f(a-h) + c_2 f(a+h), \quad c_1, c_2 \in \mathbb{R}.$$

Vi vil at metoden skal være eksakt når $f(x) = 1$ og $f(x) = x$. Bruk disse betingelsene til å regne ut c_1 og c_2 .

(b) Vis at metoden i (a) er eksakt for alle polynomer av grad 1, og sammenlign den med metodene vi har utledet i dette kapitlet.

(c) Bruk metodene i (a) og (b) til å utlede en metode for å tilnærme $f''(a)$.

$$f''(a) \approx c_1 f(a-h) + c_2 f(a) + c_3 f(a+h), \quad c_1, c_2, c_3 \in \mathbb{R},$$

ved å kreve at metodene skal være eksakt for $f(x) = 1$, x , og x^2 . Kjenner du igjen metoden?

(d) Vis at metoden i (c) er eksakt for alle kubiske polynomer.

7. I denne oppgaven skal vi sammenligne den symmetriske newtonkvotienten med de to asymmetriske newtonkvotientene.

(a) Tegn grafen til funksjonen

$$f(x) = \frac{-x^2 + 10x - 5}{4}$$

over intervallet $[0, 6]$, sammen med de tre sekantene som hører til de tre tilnærmingene til de deriverte fra (a) (bruk $a = 3$ og $h = 2$). Kan du fra dette avgjøre hvilken tilnærming som er best?

(b) Regn ut de tre tilnærmingene til de deriverte i (a) numerisk for $f(x)$ med $a = 3$ og $h_1 = 0.1$ og $h_2 = 0.001$. Hva er de relative feilene?

3.3 Trunkeringsfeilen for generelle punktmetoder (*)

Vi starter denne seksjonen med et bevis for Teorem 3.4.

Bevis. Vi starter med å uttrykke p_n som et n -te ordens taylorpolynom om a :

$$p_n(x) = f(a) + c_1(x-a) + \cdots + c_n(x-a)^n.$$

Evaluerer vi dette i $x = a + m_i h$ for $i = 1, 2, \dots, n$ får vi likningene

$$\begin{pmatrix} m_1 h & (m_1 h)^2 & \cdots & (m_1 h)^n \\ m_2 h & (m_2 h)^2 & \cdots & (m_2 h)^n \\ \vdots & \vdots & \ddots & \vdots \\ m_n h & (m_n h)^2 & \cdots & (m_n h)^n \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} f(a + m_1 h) - f(a) \\ f(a + m_2 h) - f(a) \\ \vdots \\ f(a + m_n h) - f(a) \end{pmatrix}.$$

Siden $c_k = p_n^{(k)}(a)/k!$, og med V definert som over, kan dette også skrives

$$V^T \begin{pmatrix} h^1 p_n^{(1)}(a)/1! \\ h^2 p_n^{(2)}(a)/2! \\ \vdots \\ h^n p_n^{(n)}(a)/n! \end{pmatrix} = \begin{pmatrix} f(a + m_1 h) - f(a) \\ f(a + m_2 h) - f(a) \\ \vdots \\ f(a + m_n h) - f(a) \end{pmatrix}.$$

Siden kvadratiske vandermondematriser med forskjellige generatorer (alle ulik 0) er inverterbare (se Teorem (2.1)), får vi

$$\begin{aligned} \begin{pmatrix} h^1 p_n^{(1)}(a)/1! \\ h^2 p_n^{(2)}(a)/2! \\ \vdots \\ h^n p_n^{(n)}(a)/n! \end{pmatrix} &= V^{-T} \begin{pmatrix} f(a + m_1 h) - f(a) \\ f(a + m_2 h) - f(a) \\ \vdots \\ f(a + m_n h) - f(a) \end{pmatrix} \\ &= -V^{-T} \mathbf{e} f(a) + V^{-T} \begin{pmatrix} f(a + m_1 h) \\ f(a + m_2 h) \\ \vdots \\ f(a + m_n h) \end{pmatrix} \\ &= (-V^{-T} \mathbf{e} \quad V^{-T}) \begin{pmatrix} f(a) \\ f(a + m_1 h) \\ \vdots \\ f(a + m_n h) \end{pmatrix} \end{aligned} \quad (3.14)$$

I den siste overgangen her brukte vi at $AC + BD = \begin{pmatrix} A & B \end{pmatrix} \begin{pmatrix} C \\ D \end{pmatrix}$ når A og B har like mange rader, og C og D har like mange søyler, og matrisemultiplikasjonene er definert. ■

Vi skal også vise følgende feilestimat.

Teorem 3.5. La V være matrisen fra (3.11). Vi har at

$$|E_k(f; a, h)| \leq B h^{n+1-k} M_{n+1}, \quad (3.15)$$

der

$$B = \frac{k! \sum_{i=1}^n |b_i| |m_i|^{n+1}}{(n+1)!}, \quad (3.16)$$

og der $\mathbf{b}$ er søyle k i V^{-1} .

Det følger som over at en $(n+1)$ -punkts metode er eksakt for alle polynomer av grad $\leq n$. Legg merke til at trepunkts-tilnærmingen til den andrederiverte hadde en grad høyere i Taylorrekka enn antall punkter. Dette fanges ikke opp

av teoremet over. Det er rett-fram å generalisere teoremet over slik at det fanger opp dette.

Bevis. Vi starter med å skrive opp følgende Taylorutviklinger

$$\begin{aligned} f(a + m_1 h) - f(a) &= f'(a)m_1 h + \dots + \frac{f^{(n)}(a)}{n!}(m_1 h)^n + \frac{f^{(n+1)}(c_1)}{(n+1)!}(m_1 h)^{n+1} \\ f(a + m_2 h) - f(a) &= f'(a)m_2 h + \dots + \frac{f^{(n)}(a)}{n!}(m_2 h)^n + \frac{f^{(n+1)}(c_2)}{(n+1)!}(m_2 h)^{n+1} \\ &\vdots \\ f(a + m_n h) - f(a) &= f'(a)m_n h + \dots + \frac{f^{(n)}(a)}{n!}(m_n h)^n + \frac{f^{(n+1)}(c_n)}{(n+1)!}(m_n h)^{n+1}. \end{aligned} \quad (3.17)$$

Med V definert ved (3.11) så kan dette skrives på matriseform som

$$V^T \begin{pmatrix} h^1 f'(a)/1! \\ h^2 f''(a)/2! \\ \vdots \\ h^n f^{(n)}(a)/n! \end{pmatrix} + \begin{pmatrix} \frac{f^{(n+1)}(c_1)}{(n+1)!}(m_1 h)^{n+1} \\ \frac{f^{(n+1)}(c_2)}{(n+1)!}(m_2 h)^{n+1} \\ \vdots \\ \frac{f^{(n+1)}(c_n)}{(n+1)!}(m_n h)^{n+1} \end{pmatrix} = \begin{pmatrix} f(a + m_1 h) - f(a) \\ f(a + m_2 h) - f(a) \\ \vdots \\ f(a + m_n h) - f(a) \end{pmatrix},$$

Ganger vi med $\mathbf{b}^T$ til venstre over får vi

$$\frac{h^k}{k!} f^{(k)}(a) + \mathbf{b}^T \begin{pmatrix} \frac{f^{(n+1)}(c_1)}{(n+1)!}(m_1 h)^{n+1} \\ \frac{f^{(n+1)}(c_2)}{(n+1)!}(m_2 h)^{n+1} \\ \vdots \\ \frac{f^{(n+1)}(c_n)}{(n+1)!}(m_n h)^{n+1} \end{pmatrix} = \mathbf{b}^T \begin{pmatrix} f(a + m_1 h) - f(a) \\ f(a + m_2 h) - f(a) \\ \vdots \\ f(a + m_n h) - f(a) \end{pmatrix} = \frac{h^k}{k!} p_n^{(k)}(a),$$

der vi brukte (3.14). Dette gir

$$p_n^{(k)}(a) - f^{(k)}(a) = \frac{k!}{h^k} \mathbf{b}^T \begin{pmatrix} \frac{f^{(n+1)}(c_1)}{(n+1)!}(m_1 h)^{n+1} \\ \frac{f^{(n+1)}(c_2)}{(n+1)!}(m_2 h)^{n+1} \\ \vdots \\ \frac{f^{(n+1)}(c_n)}{(n+1)!}(m_n h)^{n+1} \end{pmatrix}$$

Hvis vi tar absoluttverdier og bruker trekantulikheten får vi

$$\left| p_n^{(k)}(a) - f^{(k)}(a) \right| = \left| \frac{k!}{h^k} \sum_{i=1}^n b_i \frac{f^{(n+1)}(c_i)}{(n+1)!} (m_i h)^{n+1} \right|.$$

Hvis vi bruker definisjonen av $E_k(f; a, h)$ og den øvre skranken M_{n+1} for den deriverte av orden $n+1$, så får vi (3.15). ■

Fire-punkts metoden som vi tidligere nevnte er basert på punktene $f(a-2h)$, $f(a-h)$, $f(a)$, $f(a+h)$, og $f(a+h)$. Det vil si at vi setter $n = 4$, $m_0 = 0$, $m_1 = -2$, $m_2 = -1$, $m_3 = 1$, $m_4 = 2$. Vi finner (b_1, b_2, b_3, b_4) ved å løse systemet

$$\begin{pmatrix} -2 & -1 & 1 & 2 \\ 4 & 1 & 1 & 4 \\ -8 & -1 & 1 & 8 \\ 16 & 1 & 1 & 16 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Her er det et poeng å løse dette eksakt. Dette kan gjøres med Symbolic Math Toolbox:

```
>> rref(sym([-2 -1 1 2 1; 4 1 1 4 0; -8 -1 1 8 0; 16 1 1 16 0]))

ans =

[1, 0, 0, 0, 1/12]
[0, 1, 0, 0, -2/3]
[0, 0, 1, 0, 2/3]
[0, 0, 0, 1, -1/12]
```

slik at $\mathbf{b} = (1/12, -2/3, 2/3, -1/12)$. Vi får at første element i $V^{-T}\mathbf{e}$ blir

$$\begin{pmatrix} 1/12 & -2/3 & 2/3 & -1/12 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = 0.$$

Det følger nå fra (3.12) at

$$hp'_4(a) = \begin{pmatrix} 1/12 & -2/3 & 2/3 & -1/12 \end{pmatrix} \begin{pmatrix} f(a-2h) \\ f(a-h) \\ f(a+h) \\ a(a+2h) \end{pmatrix},$$

slik at metoden blir

$$f'(a) \approx p'_4(a) = \frac{f(a-2h) - 8f(a-h) + 8f(a+h) - f(a+2h)}{12h}.$$

Setter vi inn i (3.16) får vi at $B = 1/18$. (3.15) gir slutt estimatet $|E_1(f; a, h)| \leq M_5 \frac{h^4}{18}$ for trunkeringsfeilen.

KAPITTEL 4

Numerisk integrasjon

I numerisk derivasjon brukes kjente funksjonsverdier til å finne tilnærminger til den derivate av en funksjon. I numerisk *integrasjon* brukes samme informasjon til å finne tilnærminger til *integralet*. Metodene er nyttige når funksjonsverdiene bare er kjent i isolerte punkter, eller når det er vanskelig eller umulig å beregne integralet for hånd.

I Kapittel 3 laget vi en generell strategi for å utlede metoder for numerisk derivasjon. Den tilsvarende strategien for numerisk integrasjon er som følger: Finn polynomet som interpolerer funksjonen i et passende sett av punkter, og bruk integralet av dette polynomet som en tilnærming til integralet av funksjonen. Trunkeringsfeilen kan da analyseres på omtrent samme måte som for numerisk derivasjon.

Vi starter dette kapitlet med å gå gjennom definisjonen av integralet (Seksjon 4.1). Deretter ser vi på den enkleste metoden for numerisk integrasjon, midtpunktsmetoden (Seksjon 4.2). I Seksjon 4.3 formulerer vi så den ovennevnte generelle strategien i litt mer detalj, og ser på noen viktige spesialtilfeller.

4.1 Hva er integralet?

Integralet av en funksjon $f: [a, b] \rightarrow \mathbb{R}$ tolker vi som arealet under grafen, der de positive verdiene til f bidrar positivt til arealet, og de negative verdiene bidrar negativt (se Figur 4.1). Standard notasjon for integralet er

$$\int_a^b f(x) dx.$$

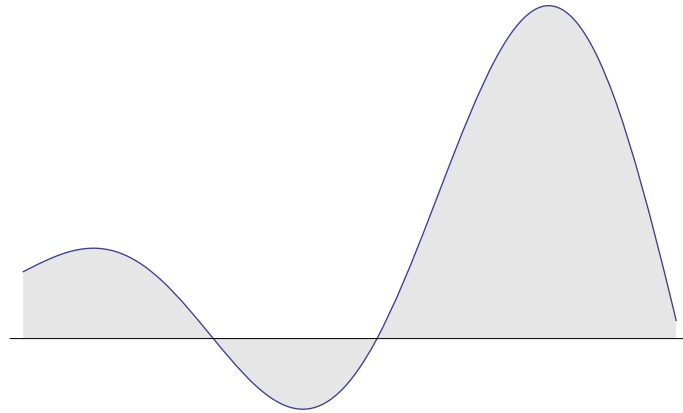
Før vi fortsetter må vi introdusere et begrep som vi skal bruke flere ganger i vår definisjon av integralet.

Definisjon 4.1 (Partisjon). La a og b være to reelle tall med $a < b$. En partisjon av $[a, b]$ er en samling av tall $\{x_0, x_1, \dots, x_n\}$ slik at

$$a = x_0 < x_1 < x_2 < \dots < x_{n-1} < x_n = b.$$

Partisjonen sies å være uniform dersom $x_i - x_{i-1} = (b - a)/n$ for $i = 1, \dots, n$. Avstanden mellom punktene, $h = (b - a)/n$, kalles da steglengden til partisjonen.

Om $\Pi = \{x_0, x_1, \dots, x_n\}$ er en partisjon, lar vi *størrelsen* på partisjonen være $|\Pi| = \max_{i=1, \dots, n} |x_i - x_{i-1}|$, altså den største avstanden mellom nabopunkter. Hvis Π er uniform vil $|\Pi| = (b - a)/n$.



Figur 4.1. Arealet under grafen til en funksjon.

4.1.1 Definisjonen av integralet

For å definere integralet bruker man såkalte *Riemann-summer*, som er numeriske tilnærminger av arealet under grafen. Vi velger en partisjon $\Pi = \{x_0, x_1, \dots, x_n\}$ av $[a, b]$, og finner maksimum og minimum til f på hvert delintervall $[x_{i-1}, x_i]$,

$$m_i = \min_{x \in [x_{i-1}, x_i]} f(x), \quad M_i = \max_{x \in [x_{i-1}, x_i]} f(x) \quad \text{for } i = 1, 2, \dots, n.$$

Dersom f er kontinuerlig, vil disse verdiene alltid eksistere, så vi antar heretter kontinuitet. Vi kan så gjøre to forskjellige tilnærminger til integralet ved å tilnærme f med to forskjellige *stegfunksjoner*, som begge er konstant på hvert delintervall $[x_{i-1}, x_i]$: Den første er lik m_i på dette delintervallet, den andre er lik M_i (se Figur 4.2). Deretter summerer vi opp arealene under de to stegfunksjonene, og ender opp med de to tilnærmingene

$$\underline{I}(\Pi) = \sum_{i=1}^n m_i(x_i - x_{i-1}), \quad \bar{I}(\Pi) = \sum_{i=1}^n M_i(x_i - x_{i-1}). \quad (4.1)$$

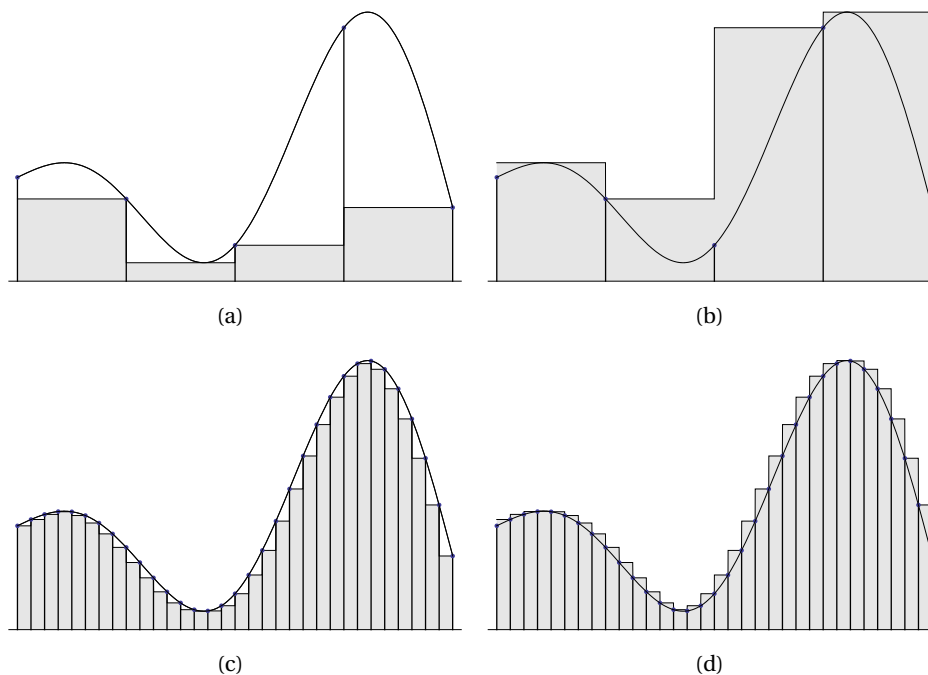
Generelt er den første tilnærmingen for liten, den andre for stor. Integralet får vi nå av å la størrelsen på partisjonen gå mot 0:

Definisjon 4.2 (Integralet). La f være en kontinuerlig funksjon definert på intervallet $[a, b]$. Hvis grenseverdiene

$$\lim_{|\Pi| \rightarrow 0} \underline{I}(\Pi) \quad \text{og} \quad \lim_{|\Pi| \rightarrow 0} \bar{I}(\Pi)$$

(der grensen tas over følger av partisjoner med synkende størrelse) begge eksisterer og er like, sier vi at f er integrerbar, og vi definerer integralet av f over $[a, b]$ som

$$I = \int_a^b f(x) dx = \lim_{|\Pi| \rightarrow 0} \underline{I}(\Pi) = \lim_{|\Pi| \rightarrow 0} \bar{I}(\Pi).$$



Figur 4.2. Innskrevne (til venstre) og omskrevne (til høyre) stegfunksjoner for en grov partisjon (øverst) og en fin partisjon (nederst).

Det viser seg at *enhver* kontinuertlig funksjon på et lukket intervall er integrerbar. Et bevis for dette finner du i *Kalkulus*. Konstruksjonen av integralet illustreres i Figur 4.2, der vi ser hvordan de stykkevis konstante tilnærmingerne blir bedre når partisjonen blir mindre.

Definisjonen over kan brukes som en numerisk metode for å finne tilnærminger til integralet: Vi velger å jobbe med enten maksimum eller minimum, velger en partisjon av $[a, b]$ som i Figur 4.2, og legger sammen arealene av rektanglene. Problemet med denne metoden er at det kan både være vanskelig og tidkrevende å finne maksimum og minimum, selv på en datamaskin. Men det viser seg at integralet har en egenskap som er veldig nyttig når det kommer til numeriske beregninger:

Teorem 4.3. La $f: [a, b] \rightarrow \mathbb{R}$ være kontinuertlig, la $\{x_0, \dots, x_n\}$ være en partisjon av $[a, b]$, og la t_i være et tall i $[x_{i-1}, x_i]$ for hver $i = 1, \dots, n$. Da vil summen

$$I = \sum_{i=1}^n f(t_i)(x_i - x_{i-1}) \quad (4.2)$$

konvergere mot integralet når størrelsen på partisjonene går mot null.

Bevis. Siden f er kontinuertlig, er den integrerbar, så tilnærmingerne $\underline{I}$ og $\bar{I}$ kon-

vergerer begge mot $\int_a^b f(x) dx$ når størrelsen på partisjonene går mot null. For enhver partisjon vil $\underline{I} \leq I \leq \bar{I}$, fordi $\min_{x \in [x_{i-1}, x_i]} f(x) \leq f(t_i) \leq \max_{x \in [x_{i-1}, x_i]} f(x)$ for hver $i = 1, \dots, n$. Dermed vil også I konvergere mot $\int_a^b f(x) dx$. ■

Generelt kan vi utlede metoder for numerisk integrasjon ved å dele opp intervallet $[a, b]$ i små delintervaller, tilnærme f med et polynom på hvert delintervall, integrere dette polynomet i stedet for f , og til slutt legge sammen bidragene fra hvert delintervall. Vi skal bruke denne strategien til å utlede mer avanserte metoder for numerisk integrasjon, og dette vil fungere så lenge f kan tilnærmes godt med polynomer på hvert delintervall.

Oppgaver for Seksjon 4.1

1. I denne oppgaven skal vi bruke definisjonen av integralet på funksjonen $f(x) = e^x$ på intervallet $[0, 1]$.

(a) Skriv og kjør et program som regner ut $\underline{I}$ og $\bar{I}$ for en uniform partisjon på 10 delintervaller (la metoden ta antall delintervaller som argument). Hva blir feilene fra den eksakte verdien $I = e - 1 = 1.718281828$ av integralet?

Hint: I stedet for en for-løkke kan du bruke sum-kommandoen når du legger sammen bidragene fra delintervallene.

(b) Lag et program som finner hvor mange delintervaller som trengs for å regne ut $\underline{I}$ med en feil mindre enn 3×10^{-3} ?

4.2 Midtpunktsmetoden

Ved hjelp av Teorem 4.3 kan vi utlede praktiske, numeriske metoder for å regne ut integralet: Vi lager en partisjon av $[a, b]$, velger t_i lik x_{i-1} eller x_i , og regner ut summen (4.2). Det viser seg at et enda bedre valg er midtpunktet $t_i = (x_i + x_{i-1})/2$, som gir oss tilnærmingen

$$I \approx \sum_{i=1}^n f((x_i + x_{i-1})/2)(x_i - x_{i-1}). \quad (4.3)$$

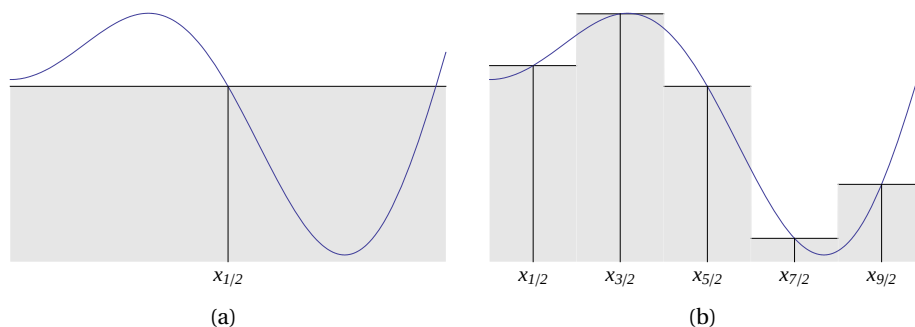
Dette er den såkalte *midtpunktsmetoden*. Vi kan tenke på dette som en konstant tilnærming på hvert delintervall, der konstanten er verdien i midtpunktet.

Algoritme 4.4. La f være en funksjon som er integrerbar på intervallet $[a, b]$, og la $\{x_i\}_{i=0}^n$ være en uniform partisjon av $[a, b]$. Midtpunktsmetoden tilnærmer integralet av f med

$$\int_a^b f(x) dx \approx I_{mid}(h) = h \sum_{i=1}^n f(x_{i-1/2}), \quad (4.4)$$

der

$$x_{i-1/2} = (x_{i-1} + x_i)/2 = a + (i - 1/2)h.$$



Figur 4.3. Midtpunktsmetoden med ett delintervall (a), og fem delintervaller (b).

Metoden er illustrert i Figur 4.3 for tilfellene med ett og fem delintervaller.

Eksempel 4.5. La oss teste midtpunktsmetoden på en funksjon der vi vet svaret. Vi velger integralet

$$\int_0^1 \cos x \, dx = \sin 1 \approx 0.8414709848$$

Det eksakte svaret er da $\sin 1$. Vi tester metoden ved å splitte intervallet i 2^k delintervaller, for $k = 1, 2, \dots, 10$, d.v.s. vi halverer steglengden i hver iterasjon. Vi får da

h	$I_{\text{mid}}(h)$	$I - I_{\text{mid}}(h)$
0.500000	0.85030065	-8.8×10^{-3}
0.250000	0.84366632	-2.2×10^{-3}
0.125000	0.84201907	-5.5×10^{-4}
0.062500	0.84160796	-1.4×10^{-4}
0.031250	0.84150523	-3.4×10^{-5}
0.015625	0.84147954	-8.6×10^{-6}
0.007813	0.84147312	-2.1×10^{-6}
0.003906	0.84147152	-5.3×10^{-7}
0.001953	0.84147112	-1.3×10^{-7}
0.000977	0.84147102	-3.3×10^{-8}

Vi ser at hver gang h is halveres, så blir feilen $|I - I_{\text{mid}}(h)|$ redusert omtrent med faktor 4.

4.2.1 Feilestimat for midtpunktsmetoden

Midtpunktsmetoden (4.4) er en tilnærming, og vil medføre en liten feil. Som i kapitlet om numerisk derivasjon skal vi nå *estimere* hvor stor denne feilen kan bli. Dette gjør vi i to steg. Først skal vi analysere feilen i tilfellet med bare ett delintervall. Dette kaller vi den *lokale feilen*. Deretter bruker vi dette til å finne et estimat for det generelle tilfellet med mange delintervaller. Dette kaller vi den *globale feilen*.

Lokal feilanalyse

Feilen i midtpunktsmetoden med bare ett delintervall er

$$E = \int_a^b f(x) dx - f(m)(b-a), \quad (4.5)$$

der $m = (a+b)/2$. Taylorutvikler vi $f(x)$ om midtpunktet m får vi

$$f(x) = f(m) + (x-m)f'(m) + \frac{(x-m)^2}{2}f''(\xi),$$

der ξ er et tall mellom m og x , som avhenger av x . Integrerer vi dette får vi

$$\begin{aligned} \int_a^b f(x) dx &= \int_a^b \left(f(m) + (x-m)f'(m) + \frac{(x-m)^2}{2}f''(\xi) \right) dx \\ &= f(m)(b-a) + \frac{f'(m)}{2} [(x-m)^2]_a^b + \frac{1}{2} \int_a^b (x-m)^2 f''(\xi) dx \\ &= f(m)(b-a) + \frac{1}{2} \int_a^b (x-m)^2 f''(\xi) dx, \end{aligned}$$

der leddet i midten var 0. Dette leder oss til uttrykket

$$|E| = \left| \int_a^b f(x) dx - f(m)(b-a) \right| = \frac{1}{2} \left| \int_a^b (x-m)^2 f''(\xi) dx \right|.$$

Litt mer regning på dette gir

$$\begin{aligned} |E| &= \frac{1}{2} \left| \int_a^b (x-m)^2 f''(\xi) dx \right| \leq \frac{1}{2} \int_a^b |(x-m)^2 f''(\xi)| dx \\ &= \frac{1}{2} \int_a^b (x-m)^2 |f''(\xi)| dx \\ &\leq \frac{M_2}{2} \int_a^b (x-m)^2 dx \\ &= \frac{M_2}{2} \frac{1}{3} [(x-m)^3]_a^b \\ &= \frac{M_2}{6} ((b-m)^3 - (a-m)^3) \\ &= \frac{M_2}{24} (b-a)^3, \end{aligned}$$

der $M_2 = \max_{x \in [a,b]} |f''(x)|$. Ulikheten på første linje holder fordi når vi flytter absoluttverdien innenfor integralet, så blir funksjonen vi integrerer ikkenegativ overalt. Dette betyr at arealer der integranden i det opprinnelige uttrykket var negativ, nå er blitt positive, slik at det andre integralet er større enn det første.

Likheten på andre linje holder siden $(x-m)^2$ aldri er negativ, og ulikheten på tredje linje holder siden vi erstatter $|f''(\xi)|$ med sitt maksimum på intervallet $[a, b]$. På fjerde linje regner vi bare ut integralet av $(x-m)^2$. Så setter vi inn grensene. Siste linje følger av at $(b-m)^3 = -(a-m)^3 = (b-a)^3/8$. Vi har dermed bevist følgende lemma.

Lemma 4.6. La $f: [a, b] \rightarrow \mathbb{R}$ være en kontinuerlig funksjon med kontinuerlige første- og andreordens deriverte. Feilen i midtpunktsmetoden med ett delintervall er da begrenset av

$$\left| \int_a^b f(x) dx - f(m)(b-a) \right| \leq \frac{M_2}{24} (b-a)^3,$$

der $m = (a+b)/2$.

Det viktige i Lemma 4.6 er faktoren $(b-a)^3$. Denne medfører at dersom vi halverer bredden på intervallene, så vil feilen i metoden reduseres med faktor 8.

Global feilanalyse

I forrige seksjon så vi på feilen for ett delintervall. La oss nå se på hva som skjer når vi legger sammen bidragene fra mange delintervaller. Generelt har vi en partisjon av $[a, b]$ i n delintervaller, alle med lengde $h = (b-a)/n$. Med

$$I = \int_a^b f(x) dx = \sum_{i=1}^n \int_{x_{i-1}}^{x_i} f(x) dx, \quad I_{\text{mid}}(h) = \sum_{i=1}^n f(x_{i-1/2})h,$$

blir den totale feilen i midtpunktsmetoden

$$I - I_{\text{mid}}(h) = \sum_{i=1}^n \left(\int_{x_{i-1}}^{x_i} f(x) dx - f(x_{i-1/2})h \right),$$

der uttrykket inne i parentesene er den lokale feilen på intervallet $[x_{i-1}, x_i]$. Vi har derfor

$$\begin{aligned} |I - I_{\text{mid}}(h)| &= \left| \sum_{i=1}^n \left(\int_{x_{i-1}}^{x_i} f(x) dx - f(x_{i-1/2})h \right) \right| \\ &\leq \sum_{i=1}^n \left| \int_{x_{i-1}}^{x_i} f(x) dx - f(x_{i-1/2})h \right| \\ &\leq \sum_{i=1}^n \frac{h^3}{24} M_2^{(i)} \end{aligned}$$

der $M_2^{(i)} = \max_{x \in [x_{i-1}, x_i]} |f''(x)|$. I den første ulikheten har vi brukt trekantulikheten, mens vi i den andre ulikheten har brukt Lemma 4.6. Vi kan forenkle (4.6) ytterligere ved å bruke maksimum over hele $[a, b]$, i stedet for over bare $[x_{i-1}, x_i]$: For alle i har vi at

$$M_2^{(i)} = \max_{x \in [x_{i-1}, x_i]} |f''(x)| \leq \max_{x \in [a, b]} |f''(x)| = M_2,$$

slik at (4.6) kan forenkles til

$$|I - I_{\text{mid}}(h)| \leq \sum_{i=1}^n \frac{h^3}{24} M_2^{(i)} \leq \sum_{i=1}^n \frac{h^3}{24} M_2 = \frac{h^3}{24} n M_2. \quad (4.6)$$

Siden $h = (b-a)/n$, får vi $hn = b-a$. Hvis vi setter dette inn i (4.6), får vi følgende estimat for den globale feilen.

Teorem 4.7. Anta at $f: [a, b] \rightarrow \mathbb{R}$ og dens første to deriverte er kontinuerlige. Da er feilen i midtpunktsmetoden begrenset av

$$|I - I_{mid}(h)| \leq (b-a) \frac{h^2}{24} M_2. \quad (4.7)$$

Dette bekrefter oppførselen vi så for feilen i Eksempel 4.5: Hvis h reduseres med faktor 2, så reduseres feilen med faktor $2^2 = 4$.

4.2.2 Estimering av steglengden

La oss si du ønsker å tilnærme et integral, og du ønsker at feilen i tilnærmingen ikke er større enn en øvre grense ϵ . Vi kan bruke feilestimatet (4.7) ved å kreve at

$$(b-a) \frac{h^2}{24} \max_{x \in [a, b]} M_2 \leq \epsilon$$

som vi lett kan løse for h ,

$$h \leq \sqrt{\frac{24\epsilon}{(b-a)M_2}}.$$

Dersom vi klarer å beregne (eller estimere) M_2 , maksimumsverdien til den andreriverte, kan vi sette inn i uttrykket ovenfor, og finne en øvre grense for h som garanterer at integrasjonsfeilen blir mindre enn ϵ .

Oppgaver for Seksjon 4.2

1. Markér hver av følgende påstander som sanne eller gale.

(a) Midtpunktsmetoden er eksakt for alle polynomer av grad 1.

(b) Midtpunktsmetoden er eksakt for alle polynomer av grad 2.

(c) Den globale feilen i midtpunktsmetoden er av orden en lavere enn den lokale feilen.

(d) Når vi reduserer steglengden h i midtpunktsmetoden med faktor 3, så reduseres feilen med omtrent faktor 9.

2. Vi bruker midtpunktsmetoden med 2 delintervaller for å finne en tilnærming til integralet

$$\int_0^1 x^2 dx.$$

Hva får vi?

☐ 5/16

☐ 1/4

☐ 4/9

☐ 2/5

3. Finn en tilnærming til

$$\int_0^{\pi/2} \frac{\sin x}{1+x^2} dx = 0.526978557614\dots$$

ved hjelp av midtpunktsmetoden med 6 delintervaller.

4. (a) Test midtpunktsmetoden med 10 delintervaller på integralet

$$\int_0^1 e^x dx = e - 1.$$

(b) Finn en verdi av h som garanterer at den absolutte feilen er mindre enn 10^{-10} .

5. Gjenta forrige oppgave, men med integralet byttet ut med

$$\int_2^6 \ln x dx = \ln(11664) - 4.$$

4.3 Alternativer til midtpunktsmetoden

Det finnes også mange andre metoder for numerisk integrasjon. Følgende strategi for å lage slike er en direkte parallell til Strategi 3.1 for numerisk derivasjon.

Strategi 4.8. La $n, k \geq 1$. En approksimasjon av

$$I = \int_a^b f(x) dx,$$

konstrueres som følger:

1. Del intervallet $[a, b]$ i nk like store delintervaller

$$a = x_0 < x_1 < x_2 < \dots < x_{nk-1} < x_{nk} = b.$$

Sett $h = (b - a) / n$.

2. For hvert av delintervallene $J_i = [x_{(i-1)k}, x_{ik}]$, finn polynomet $p_{k,i}$ av grad k som interpolerer f i de $k+1$ punktene $\{x_{(i-1)k}, \dots, x_{ik}\}$.
3. Bruk $\int_{J_i} p_{k,i}(x) dx$ som en approksimasjon til $\int_{J_i} f(x) dx$, og legg sammen disse:

$$I(h) = \sum_{i=1}^n \int_{J_i} p_{k,i}(x) dx.$$

Strategi 4.8 kalles også *Newton–Cotes-metoden*. Ved hjelp av feilanalysen i Kapittel 2 kan vi estimere feilen i denne metoden:

Teorem 4.9. La $n, k \geq 1$ og la $f: [a, b] \rightarrow \mathbb{R}$ være en funksjon med $k+1$ kontinuerlige deriverte. Da vil feilen i Newton–Cotes-metoden av orden k med steglengde $h = (b-a)/n$ være høyst

$$|I - I(h)| \leq \frac{(b-a)M_{k+1}}{(k+1)!} h^{k+1}. \quad (4.8)$$

Bevis. Velg et delintervall J_i , og la $p_{k,i}$ være polynomet av orden k som interpolerer f i punktene $x_{ik}, \dots, x_{(i+1)k}$ (som beskrevet i Strategi 4.8). Fra Teorem 2.8 vet vi at for $x \in J_i$ vil

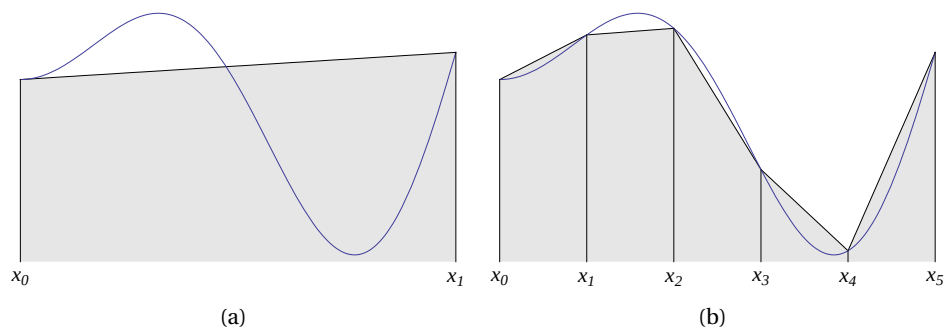
$$|f(x) - p_{k,i}(x)| \leq \frac{M_{k+1}}{(k+1)!} (x_{(i+1)k} - x_{ik})^{k+1} = \frac{M_{k+1}}{(k+1)!} h^{k+1}.$$

Dermed er

$$\begin{aligned} |I - I(h)| &= \left| \int_a^b f(x) dx - \sum_{i=1}^n \int_{J_i} p_{k,i}(x) dx \right| = \left| \sum_{i=1}^n \int_{J_i} f(x) - p_{k,i}(x) dx \right| \\ &\leq \sum_{i=1}^n \int_{J_i} |f(x) - p_{k,i}(x)| dx \\ &\leq \sum_{i=1}^n \int_{J_i} \frac{M_{k+1}}{(k+1)!} h^{k+1} dx \\ &= \frac{M_{k+1}}{(k+1)!} \sum_{i=1}^n h^{k+2} \\ &= \frac{M_{k+1}}{(k+1)!} n h^{k+2}. \end{aligned}$$

(Fra første til andre linje brukte vi trekantulikheten, så satt vi inn feilestimatet for interpolanten, og så integrerte vi konstanten $M_{k+1}/(k+1)!h^{k+1}$.) Om vi nå setter inn $h = (b-a)/n$ får vi (4.8). ■

Av feilestimatet (4.8) ser vi at for å senke feilen, kan vi øke k , senke h (det vil si øke n), eller en kombinasjon av disse. Hvilken kombinasjon som er mest effektiv kommer an på hvor stor faktoren M_{k+1} er for forskjellige k .



Figur 4.4. Trapesmetoden med ett delintervall (a) og fem delintervaller (b).

4.3.1 Trapesmetoden ($k = 1$)

Vi vet at linja som interpolerer f i to punkter x_{i-1} og x_i er

$$p_{1,i}(x) = f(x_{i-1}) + \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}(x - x_{i-1}).$$

Bruker vi denne i Strategi 4.8 får vi *trapesmetoden*; som vi ser av Figur 4.4 tegner tilnærmingen i hvert delintervall J_i et trapes. Vi får da tilnærmingen

$$\int_{x_{i-1}}^{x_i} p_{1,i}(x) dx = \frac{f(x_{i-1}) + f(x_i)}{2} h.$$

Legger vi sammen bidragene fra alle delintervallene får vi approksimasjonen

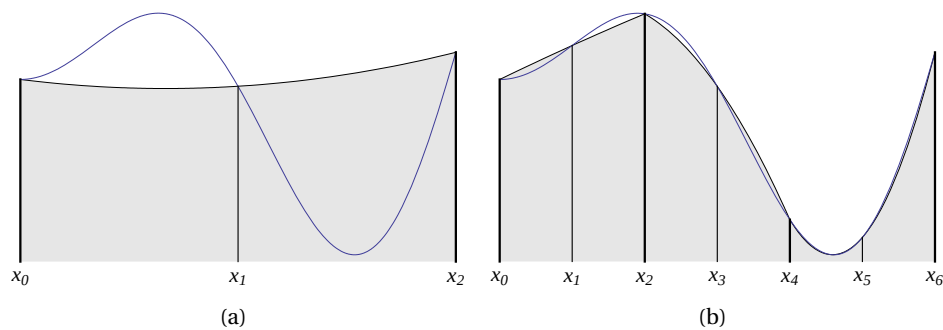
$$I_{\text{trap}}(h) = \sum_{i=1}^n \frac{f(x_{i-1}) + f(x_i)}{2} h. \quad (4.9)$$

Om vi anvender Teorem 2.8 med $k = 1$ får vi følgende feilestimat for trapesmetoden.

Teorem 4.10 (Feilestimat for trapesmetoden). *La $f: [a, b] \rightarrow \mathbb{R}$ være en to ganger kontinuerlig deriverbar funksjon. Feilen i trapesmetoden med steglengde h er da begrenset av*

$$|I - I_{\text{trap}}(h)| \leq \frac{(b-a)M_2}{2} h^2. \quad (4.10)$$

Med litt mer regning kan man vise at trapesmetoden har en feil som er høyst $\frac{(b-a)M_2}{12} h^2$, altså seks ganger lavere enn høyresiden i (4.10) (se Oppgave 7). Feilestimatet (4.10) er altså et *overestimat*. Denne konstante faktoren på $\frac{1}{6}$ er ikke spesielt viktig; det viktige i (4.10) er



Figur 4.5. Simpsons metode med ett delintervall (a) og tre delintervaller (b).

- faktoren h^2 , som gjør at hvis vi for eksempel halverer h , vil feilen gå ned med en faktor $1/4$
- faktoren M_2 , som sier at feilen blir større jo mer f oscillerer (det vil si, jo større f'' er).

Eksempel 4.11. La oss gjenta Eksempel 4.5 med trapesmetoden. Hvis vi igjen splitter intervallet i 2^k delintervaller, for $k = 1, 2, \dots, 10$, så får vi

h	$I_{\text{trap}}(h)$	$I - I_{\text{trap}}(h)$
0.500000	0.82386686	1.8×10^{-2}
0.250000	0.83708375	4.4×10^{-3}
0.125000	0.84037503	1.1×10^{-3}
0.062500	0.84119705	2.7×10^{-4}
0.031250	0.84140250	6.8×10^{-5}
0.015625	0.84145386	1.7×10^{-5}
0.007813	0.84146670	4.3×10^{-6}
0.003906	0.84146991	1.1×10^{-6}
0.001953	0.84147072	2.7×10^{-7}
0.000977	0.84147092	6.7×10^{-8}

Vi ser at hver gang steglengden halveres, så blir feilen redusert med omtrent faktor 4. Men vi ser også at selv om vi nå bruker to funksjonsverdier i hvert delintervall for å estimere integralet, så er feilen faktisk dobbelt så stor som for midtpunktsmetoden. Dette kan sees fra feilestimatet (4.10), siden $1/12$ er dobbelt så stor som $1/24$ i (4.7).

4.3.2 $k = 2$: Simpsons metode

For $k = 2$ tilnærmer vi med et kvadratisk polynom på hvert delintervall J_i . Metoden blir i dette tilfellet kalt *Simpsons metode*, og er illustrert i Figur 4.5. I Oppgave 10 blir du bedt å vise at Newtonformen til polynomet som interpolerer f i a , $m = (a + b)/2$, og b er

$$p_2(x) = f(a) + \frac{f(m) - f(a)}{h}(x - a) + \frac{f(a) - 2f(m) + f(b)}{2h^2}(x - a)(x - m),$$

der $h = (b - a)/2$, og at

$$\int_a^b p_2(x) dx = \frac{h}{3} (f(a) + 4f(m) + f(b)).$$

Dette er Simpsons metode med ett delintervall. Legger vi sammen bidragene fra alle delintervallene får vi

$$I_{\text{Simp}}(h) = \frac{h}{3} \left(f(a) + f(b) + 2 \sum_{i=1}^{n-1} f(x_{2i}) + 4 \sum_{i=1}^n f(x_{2i-1}) \right), \quad (4.11)$$

der nå $h = (b - a)/(2n)$ Setter vi inn $k = 2$ i Teorem 2.8, får vi følgende feilestimat for denne metoden.

Teorem 4.12 (Feilestimat for Simpsons metode). *La $f: [a, b] \rightarrow \mathbb{R}$ være en fire ganger kontinuerlig deriverbar funksjon. Feilen i Simpsons metode med steglengde h er da begrenset av*

$$|I - I_{\text{Simp}}(h)| \leq \frac{(b-a)M_3}{6} h^3. \quad (4.12)$$

Simpsons metode er mye brukt. Siden metoden baserer seg på kvadratiske polynomer, så må den være eksakt for slike. Den er faktisk eksakt for alle kubiske polynomer også (se Oppgave 8). Det finnes skarpere feilestimater for Simpsons metode enn dette (se Oppgave 9, som går et stykke utenfor pensum).

Oppgaver for Seksjon 4.3

1. Markér hver av de følgende påstandene som sanne eller gale.

- (a) Trapesmetoden er vanligvis mer nøyaktig enn midtpunktsmetoden.
- (b) Siden hvert evalueringspunkt i trapesmetoden forekommer som endepunkt i to forskjellige delintervaller, så må vi evaluere funksjonen vi integrerer to ganger i hvert punkt.
- (c) Simpsons metode krever at vi bruker et odde antall punkter.
- (d) Simpsons metode er eksakt for polynomer av grad 3 eller lavere.

2. Finn det korrekte alternativet i følgende flervalgsoppgaver.

- (a) Vi bruker trapesmetoden med to delintervaller for å tilnærme integralet

$$\int_0^1 x^2 dx.$$

Hva blir resultatet?

- ☐ 1/2
- ☐ 3/8
- ☐ 5/9
- ☐ 3/5

(b) Hvilke(n) av trapes-, midtpunkts- eller Simpsons metode er mest nøyaktig for lineære polynomer?

- ☐ Trapesmetoden.
- ☐ Simpsons metode.
- ☐ Midtpunktsmetoden.
- ☐ Alle er eksakte.

(c) Vi bruker Simpsons metode for å regne ut tilnærminger til $\int_a^b f(x) dx$. Da blir resultatet alltid eksakt hvis $f(x)$ er

- ☐ en trigonometrisk funksjon.
- ☐ en logaritmisk funksjon.
- ☐ et polynom av grad to.
- ☐ på formen $g(x)/h(x)$ der f og g er polynomer av grad to.

- (d)** ☐ Midtpunktsmetoden er mer nøyaktig enn Simpsons metode.
- ☐ Midtpunktsmetoden og trapesmetoden gir alltid samme feil.
 - ☐ Midtpunktsmetoden gir 0 feil bare hvis $f(x) = c$ for en konstant c .
 - ☐ Midtpunktsmetoden gir 0 feil hvis $f(x)$ er en vilkårlig rett linje i planet.

3. Gjenta Oppgave 3 og 4 i Seksjon 4.2 for trapesmetoden.

4. Hvor mange funksjonsevalueringer trenger du for å beregne integralet

$$\int_0^1 \frac{dx}{1+2x}$$

med feil mindre enn 10^{-10} for

- (a)** trapesmetoden,
- (b)** midtpunktsmetoden,
- (c)** og Simpsons metode?

5. Gjenta Oppgave 3 og 4 i Seksjon 4.2 med Simpsons metode.

6. Vi vil lage en metode for numerisk integrasjon på formen

$$\int_a^b f(x) dx \approx w_1 f(a) + w_2 f((a+b)/2) + w_3 f(b),$$

der w_i skal bestemmes. Vis at, hvis metoden er eksakt for de tre polynomene $f(x) = x^i$ for $i = 0, 1, 2$, så blir metoden lik Simpsons metode.

7. La $m = \frac{a+b}{2}$ være midtpunktet på intervallet $[a, b]$, og se på følgende anvendelse av delvis integrasjon:

$$\begin{aligned} - \int_a^b (x-m) f'(x) dx &= - [(x-m) f(x)]_a^b + \int_a^b f(x) dx \\ &= - \frac{b-a}{2} f(b) - \frac{b-a}{2} f(a) + \int_a^b f(x) dx \\ &= \int_a^b f(x) dx - (b-a) \frac{f(a) + f(b)}{2}. \end{aligned}$$

Dette viser at den lokale feilen i trapesmetoden kan skrives som

$$L = - \int_a^b (x-m) f'(x) dx. \quad (4.13)$$

(a) Anvend delvis integrasjon på (4.13) for å vise at

$$L = \int_a^b \left(C + \frac{1}{2}(x-m)^2 \right) f''(x) dx. \quad (4.14)$$

der C er en konstant (vi skal komme tilbake tilbake hvordan vi skal velge C).

(b) Vis at, med $C = -(b-a)^2/8$, så har vi at $C + \frac{1}{2}(x-m)^2 \leq 0$ for $x \in [a, b]$. Vis også at, med dette valget av C , så følger det fra (4.14) at $|L| \leq \frac{M_2(b-a)^3}{12}$.

(c) Bruk (b) til å vise at den globale feilen i trapesmetoden er $\leq (b-a) \frac{h^2}{12} M_2$.

For å forenkle notasjonen i de neste oppgavene definerer vi $m = a + h$, og $b = a + 2h$. Vi definerer også

$$q(x) = (x-a)(x-m)(x-b),$$

og $p_2(x)$ som det kvadratiske polynomet som interpolerer f i a, m , og b .

8. I denne oppgaven skal vi se på forskjellige måter å vise at Simpsons metode er eksakt for alle polynomer av grad 3.

(a) Verifiser at Simpsons metode er eksakt for $f(x) = x^i$ for $i = 0, 1, 2, 3$.

(b) Bruk (a) til å vise at Simpsons metode er eksakt for ethvert kubisk polynom.

(c) Kunne du komme frem til samme konklusjon som i (b) ved bare å betrakte feilestimatet (4.12)?

(d) Forklar at q er anti-symmetrisk om m , og at, hvis f er et polynom av grad 3, så er $f - p_2$ et multiplum av q . Hvorfor følger eksakthet av Simpsons metode for polynomer av grad 3 fra dette?

9. I denne oppgaven skal vi bevise et skarpere feilestimat enn (4.12) for Simpsons metode. Vi lar

$$E = \int_a^b f(x) dx - \int_a^b p_2(x) dx$$

være feilen i Simpsons metode med ett intervall.

(a) Vis at $\int_a^b xq(x) dx = -\frac{1}{120}(b-a)^5$, og at $(xq(x))^{(4)} = 4!$.

(b) Se på den kontinuerlige funksjonen

$$\varphi(x) = f(x) - p_2(x) - kq(x) - \frac{E}{\int_a^b xq(x) dx} xq(x),$$

Vis at

$$\varphi^{(4)}(x) = f^{(4)}(x) + 120 \frac{4!E}{(b-a)^5}, \quad (4.15)$$

og at $\int_a^b \varphi(x) dx = 0$. Til slutt, forklar at $\varphi(a) = \varphi(m) = \varphi(b) = 0$.

(c) Vi at det eksisterer en k slik at $\int_a^m \varphi(x) dx = 0$.

For siste del av denne oppgaven lar vi k være definert som i (c).

(d) Vis at φ har minst fem forskjellige nullpunkter i $[a, b]$.

(e) Rolles theorem sier at, for en kontinuerlig deriverbar funksjon definert på $[a, b]$ slik at $\varphi(a) = \varphi(b) = 0$, så eksisterer det et punkt $c \in (a, b)$ slik at $\varphi'(c) = 0$. Bruk Rolles teorem til å vise at det finnes en $\xi \in (a, b)$ slik at $\varphi^{(4)}(\xi) = 0$.

(f) Forklar at, med ξ som i (e), så har vi at $E = -f^{(4)}(\xi)(b-a)^5/2880$. Det følger at feilen for ett intervall er $\leq M_4(b-a)^5/2880$. Siden vi har bidrag fra n slike, så blir den en øvre grense for feilen

$$nM_4(2h)^5/2880 = (b-a)M_4(2h)^4/2880 = M_4(b-a)h^4/180,$$

Dette er betydelig bedre enn det globale feilestimatet (4.12).

- 10.** (a) Vis at Newtonformen til polynomet som interpolerer f i a , $m = (a+b)/2$, og b er

$$p_2(x) = f(a) + \frac{f(m) - f(a)}{h}(x - a) + \frac{f(a) - 2f(m) + f(b)}{2h^2}(x - a)(x - m),$$

der $h = (b - a)/2$.

- (b) Vis at $\int_a^b p_2(x) dx = \frac{h}{3} (f(a) + 4f(m) + f(b))$.

Hint: Regn ut $\int_a^b (x - a)(x - m) dx$ først, se Oppgave 3 i Seksjon 2.3.

- 11.** På intervallet $[x_{i-1}, x_i]$ bruker trapesmetoden kun funksjonsverdiene $f(x_{i-1})$ og $f(x_i)$. På det neste intervallet brukes verdiene $f(x_i)$ og $f(x_{i+1})$. Alle funksjonsverdiene, bortsett fra første og siste, forekommer derfor to ganger i summen til høyre i (4.9). Funksjonsevalueringer kan være kostbare — de kan for eksempel stamme fra sensorer som er dyre i bruk. Skriv om (4.9) på en måte slik at vi unngår gjentagende funksjonsevalueringer.

KAPITTEL 5

Kondisjonstall

Anta vi skal løse systemet $A\mathbf{x} = \mathbf{b}$ på en datamaskin. På grunn av målingsfeil eller avrundingsfeil vil maskinen i stedet få servert et problem på formen

$$(A + E)\mathbf{x} = \mathbf{b} + \mathbf{e},$$

der E er en matrise med små elementer, som representerer et avvik (en feil) fra den faktiske koeffisientmatrisen A . Det samme gjelder for $\mathbf{e}$ og høyresiden $\mathbf{b}$. Når vi snakker om feil mellom to størrelser er det naturlig å bruke begrepet relativ feil.

Definisjon 5.1. Anta a og b er tall. Vi kaller $\left| \frac{b-a}{a} \right|$ for den relative feilen mellom a og b . Hvis $\mathbf{a}$ og $\mathbf{b}$ mer generelt er vektorer definerer vi den relative feilen mellom disse som $\frac{|\mathbf{b}-\mathbf{a}|}{|\mathbf{a}|}$.

Relativ feil sier mer om størrelsesforholdet mellom a og b enn hva den absolute feilen $|b - a|$ gjør, siden den ikke endrer seg hvis vi skalerer tallene: Siden $\left| \frac{cb-ca}{ca} \right| = \left| \frac{b-a}{a} \right|$ for enhver $c \neq 0$, så er den relative feilen mellom a og b identisk med den relative feilen mellom ca og cb . Det samme er ikke tilfelle for den absolute feilen mellom a og b , siden $|cb - ca| \neq |b - a|$.

Relativ feil sier også noe om hvor mange desimaler tallene a og b har felles: Hvis den relative feilen er omtrent 10^{-k} , så betyr det at omtrent de k første desimale sifrene i a og b stemmer overens.

5.1 Definisjon av kondisjonstallet

Vi skal nå se nærmere på den relative feilen mellom to vektorer $\mathbf{a}$ og $\mathbf{b}$. La oss for enkelthets skyld anta at vi ikke har avrundingsfeil i koeffisientmatrisen A , slik at $E = 0$. La $\mathbf{x}$ være løsningen av $A\mathbf{x} = \mathbf{b}$, og la $\mathbf{y}$ være løsningen av det *perturberte* systemet $A\mathbf{y} = \mathbf{b} + \mathbf{e}$. Vi har nå to forskjellige relative feil:

1. Den relative feilen i høyresiden i systemet: $|(\mathbf{b} + \mathbf{e}) - \mathbf{b}|/|\mathbf{b}| = |\mathbf{e}|/|\mathbf{b}|$.

2. Den relative feilen mellom løsningene $\mathbf{x}$ og $\mathbf{y}$: $|\mathbf{y} - \mathbf{x}|/|\mathbf{x}|$.

Det vi ikke vil er at en liten relativ feil i høyresiden skal forårsake en stor relativ feil i løsningen. *Kondisjonstallet til A* , $K(A)$, er definert som det minste tallet som oppfyller

$$\frac{|\mathbf{y} - \mathbf{x}|}{|\mathbf{x}|} \leq K(A) \frac{|\mathbf{e}|}{|\mathbf{b}|} \quad (5.1)$$

for alle $\mathbf{b}$ og $\mathbf{e}$. Med andre ord, $K(A)$ sier hvor mye den relative feilen i høyresiden kan propagere til løsningene. Som et eksempel, hvis $K(A) \leq 10^k$, og hvis $|\mathbf{e}|/|\mathbf{b}| \leq 10^{-r}$ (det vil si at høyresiden har minst r riktige sifre), så sier (5.1) at

$$\frac{|\mathbf{y} - \mathbf{x}|}{|\mathbf{x}|} \leq 10^k 10^{-r} = 10^{k-r},$$

det vil si at vi kan ha mistet k riktige sifre når vi løser systemet:

Observasjon 5.2. Hvis kondisjonstallet er $\approx 10^k$, så risikerer vi å miste k riktige signifikante sifre når vi løser det perturberte systemet.

Vi vil derfor helst unngå matriser med høye kondisjonstall.

Proposisjon 5.3. Anta at A er symmetrisk og inverterbar, og la

$$|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_n| > 0$$

være egenverdiene til A listet opp etter avtagende absoluttverdi. Da har vi at $K(A) = |\lambda_1/\lambda_n|$, det vil si, $|\lambda_1/\lambda_n|$ er det minste tallet som tilfredsstiller (5.1).

I Oppgave 3 blir du bedt om å vise at også

$$\frac{1}{K(A)} \frac{|\mathbf{e}|}{|\mathbf{b}|} \leq \frac{|\mathbf{y} - \mathbf{x}|}{|\mathbf{x}|}, \quad (5.2)$$

og at vi kan ha likhet for et konkret valg av $\mathbf{b}$ og $\mathbf{e}$. Med andre ord, mens (5.1) sier at den relative feilen i løsningen av det perturberte systemet kan propagere med maksimum faktor $K(A) = |\lambda_1/\lambda_n|$, så sier (5.2) at den samme relative feilen nødvendigvis må propagere med minimum faktor $1/K(A) = |\lambda_n/\lambda_1|$.

Bevis for Proposisjon 5.3. Vi vet at symmetriske matriser er ortogonalt diagonaliserbare, slik at vi kan skrive $A = PDP^T$, der P er en ortogonal matrise. Dette skal vi først bruke for å vise at $|\lambda_n||\mathbf{x}| \leq |A\mathbf{x}| \leq |\lambda_1||\mathbf{x}|$. Vi får at

$$|A\mathbf{x}|^2 = |PDP^T\mathbf{x}|^2 = \mathbf{x}^T PDP^T PDP^T \mathbf{x} = (P^T \mathbf{x})^T D^2 P^T \mathbf{x}.$$

Setter vi $\mathbf{u} = P^T \mathbf{x}$ får vi

$$\begin{aligned}\mathbf{u}^T D^2 \mathbf{u} &= \sum_{i=1}^n \lambda_i^2 u_i^2 \leq \lambda_1^2 \sum_{i=1}^n u_i^2 = \lambda_1^2 |\mathbf{u}|^2 = \lambda_1^2 |\mathbf{x}|^2 \\ \mathbf{u}^T D^2 \mathbf{u} &= \sum_{i=1}^n \lambda_i^2 u_i^2 \geq \lambda_n^2 \sum_{i=1}^n u_i^2 = \lambda_n^2 |\mathbf{u}|^2 = \lambda_n^2 |\mathbf{x}|^2.\end{aligned}$$

Ved å ta kvadratrøtter følger at $|\lambda_n| |\mathbf{x}| \leq |A\mathbf{x}| \leq |\lambda_1| |\mathbf{x}|$.

La oss så vise at (5.1) holder. Vi får først at $A(\mathbf{y}-\mathbf{x}) = \mathbf{e}$ og $\mathbf{y}-\mathbf{x} = A^{-1}\mathbf{e}$. Deretter får vi at

$$\begin{aligned}|\mathbf{y}-\mathbf{x}| &= |A^{-1}\mathbf{e}| \leq \frac{1}{|\lambda_n|} |\mathbf{e}| \\ |\mathbf{b}| &= |A\mathbf{x}| \leq |\lambda_1| |\mathbf{x}|.\end{aligned}$$

Her brukte vi at A^{-1} også er en symmetrisk matrise, med egenverdier $1/\lambda_n, \dots, 1/\lambda_1$ (disse ble her listet opp etter avtagende absoluttverdi), se Oppgave 1. Det følger nå at

$$\frac{|\mathbf{y}-\mathbf{x}|}{|\mathbf{x}|} \leq \frac{\frac{1}{|\lambda_n|} |\mathbf{e}|}{\frac{1}{|\lambda_1|} |\mathbf{b}|} = \left| \frac{\lambda_1}{\lambda_n} \right| \frac{|\mathbf{e}|}{|\mathbf{b}|}.$$

Det følger at $K(A) \leq \left| \frac{\lambda_1}{\lambda_n} \right|$. I Oppgave 2 viser vi at vi kan få likhet over for et bestemt valg av $\mathbf{b}$ og $\mathbf{e}$. For dette valget har vi da at

$$\frac{|\mathbf{y}-\mathbf{x}|}{|\mathbf{x}|} = \left| \frac{\lambda_1}{\lambda_n} \right| \frac{|\mathbf{e}|}{|\mathbf{b}|} \leq K(A) \frac{|\mathbf{e}|}{|\mathbf{b}|},$$

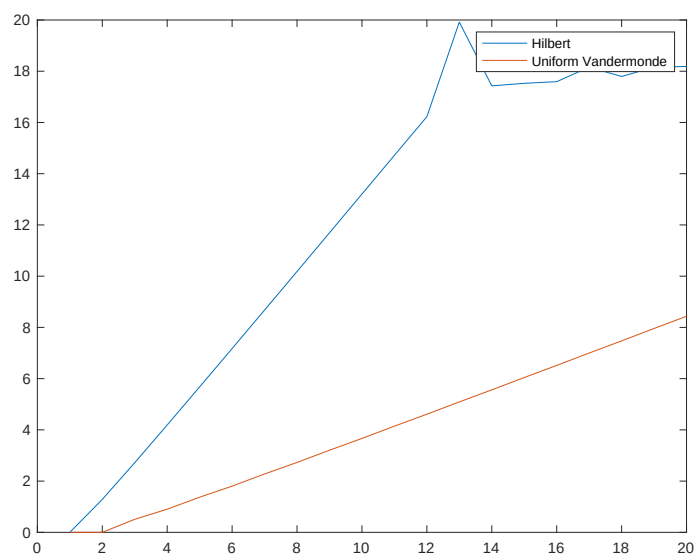
slik at også $\left| \frac{\lambda_1}{\lambda_n} \right| \leq K(A)$. Det følger at $K(A) = |\lambda_1/\lambda_n|$. ■

Kondisjonstallet kan mer generelt defineres for alle matriser, ikke bare for de symmetriske. Vi skal dog ikke gå inn på den generelle definisjonen. Ved å skrive `cond(A)` i Matlab får du ut kondisjonstallet til matrisen.

Noen matriser har veldig høye kondisjonstall. Hilbertmatrisen H har elementer gitt ved $h_{ij} = \frac{1}{i+j-1}$. Koden under plotter kondisjonstall for slike matriser, samt for vandermondematriser med generatorer uniformt fordelt på $[-1, 1]$.

```
n = 1:20;
ch = zeros(1,20);
cv = zeros(1,20);
for k = n
    ch(k) = cond(hilb(k));
    gens = linspace(-1,1,k);
    cv(k) = cond(gens.^(0:(k-1)'));
end
semilogy(n,ch,n,cv)
legend('Hilbert','Uniform Vandermonde')
```

Program 5.1. Sammenligning av kondisjonstall



Figur 5.1. Kondisjonstall for hilbert- og vandermondematrixene

Plottet er logaritmisk (se Oppgave 2 i Kapittel 1 med tanke på kommandoen `semilog`). Langs x -aksen har vi n , langs y -aksen har vi antall riktige siffer vi mister. Vi ser at hilbertmatrisen mister 2 sifres presisjon hver gang vi øker n , mens vandermondematrixen bare mister ett siffer. Fra plottet virker det som vi kan få så store kondisjonstall vi vil, ved bare å velge matrixene store nok. Det neste eksemplet at vi kan få store kondisjonstall vi vil ved kun å se på 2-matriser også.

Eksempel 5.4. La oss se på matrixen $A = \begin{pmatrix} 1 & 1-\epsilon \\ 1-\epsilon & 1 \end{pmatrix}$. Først finner vi egenverdiene til A :

$$\det(\lambda I - A) = (\lambda - 1)^2 - (1 - \epsilon)^2 = \lambda^2 - 2\lambda + 1 - (1 - \epsilon)^2.$$

Denne har røtter

$$\frac{2 \pm \sqrt{4 - 4(1 - (1 - \epsilon)^2)}}{2} = 1 \pm (1 - \epsilon).$$

Største egenverdi er $\lambda_1 = 2 - \epsilon$, minste er $\lambda_2 = \epsilon$, slik at kondisjonstallet er $K(A) = (2 - \epsilon)/\epsilon = 2/\epsilon - 1$. Med andre ord, her kan vi få så stort kondisjonstall vi vil ved å velge ϵ liten nok.

La oss i den videre regningen sette $\epsilon = 1/4$, slik at $K(A) = 8 - 1 = 7$. Egenverdiene blir da $\lambda_1 = 7/4$, $\lambda_2 = 1/4$.

- En egenvektor for $\lambda_1 = 7/4$.

$$\frac{7}{4}I - A = \begin{pmatrix} 3/4 & -3/4 \\ -3/4 & 3/4 \end{pmatrix} \sim \begin{pmatrix} 1 & -1 \\ 0 & 0 \end{pmatrix},$$

slik at $\mathbf{v}_1 = (1, 1)$.

- En egenvektor for $\lambda_1 = 1/4$:

$$\frac{1}{4}I - A = \begin{pmatrix} -3/4 & -3/4 \\ -3/4 & -3/4 \end{pmatrix} \sim \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix},$$

slik at $\mathbf{v}_2 = (1, -1)$.

Sett nå spesielt $\mathbf{b} = (7/4, 7/4)$, $\mathbf{e} = (1, -1)$. Vi får

$$\frac{|\mathbf{e}|}{|\mathbf{b}|} = \frac{\sqrt{2}}{\frac{7}{4}\sqrt{2}} = \frac{4}{7}.$$

Høyresiden i (5.1) blir nå $28/7 = 4$.

Løsningen $\mathbf{x}$ for det uperturberte systemet, samt løsningen $\mathbf{y}$ for det perturberte systemet kan vi finne simultant ved å radredusere

$$\begin{aligned} (A \quad \mathbf{b} \quad \mathbf{b} + \mathbf{e}) &= \begin{pmatrix} 1 & 3/4 & 7/4 & 11/4 \\ 3/4 & 1 & 7/4 & 3/4 \end{pmatrix} \sim \begin{pmatrix} 1 & 3/4 & 7/4 & 11/4 \\ 0 & 7/16 & 7/16 & -21/16 \end{pmatrix} \\ &\sim \begin{pmatrix} 1 & 0 & 1 & 5 \\ 0 & 1 & 1 & -3 \end{pmatrix}, \end{aligned}$$

slik at $\mathbf{x} = (1, 1)$, og $\mathbf{y} = (5, -3)$. Den relative feilen i løsningene blir dermed

$$\frac{|\mathbf{y} - \mathbf{x}|}{|\mathbf{x}|} = \frac{4\sqrt{2}}{\sqrt{2}} = 4,$$

som er venstresiden i (5.1). Dette valget av høyreside $\mathbf{b}$ og perturbasjon $\mathbf{e}$ ga altså likhet i (5.1). Det følger at dette valget gir maksimal propagering av relativ feil til det perturberte systemet. I oppgave 2 vil du se at det vi gjorde her holder mer generelt også.

Oppgaver for Seksjon 5.1

1. Anta A er en inverterbar, symmetrisk $n \times n$ -matrise, med egenverdier $\lambda_1, \lambda_2, \dots, \lambda_n$. Vis at A^{-1} også er en symmetrisk matrise, med egenverdier $1/\lambda_1, \dots, 1/\lambda_n$.

2. Vis at, hvis $\mathbf{e}$ er en egenvektor for den minste egenverdien til A , og $\mathbf{x}$ er en egenvektor for den største egenverdien til A , så har vi likhet i (5.1). Med andre ord, vi kan ikke gjøre ulikheten (5.1) strengere, slik at den fremdeles holder universelt for alle høyresider $\mathbf{b}$ og perturbasjoner $\mathbf{e}$.

3. Anta A er symmetrisk og inverterbar. Vis at

$$\frac{1}{K(A)} \frac{|\mathbf{e}|}{|\mathbf{b}|} \leq \frac{|\mathbf{y} - \mathbf{x}|}{|\mathbf{x}|}$$

for alle valg av $\mathbf{b}$ og $\mathbf{e}$. For hvilket valg av $\mathbf{e}$ og $\mathbf{x}$ får du likhet her?

4. En $n \times n$ -matrise kalles ortogonal hvis $U^{-1} = U^T$. Forklar at kondisjonstallet til U er 1. Dette gjelder uavhengig av om U er symmetrisk eller ikke. Hva sier dette oss om slike matriser med tanke på løse et perturbert system?

APPENDIX

Fasit

Seksjon 1.1

Seksjon 1.2

Seksjon 2.1

Exercise 1(a). Sant

Exercise 1(b). Usant

Exercise 1(c). Usant

Seksjon 2.2

Exercise 1. 16

Exercise 2(a).

$$p_3(x) = -\frac{(x-1)(x-3)(x-4)}{12} - \frac{x(x-1)(x-4)}{3} + \frac{x(x-1)(x-3)}{12}.$$

Exercise 2(b).

$$p_3(x) = 1 - x + \frac{2}{3}x(x-1) - \frac{1}{3}x(x-1)(x-3).$$

Exercise 3(a). Newtonformen er

$$p_2(x) = 2 - x.$$

Exercise 4(a). Lineær interpolant p_1 :

$$p_1(x) = y_1 + (y_2 - y_1)(x - 1).$$

Exercise 4(b). Kubisk interpolant:

$$p_3(x) = y_0 + (y_1 - y_0)x + \frac{y_2 - 2y_1 + y_0}{2}x(x-1) + \frac{y_3 - 3y_2 + 3y_1 - y_0}{6}x(x-1)(x-2).$$

Seksjon 2.3

Seksjon 3.1

Exercise 3(a). Et feilestimat er nå

$$\frac{h}{2}|f''(a)| + \frac{h^2}{6} \max_{x \in [a, a+h]} |f'''(x)|.$$

Exercise 3(b). Det er ikke mulig å få et estimat på trunkeringsfeilen ved å bruke en slik Taylorutvikling.

Exercise 3(c). Det lineære Taylorpolynomet er best siden det er den korteste mulige Taylorutviklingen som kan gi et estimat for $f'(a)$ (som vi viste i (b)), og også det som gir det enkleste uttrykket for trunkeringsfeilen, på den måten at den ikke er avhengig av deriverte av høyere orden enn to (som vi viste i (a)).

Seksjon 3.2

Exercise 5(a). $f'(a) \approx p'_2(a) = \frac{3f(a)-4f(a-h)+f(a+2h)}{6h}.$

Exercise 5(b). $f'(a) \approx p'_2(a) = \frac{-3f(a)+4f(a-h)-f(a+2h)}{2h}.$

Exercise 6(a). $c_1 = -1/(2h), c_2 = 1/(2h).$

Exercise 6(c). $c_1 = -1/h^2, c_2 = 2/h^2, c_3 = -1/h^2.$

Exercise 7(b). Med 6 sifre:

$(f(a+h) - f(a))/h = 0.975,$ relativ feil: 0.025.

$(f(a) - f(a-h))/h = 1.025,$ relativ feil: 0.025.

$(f(a+h) - f(a-h))/(2h) = 1,$ relativ feil: $8.88178 \times 10^{-16}.$

Seksjon 4.1

Seksjon 4.2

Exercise 1(a). Sant

Exercise 1(b). Galt

Exercise 1(c). Sant

Exercise 1(d). True

Exercise 2. 5/16.

Exercise 3. Tilnærming: 0.530624 (med 6 sifre).

Exercise 4(a). Tilnærmingen med 10 delintervaller: 1.71757 (med 6 sifre)

Exercise 4(b). $h \leq 2.97 \times 10^{-5}.$

Exercise 5. Tilnærming med 10 delintervaller: 5.36648 (med 6 sifre). $h \leq 4.89 \times 10^{-5}.$

Seksjon 4.3

Exercise 2(a). 3/8

Exercise 3. Oppgave 3: Tilnærming: 0.519725 (med 6 sifre).

Oppgave 4: Tilnærming med 10 delintervaller: 1.71971 (med 6 sifre). $h \leq 2.1 \times 10^{-5}$.

Exercise 4(a). 81 650 evalueringer.

Exercise 4(b). 57 736 evalueringer.

Exercise 4(c). 383 evalueringer.

Exercise 5. Oppgave 3: Tilnærming: 0.527217 (med 6 sifre).

Oppgave 4: Tilnærming med 10 delintervaller: 1.718282782 (med 10 sifre).
 $h \leq 1.8 \times 10^{-2}$

Seksjon 5.1

APPENDIX

Løsningsforslag

Seksjon 1.1

Exercise 1. Koden kan se slik ut:

```
x = linspace(0,1,100)';  
n = 3;  
x_powers = x.^(0:n);  
Y = x_powers*[[0 1 0 0]', [0 0 1 0]', [0 0 0 1]'];  
plot(x,Y)  
legend('p_1','p_2','p_3')
```

Exercise 2(a). En linje i et slikt plott betyr at $y = a \log x + b$, der a og b er konstanter vi kan lese ut fra linja (a er stigningstallet, b er skjæringen med y -aksen). $y = a \log x + b$ gir at $x = e^{(y-b)/a} = e^{-b/a} e^{y/a}$, slik at x er eksponensiell i y med konstant $K = e^{-b/a}$ og eksponent $1/a$.

Exercise 2(b). En linje i et slikt plott betyr at $\log y = ax + b$. Dette gir at $y = e^b e^{ax}$, slik at y er eksponensiell i x med konstant $K = e^b$ og eksponent a .

Seksjon 1.2

Exercise 1(a). La $\mathbf{v}_1$ være en egenvektor for A , med λ som tilhørende egenverdi. Da er

$$C \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} A\mathbf{v}_1 \\ \mathbf{0} \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{0} \end{pmatrix},$$

slik at $\begin{pmatrix} \mathbf{v}_1 \\ \mathbf{0} \end{pmatrix}$ er en egenvektor for C med samme egenverdi. Tilfellet der λ er en egenverdi for B følger tilsvarende.

Exercise 1(b). Anta $C \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{pmatrix}$. Det følger at $A\mathbf{v}_1 = \lambda\mathbf{v}_1$ og $B\mathbf{v}_2 = \lambda\mathbf{v}_2$. Siden en egenvektor er $\neq \mathbf{0}$ så må en av $\mathbf{v}_1$ og $\mathbf{v}_2$ være $\neq \mathbf{0}$. Det følger at λ må være en egenverdi for enten A eller B (eller begge).

Seksjon 2.1

Exercise 2. Hvis vi antar at datamaskinen regner ut x^n ved å gange inn x $n - 1$ ganger, så vil det første kodesegmentet bruke $n(1 + \dots + n - 2) = n(n - 1)(n - 2)/2$ operasjoner, mens det andre kodesegmentet bruker $n^2 - n$ operasjoner, og derfor er å foretrekke.

Det er forøvrig ikke sikkert at datamaskinen regner ut x^n ved hjelp av $n - 1$ multiplikasjoner. Sannsynligvis så bruker maskinen en mer effektiv algoritme. Legg spesielt merke til at $x^n = e^{n \ln x}$, som maskinen kan utnytte til å regne ut potenser såfremt den har effektive algoritmer for å regne ut logaritmer og eksponensialfunksjonen.

Exercise 3. I første del av Gausseliminering nuller vi ut elementene under diagonalen, søyle for søyle. La oss se på hvor mange operasjoner dette krever for søyle nr. k . Under diagonalelement k er det $n - k$ rader. For hver av disse ganger vi rad k med et tall (dette svarer til $n - k$ multiplikasjoner, siden raden starter med $k - 1$ nuller), og legger dette til raden (dette svarer til $n - k$ addisjoner). Til sammen får vi $(n - k)((n - k) + (n - k)) = 2(n - k)^2$ operasjoner her. Summerer vi over alle k får vi $\approx 2 \sum_{k=0}^n (n - k)^2 = 2 \sum_{k=0}^n k^2 = n(n + 1)(2n + 1)/6 \approx n^3/3$.

I andre del av Gausseliminering nuller vi ut elementer over diagonalen, søyle for søyle. For søyle nr. k har vi $k - 1$ rader over. For hver av disse ganger vi rad k med et tall ($n - k$ multiplikasjoner, siden raden starter med $k - 1$ nuller), og legger dette til raden ($(n - k)$ addisjoner). Til sammen gir dette $(n - k)((n - k) + (n - k)) = 2(n - k)^2$ operasjoner her. Summerer vi over k som over får vi igjen $\approx n^3/3$ regneoperasjoner. Totalt for de to delene i Gausseliminering får vi da $\approx n^3/3 + n^3/3 = \frac{2}{3}n^3$ regneoperasjoner.

Exercise 4(a). Siden $p_1(x_0) = p_2(x_0) = f(x_0)$, $p_1(x_1) = p_2(x_1) = f(x_1)$, $p_1(x_2) = p_2(x_2) = f(x_2)$, så får vi at

$$p(x_0) = p_2(x_0) - p_1(x_0) = f(x_0) - f(x_0) = 0$$

$$p(x_1) = p_2(x_1) - p_1(x_1) = f(x_1) - f(x_1) = 0$$

$$p(x_2) = p_2(x_2) - p_1(x_2) = f(x_2) - f(x_2) = 0.$$

Derfor er verdien i interpolasjonspunktene 0.

Exercise 4(b). Andregradspolynomet p har tre forskjellige nuller, men et andregradspolynom forskjellig fra 0 kan ikke ha mer enn to nuller. Vi må derfor ha at $p = 0$, slik at $p_1 = p_2$, slik at det interpolerende polynomet er unikt.

Exercise 4(c). Hvis vi definerer $p = p_1 - p_2$ som over, så ser vi som før at $p(x_0) = p(x_1) = \dots = p(x_n) = 0$, slik at p er et polynom av grad n med $n + 1$ forskjellige nuller. Fra algebraens fundamentalteorem (d.v.s. at et polynom av grad n ikke kan ha mer enn n røtter) følger det at $p = 0$, slik at $p_1 = p_2$, slik at det interpolerende polynomet er unikt også i dette tilfellet.

Exercise 5(a). Hvis vi kaller beregningene av høyresidene for s , og starter med siste likning, så blir stegene

$$\begin{aligned}s &= c_n \\ s &= c_{n-1} + xs \\ \vdots & \quad \quad \quad \vdots \\ s &= c_2 + xs \\ s &= c_1 + xs \\ s &= c_0 + xs.\end{aligned}$$

Etter dette får vi at $s = p_n(x)$. Koden kan se slik ut

```
s = c(n+1);
for k = n:-1:1
    s = c(k) + x*s;
end
```

Exercise 5(b). I hver iterasjon av løkka gjøres en addisjon og en multiplikasjon. Det totale antallet er derfor $2n$ operasjoner, som er en forbedring fra $3n - 1$. Horner's regel er faktisk en optimal metode når det kommer til å regne ut verdier av polynomet. Den er også optimal når det kommer til minnebruk — man trenger bare variabelen s i tillegg til x og c . Skal du integrere den i koden i Program 2.1 så skal du bytte ut `x_plot_powers*c` med det du får fra koden over.

Seksjon 2.2

Exercise 2(a). Setter vi $x = 0$ får vi at $1 = c_0(-1)(-3)(-4)$, slik at $c_0 = -1/12$.

Setter vi $x = 1$ får vi at $0 = c_1(1)(1-3)(1-4)$, slik at $c_1 = 0$.

Setter vi $x = 3$ får vi at $2 = c_2(3)(3-1)(3-4)$, slik at $c_2 = -1/3$.

Setter vi $x = 4$ får vi at $1 = c_3(4)(4-1)(4-3)$, slik at $c_3 = 1/12$.

Vi får derfor

$$p_3(x) = -\frac{(x-1)(x-3)(x-4)}{12} - \frac{x(x-1)(x-4)}{3} + \frac{x(x-1)(x-3)}{12}.$$

Exercise 2(b). Setter vi $p_3(x) = c_0 + c_1x + c_2x(x-1) + c_3x(x-1)(x-3)$, så får vi likningene

$$\begin{aligned}1 &= c_0 \\ 0 &= c_0 + c_1 \\ 2 &= c_0 + 3c_1 + 6c_2 \\ 1 &= c_0 + 4c_1 + 12c_2 + 12c_3.\end{aligned}$$

De første to likningene gir at $c_0 = 1$ og $c_1 = -1$. Setter vi dette inn i den tredje likningen får vi at $c_2 = (2 - 1 + 3)/6 = 2/3$. Setter vi dette inn i den fjerde likningen får vi at $c_3 = (1 - 1 + 4 - 8)/12 = -1/3$. Dette gir

$$p_3(x) = 1 - x + \frac{2}{3}x(x-1) - \frac{1}{3}x(x-1)(x-3).$$

Exercise 3(a). Newtonformen til det interpolerende andregangspolynomiet er

$$p_2(x) = f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1).$$

Vi har at

$$\begin{aligned} f[x_0, x_1] &= \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{1 - 2}{1 - 0} = -1 \\ f[x_1, x_2] &= \frac{f(x_2) - f(x_1)}{x_2 - x_1} = \frac{0 - 1}{2 - 1} = -1 \\ f[x_0, x_1, x_2] &= \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = \frac{-1 - (-1)}{2 - 0} = 0. \end{aligned}$$

Setter vi dette inn i $p_2(x)$ får vi at $p_2(x) = 2 - (x - x_0) = 2 - x$.

Exercise 3(b). Både f og p_n er interpolerende polynomer av grad n , og da vet vi fra 4 at de må være like. I (a) var funksjonen allerede et andregradspolynom, slik at den må være lik dens interpolant også. Derfor fikk vi at p_n var lik funksjonen selv.

Exercise 4(a). Lineær interpolant p_1 :

$$p_1(x) = y_1 + (y_2 - y_1)(x - 1).$$

Dette gir $p_1(3/2) = y_1 + \frac{1}{2}(y_2 - y_1) = \frac{1}{2}(y_1 + y_2)$.

Exercise 4(b). Newtonformen for polynomiet som interpolerer i de 4 punktene kan vi finne ved å løse

$$\begin{aligned} y_0 &= c_0 \\ y_1 &= c_0 + c_1 \\ y_2 &= c_0 + 2c_1 + 2c_2 \\ y_3 &= c_0 + 3c_1 + 6c_2 + 6c_3. \end{aligned}$$

Dette gir først $c_0 = y_0$, så $c_1 = y_1 - y_0$, så

$$\begin{aligned} c_2 &= \frac{y_2 - y_0 - 2(y_1 - y_0)}{2} = \frac{y_2 - 2y_1 + y_0}{2} \\ c_3 &= \frac{y_3 - y_0 - 3(y_1 - y_0) - 6(y_2 - 2y_1 + y_0)/2}{6} = \frac{y_3 - 3y_2 + 3y_1 - y_0}{6}, \end{aligned}$$

slik at det interpolerende kubiske polynomiet blir

$$p_3(x) = y_0 + (y_1 - y_0)x + \frac{y_2 - 2y_1 + y_0}{2}x(x - 1) + \frac{y_3 - 3y_2 + 3y_1 - y_0}{6}x(x - 1)(x - 2).$$

Dette gir til slutt

$$\begin{aligned} p_3(3/2) &= y_0 + (y_1 - y_0)\frac{3}{2} + \frac{y_2 - 2y_1 + y_0}{2}\frac{3}{4} - \frac{y_3 - 3y_2 + 3y_1 - y_0}{6}\frac{3}{8} \\ &= \frac{16y_0 + 24y_1 - 24y_0 + 6y_2 - 12y_1 + 6y_0 - y_3 + 3y_2 - 3y_1 + y_0}{16} \\ &= \frac{-y_0 + 9y_1 + 9y_2 - y_3}{16}. \end{aligned}$$

Exercise 5. I den første løkka regner vi ut koeffisientmatrisen A , som vi trenger for å finne newtonformen. I den neste løkka gjør vi fremoversubstitusjon for å finne newtonformen. Hver av disse to løkkene trenger omtrent n^2 operasjoner, slik at vi totalt får $2n^2$ operasjoner.

Exercise 6(a). La oss prøve med et kubisk polynom på formen

$$p_3(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + c_3(x - x_0)(x - x_1)^2.$$

Vi har at $p'_3(x) = c_1 + c_2(2x - x_0 - x_1) + c_2(x - x_1)(3x - 2x_0 - x_1)$, og interpolasjonsbetingelsene gir

$$\begin{aligned} f(x_0) &= c_0 \\ f(x_1) &= c_0 + c_1(x_1 - x_0) \\ f'(x_1) &= \quad + c_1 \quad + c_2(x_1 - x_0) \\ f(x_2) &= c_0 + c_1(x_2 - x_0) + c_2(x_2 - x_0)(x_2 - x_1) + c_3(x_2 - x_0)(x_2 - x_1)^2 \end{aligned}$$

Dette gir matriselikningen

$$\begin{pmatrix} f(x_0) \\ f(x_1) \\ f'(x_1) \\ f(x_2) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & x_1 - x_0 & 0 & 0 \\ 0 & 1 & x_1 - x_0 & 0 \\ 1 & x_2 - x_0 & (x_2 - x_0)(x_2 - x_1) & (x_2 - x_0)(x_2 - x_1)^2 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix}$$

Igjen er koeffisientmatrisen her nedretriangulær med ikkenuller på diagonalen, slik at den er inverterbar. Det følger at koeffisientene er unike, slik at det finnes et unikt polynom.

Exercise 6(b). Ja, det er mulig å generalisere dette. Anta at $f(x_i), f'(x_i), \dots, f^{(k-1)}(x_i)$ er kjente. Vi har bidrag fra $r_n(x)(x - x_i), r_n(x)(x - x_i)^2, \dots, r_n(x)(x - x_i)^k, r_n(x)(x - x_i)^k s_n(x)$, der r_n representerer bidraget fra $x_0, \dots, x_{i-1}$, s_n bidraget fra $x_{i+1}, \dots, x_n$.

Verdien til den deriverte av $r_n(x)(x - x_i)$ i x_i er $r_n(x_i)$. Den deriverte til $r_n(x)(x - x_i)^2$ o.s.v. i x_i er 0.

Verdien til derivert nr. $k - 1$ til $r_n(x)(x - x_i)^{k-1}$ er $(k - 1)!r_n(x)$, som er $\neq 0$.

Igjen får vi et triangulært system med ikkenummer på diagonalen.

Seksjon 2.3

Exercise 1. Vi har at $f^{(n+1)}(x) = e^x$, slik at $M_{n+1} = e$. Teorem 2.17 gir da at, for alle $x \in [0, 1]$, så er

$$|f(x) - p_n(x)| \leq \frac{1 \cdot (1 - 0)^{(n+1)}}{(n+1)!} = \frac{1}{(n+1)!}$$

Vi krever da at $\frac{1}{(n+1)!} \leq 10^{-3}$, det vil si at $(n+1)! \geq 1000$. Dette kan vi få til ved å velge $n \geq 6$.

Exercise 3(b). Vi får

$$\begin{aligned}\int_{x_0}^{x_2} (x-x_0)(x-x_1) dx &= \left[\frac{1}{2} (x-x_0)^2 (x-x_1) \right]_{x_0}^{x_2} - \frac{1}{2} \int_{x_0}^{x_2} (x-x_0)^2 dx \\ &= 2h^3 - \frac{8}{6} h^3 = \frac{2}{3} h^3.\end{aligned}$$

og

$$\begin{aligned}\int_{x_0}^{x_3} (x-x_0)(x-x_1)(x-x_2) dx &= \left[\frac{1}{2} (x-x_0)^2 (x-x_1)(x-x_2) \right]_{x_0}^{x_3} - \frac{1}{2} \int_{x_0}^{x_3} (x-x_0)^2 ((x-x_1) + (x-x_2)) dx \\ &= 9h^4 - \frac{1}{6} [(x-x_0)^3 ((x-x_1) + (x-x_2))]_{x_0}^{x_3} + \frac{1}{6} \int_{x_0}^{x_3} 2(x-x_0)^3 dx \\ &= 9h^4 - \frac{27}{2} h^4 + \frac{1}{24} [(x-x_0)^4]_{x_0}^{x_3} \\ &= -\frac{9}{2} h^4 + \frac{27}{8} h^4 = -\frac{9}{8} h^4\end{aligned}$$

Seksjon 3.1

Exercise 1. Polynomet som interpolerer f i a og $a-h$ er linja $p_1(x) = f(a) + \frac{f(a)-f(a-h)}{h}(x-a)$. Denne har derivert $(f(a)-f(a-h))/h$, som er den venstresidige varianten til newtonkvotienten.

Exercise 3(a). Vi skriver nå

$$f(a+h) = f(a) + hf'(a) + \frac{h^2}{2} f''(a) + \frac{h^3}{6} f'''(\xi_h)$$

for en ξ_h i intervallet fra a til $a+h$. Vi kan nå skrive

$$f'(a) - \frac{f(a+h) - f(a)}{h} = -\frac{h}{2} f''(a) - \frac{h^2}{6} f'''(\xi_h).$$

Etter at vi tar absoluttverdier blir feilestimatet

$$\frac{h}{2} |f''(a)| + \frac{h^2}{6} \max_{x \in [a, a+h]} |f'''(x)|.$$

Dette feilestimatet ligner på det vi fikk da vi brukte et Taylorutvikling av orden én, men vi trenger også en generelt estimat på den tredjederiverte. Vi trenger fremdeles et estimat på den andreriverte, men bare i punktet a .

Exercise 3(b). Vi skriver nå $f(a+h) = f(a) + f'(\xi_h)h$ for en ξ_h i intervallet mellom a og $a+h$. I dette uttrykket er ikke $f'(a)$ til stede, så det er ikke mulig å få et estimat på trunkeringsfeilen med en slik Taylorutvikling.

Exercise 3(c). Det lineære Taylorpolynomet er best siden det er den korteste mulige Taylorutviklingen som kan gi et estimat for $f'(a)$ (som vi viste i (b)), og også det som gir det enkleste uttrykket for trunkeringsfeilen, på den måten at den ikke er avhengig av deriverte av høyere orden enn to (som vi viste i (a)).

Seksjon 3.2

Exercise 1(a). Tilnærmingen er eksakt for alle polynomer av grad 3 siden $M_4 = 0$ for disse (den fjerdederiverte av et tredjegradspolynom er 0). Hvis vi prøver med $f(x) = x^4$ blir tilnærmingen $\frac{h^4 + h^4}{h^2} = h^2$, som er forskjellig fra $f''(0) = 0$.

Exercise 1(b). $\frac{h^2}{12} \max_{x \in [-h, h]} |f^{(4)}(x)|$ er riktig svar

Exercise 2(a). Sant. Siden $M_3 = 0$ for alle andregradspolynomer, og siden den symmetriske newtonkvotienten har trunkeringsfeil $\leq M_3 \frac{h^2}{6}$, så er den symmetriske newtonkvotienten eksakt for alle andregradspolynomer.

Exercise 2(d). Sant. Siden $M_5 = 0$ for alle fjerdegradspolynomer, og siden firepunktsmetoden har trunkeringsfeil $\leq \frac{M_5}{18} h^4$, så er firepunktsmetoden eksakt for alle fjerdegradspolynomer. Hvis vi prøver med $f(x) = x^5$ så blir tilnærmingen til den deriverte i 0

$$\frac{(-2h)^5 - 8(-h)^5 + 8h^5 - (2h)^5}{12h} = \frac{h^5(-32 + 8 + 8 - 32)}{12h} = -4h^4,$$

mens $f'(0) = 0$, slik at metoden ikke er eksakt for $f(x) = x^5$.

Exercise 4(a). Det er kanskje enklest å vise først at dette holder for de tre funksjonene $f(x) = 1$, $f(x) = x$, og $f(x) = x^2$:

- For $f(x) = 1$ er $f'(a) = 0$, og den symmetriske Newtonkvotienten er

$$\frac{1 - 1}{2h} = 0.$$

- For $f(x) = x$ er $f'(a) = 1$, og den symmetriske Newtonkvotienten er

$$\frac{a + h - (a - h)}{2h} = 1.$$

- For $f(x) = x^2$ er $f'(a) = 2a$, og den symmetriske Newtonkvotienten er

$$\frac{(a + h)^2 - (a - h)^2}{2h} = \frac{4ah}{2h} = 2a.$$

Den symmetriske Newtonkvotienten blir da også riktig for alle andregradspolynomer, siden alle slike kan skrives som lineære kombinasjoner av 1, x , og x^2 , og siden derivasjon og Symmetrisk Newtonkvotient er lineære operasjoner.

Exercise 4(b). Vi har at

$$\begin{aligned} \frac{f_r(a) + f_l(a)}{2} &= \frac{\frac{f(a+h) - f(a)}{h} + \frac{f(a) - f(a-h)}{h}}{2} \\ &= \frac{f(a+h) - f(a) + f(a) - f(a-h)}{2h} = \frac{f(a+h) - f(a-h)}{2h}, \end{aligned}$$

som er den symmetriske Newtonkvotienten.

Exercise 5. Vi skal løse disse to deloppgavene på tre forskjellige måter. Først skal vi finne tilnærmingene ved å derivere det interpolerende polynomet. Deretter skal vi bruke Teorem 3.4. Til slutt utleder vi de samme tilnærmingene ved å kombinere Taylorrekker av passende orden i de gitte punktene

Exercise 5(a). Newtonformen til polynomet som interpolerer i $a-h$, a , og $a+2h$ er

$$p_2(x) = c_0 + c_1(x - (a - h)) + c_2(x - a)(x - (a - h)).$$

Vi evaluerer dette i de tre punktene, og løser for c_0, c_1, c_2 :

$$f(a - h) = c_0$$

$$f(a) = c_0 + c_1 h f(a + 2h) = c_0 + 3c_1 h + 6c_2 h^2.$$

Løser vi dette får vi først $c_0 = f(a - h)$, $c_1 = \frac{f(a) - f(a - h)}{h}$, og så

$$c_2 = \frac{f(a + 2h) - f(a - h) - 3(f(a) - f(a - h))}{6h^2} = \frac{f(a + 2h) - 3f(a) + 2f(a - h)}{6h^2}.$$

Dermed får vi

$$p_2(x) = f(a - h) + \frac{f(a) - f(a - h)}{h}(x - (a - h)) + \frac{f(a + 2h) - 3f(a) + 2f(a - h)}{6h^2}(x - a)(x - (a - h)),$$

og så

$$p_2'(x) = \frac{f(a) - f(a - h)}{h} + \frac{f(a + 2h) - 3f(a) + 2f(a - h)}{6h^2}((x - a) + (x - (a - h))),$$

og til slutt

$$p_2'(a) = \frac{f(a) - f(a - h)}{h} + \frac{f(a + 2h) - 3f(a) + 2f(a - h)}{6h} = \frac{f(a + 2h) + 3f(a) - 4f(a - h)}{6h}.$$

Teorem 3.4 gir i dette tilfellet (3.11) at $V = \begin{pmatrix} -1 & 2 \\ 1 & 4 \end{pmatrix}$. Vi har $V^{-1} = \begin{pmatrix} -2/3 & 1/3 \\ 1/6 & 1/6 \end{pmatrix}$,

og $V^{-T} = \begin{pmatrix} -2/3 & 1/6 \\ 1/3 & 1/6 \end{pmatrix}$, slik at

$$(-V^{-T}\mathbf{e} \quad V^{-T}) = \begin{pmatrix} 1/2 & -2/3 & 1/6 \\ -1/2 & 1/3 & 1/6 \end{pmatrix}$$

Første raden her sier at en tilnærming til $f'(a)$ blir

$$\frac{f(a)/2 - 2f(a - h)/3 + f(a + 2h)/6}{h} = \frac{3f(a) - 4f(a - h) + f(a + 2h)}{6h}.$$

Alternativt kan vi skrive opp

$$\begin{aligned} f(a - h) &= f(a) - f'(a)h + \frac{f''(a)}{2}h^2 + \frac{f'''(c_1)}{6}h^3 \\ f(a + 2h) &= f(a) + 2f'(a)h + 2f''(a)h^2 + 4\frac{f'''(c_2)}{3}h^3 \end{aligned}$$

Ganger vi den første likningen med 4 og trekker de fra hverandre (dette eliminerer $f''(a)$) får vi at

$$4f(a-h) - f(a+2h) = 3f(a) - 6f'(a)h + CM_3h^3,$$

der C er et tall. Dette gir tilnærmingen $f'(a) \approx \frac{3f(a)-4f(a-h)+f(a+2h)}{6h}$, der feilen er $\approx CM_3h^2$.

Exercise 5(b). Newtonformen til polynomet som interpolerer i a , $a+h$, og $a+2h$ er

$$p_2(x) = c_0 + c_1(x-a) + c_2(x-a)(x-(a+h)).$$

Vi evaluerer dette i de tre punktene, og løser for c_0, c_1, c_2 :

$$\begin{aligned} f(a) &= c_0 \\ f(a+h) &= c_0 + c_1h + c_2h^2 \\ f(a+2h) &= c_0 + c_1(2h) + c_2(2h)^2 \end{aligned}$$

Løser vi dette får vi først $c_0 = f(a)$, $c_1 = \frac{f(a+h)-f(a)}{h}$, og så

$$c_2 = \frac{f(a+2h) - f(a) - 2f(a+h) + 2f(a)}{2h^2} = \frac{f(a+2h) - 2f(a+h) + f(a)}{2h^2}.$$

Dermed får vi

$$p_2(x) = f(a) + \frac{f(a+h)-f(a)}{h}(x-a) + \frac{f(a+2h)-2f(a+h)+f(a)}{2h^2}(x-a)(x-(a+h)),$$

og så

$$p_2'(x) = \frac{f(a+h)-f(a)}{h} + \frac{f(a+2h)-2f(a+h)+f(a)}{2h^2}((x-a) + (x-(a+h))),$$

og til slutt

$$p_2'(a) = \frac{f(a+h)-f(a)}{h} - \frac{f(a+2h)-2f(a+h)+f(a)}{2h} = \frac{-f(a+2h)+4f(a+h)-3f(a)}{2h}$$

Teorem 3.4 gir i dette tilfellet (3.11) at $V = \begin{pmatrix} 1 & 2 \\ 1 & 4 \end{pmatrix}$. Vi har $V^{-1} = \begin{pmatrix} 2 & -1 \\ -1/2 & 1/2 \end{pmatrix}$,

og $V^{-T} = \begin{pmatrix} 2 & -1/2 \\ -1 & 1/2 \end{pmatrix}$, slik at

$$(-V^{-T}\mathbf{e} \quad V^{-T}) = \begin{pmatrix} -3/2 & 2 & -1/2 \\ 1/2 & -1 & 1/2 \end{pmatrix}$$

Første raden her sier at en tilnærming til $f'(a)$ blir

$$\frac{-3f(a)/2 + 2f(a+h) - f(a+2h)/2}{h} = \frac{-3f(a) + 4f(a+h) - f(a+2h)}{2h}.$$

Alternativt kan vi skrive opp

$$f(a+h) \approx f(a) + f'(a)h + \frac{f''(a)}{2}h^2 + \frac{f'''(c_1)}{6}h^3$$

$$f(a+2h) \approx f(a) + 2f'(a)h + 2f''(a)h^2 + 4\frac{f'''(c_2)}{3}h^3$$

Ganger vi den første likningen med 4 og trekker de fra hverandre (dette eliminerer $f''(a)$) får vi at

$$4f(a+h) - f(a+2h) = 3f(a) + 2f'(a)h + CM_3h^3,$$

der C er et tall. Dette gir tilnærmingen $f'(a) \approx \frac{-3f(a)+4f(a+h)-f(a+2h)}{2h}$, der feilen er $\approx CM_3h^2$ også her.

Exercise 6(a). Hvis tilnærmingsmetoden $f'(a) \approx c_1f(a-h) + c_2f(a+h)$ skal være eksakt for $f(x) = 1$, så må vi ha at $0 = c_1 + c_2$, siden $f(a-h) = f(a+h) = 1$, og siden $f'(x) = 0$. Derfor må vi ha at $c_2 = -c_1$.

Hvis metoden skal være eksakt for $f(x) = x$ så må vi på samme måte ha at

$$1 = c_1(a-h) + c_2(a+h) = c_1(a-h) - c_1(a+h) = -2c_1h,$$

slik at $c_1 = -\frac{1}{2h}$, slik at også $c_2 = \frac{1}{2h}$. Metoden blir derfor $-\frac{1}{2h}f(a-h) + \frac{1}{2h}f(a+h) = \frac{f(a+h)-f(a-h)}{2h}$

Exercise 6(b). Hvis $f(x) = cx + d$ så har vi at $f'(x) = c$, og metoden blir

$$\frac{f(a+h)-f(a-h)}{2h} = \frac{c(a+h)+d-(c(a-h)+d)}{2h} = \frac{2ch}{2h} = c,$$

slik at metoden er eksakt for alle polynomer av grad ≤ 1 . Vi ser at metoden faller sammen med den symmetriske Newtonkvotienten, og den har derfor en feil av orden h^2 , som er bedre enn Newtonkvotienten (som har en feil av orden h). Den er verre enn firepunktsmetoden, som har orden h^4 .

Metoden også er eksakt for polynomer av grad ≤ 2 . Det er flere måter å se dette på. For det første, feilestimatet fra den seksjonen bruker $f^{(3)}(x)$, og siden alle andregradspolynomer har en tredjederivert lik 0, så må feilen også være 0. Man kan også som over substituere $f(x) = x^2$ i formelen:

$$\frac{f(a+h)-f(a-h)}{2h} = \frac{(a+h)^2-(a-h)^2}{2h} = \frac{4ah}{2h} = 2a,$$

som også er $f'(a)$. Til slutt, den symmetriske Newtonkvotienten var definert som den deriverte i a til den unike parabolen som interpolerer f i $a-h$, a , og $a+h$. Hvis f selv er en parabel så er den lik interpolanten siden denne er unik, slik at den symmetriske Newtonkvotienten må returnere den deriverte.

Exercise 6(c). Hvis approksimasjonsmetoden $f''(a) \approx c_1 f(a-h) + c_2 f(a) + c_3 f(a+h)$ er eksakt for $f(x) = 1$, så må vi ha at $0 = c_1 + c_2 + c_3$. Hvis den er eksakt for $f(x) = x$ så må vi ha at

$$0 = c_1(a-h) + c_2 a + c_3(a+h) = a(c_1 + c_2 + c_3) + h(-c_1 + c_3) = h(-c_1 + c_3),$$

som gir at $c_1 = c_3$. Hvis den er eksakt for $f(x) = x^2$ så må vi ha at

$$\begin{aligned} 2 &= c_1(a-h)^2 + c_2 a^2 + c_3(a+h)^2 \\ &= a^2(c_1 + c_2 + c_3) - 2ahc_1 + 2ahc_3 + h^2(c_1 + c_3) = 2c_1 h^2, \end{aligned}$$

som gir at $c_1 = \frac{1}{h^2}$. Vi får derfor også at $c_3 = \frac{1}{h^2}$, og at $c_2 = -c_1 - c_3 = -\frac{2}{h^2}$, slik at metoden blir

$$\frac{1}{2h} f(a-h) - \frac{1}{h} f(a) + \frac{1}{2h} f(a+h) = \frac{f(a-h) - 2f(a) + f(a+h)}{h^2}.$$

Vi ser at dette faller sammen med trepunktsmetoden for den andrederiverte som vi allerede har sett på.

Exercise 6(d). Alle tredjegradspolynomer har fjerdederivert lik 0, og derfor blir trunkeringsfeilen 0 ($M_4 = 0$). Alternativt kan vi substituere $f(x) = x^3$ i formelen:

$$\begin{aligned} &\frac{f(a-h) - 2f(a) + f(a+h)}{2h} \\ &= \frac{(a-h)^3 - 2a^3 + (a+h)^3}{h^2} \\ &= \frac{a^3 - 3a^2h + 3ah^2 - h^3 - 2a^3 + a^3 + 3a^2h + 3ah^2 + h^3}{h^2} \\ &= \frac{6ah^2}{h^2} = 6a, \end{aligned}$$

som faller sammen med den andrederiverte til f i a .

Exercise 7(a). Vi kan tegne grafen sammen med sekantene som følger:

```
x = 0:0.05:6;
f = @(x) (-x.^2+10*x-5)/4;
plot(x,f(x))
hold on
plot([1 3],[f(1) f(3)])
plot([1 5],[f(1) f(5)])
plot([3 5],[f(3) f(5)])
```

Seksjon 4.1

Exercise 1(a). På hvert delintervall antar f minste verdi i venstre endepunkt, største verdi i høyre endepunkt. Her kan vi bruke `sum`-kommandoen på følgende måte til å finne $\underline{I}$ og $\bar{I}$.

```
function [l,u,l_err,u_err]=int_vals(n)
    l = (1/n)*sum(exp((0:(n-1))/n));
    u = (1/n)*sum(exp((1:n)/n));
    l_err = abs(l-(exp(1)-1));
    u_err = abs(u-(exp(1)-1));
```

Program 2. Utregning av øvre/nedre-integralet

kjører vi koden får vi

```
>> [l,u,l_err,u_err]=int_vals(10)

l =

    1.6338

u =

    1.8056

l_err =

    0.0845

u_err =

    0.0873
```

Exercise 1. Kjører vi koden

```
n = 1;
[l,u,l_err,u_err]=int_vals(n);
while l_err > 3*10^(-3)
    n = n+1;
    [l,u,l_err,u_err]=int_vals(n);
end
n
```

får vi at $n = 287$.

Seksjon 4.2

Exercise 2. Vi får at

$$I_{mid} = \frac{1}{2}f(1/4) + \frac{1}{2}f(3/4) = \frac{1}{2} \frac{1}{16} + \frac{1}{2} \frac{9}{16} = \frac{1}{2} \frac{10}{16} = \frac{5}{16},$$

slik at det første alternativet er riktig.

Exercise 3. Intervallet $[0, \pi/2]$ splittes her opp i intervallene

$$[0, \pi/12], [\pi/12, 2\pi/12], [2\pi/12, 3\pi/12], [3\pi/12, 4\pi/12], \\ [4\pi/12, 5\pi/12], [5\pi/12, 6\pi/12].$$

Midtpunktene på disse intervallene er $\pi/24, 3\pi/24, 5\pi/24, 7\pi/24, 9\pi/24, 11\pi/24$. Med $h = \pi/12$ gir midtpunktsmetoden

$$I_{mid} = \frac{\pi}{12} (f(\pi/24) + f(3\pi/24) + f(5\pi/24) + f(7\pi/24) + f(9\pi/24) + f(11\pi/24)) \\ = \frac{\pi}{12} \left(\frac{\sin(\pi/24)}{1 + (\pi/24)^2} + \frac{\sin(3\pi/24)}{1 + (3\pi/24)^2} + \frac{\sin(5\pi/24)}{1 + (5\pi/24)^2} + \frac{\sin(7\pi/24)}{1 + (7\pi/24)^2} \right. \\ \left. + \frac{\sin(9\pi/24)}{1 + (9\pi/24)^2} + \frac{\sin(11\pi/24)}{1 + (11\pi/24)^2} \right) \\ \approx 0.530624.$$

Exercise 4. Vi kan godt lage en funksjon her, som vi kan bruke på forskjellige input:

```
function I=midpointn(f, a, b, n)
    h = (b - a)/n;
    I=h*sum( f(a+h/2+(0:(n-1))*h) );
```

Program 3. Midtpunktsmetoden

Kode som løser oppgaven blir da som følger

```
midpointn(@exp,0,1,10)
```

Exercise 4(b). Fra Teorem 4.7 vet vi at feilen er begrenset av $(b-a) \frac{h^2}{24} \max_{x \in [a,b]} |f''(x)|$. Siden $f''(x) = e^x$, $\max_{x \in [0,1]} |f''(x)| = e$, slik at dette er $\frac{eh^2}{24}$. Vi må derfor sørge for at $\frac{eh^2}{24} < 10^{-10}$, slik at $h < \sqrt{\frac{24}{e}} 10^{-5} \approx 2.97 \times 10^{-5}$.

Seksjon 4.3

Exercise 1(a). Feil. Midtpunktsmetoden har feil begrenset av $(b-a)h^2 M_2/24$, mens det beste estimatet vi fant for trapesmetoden var $(b-a)h^2 M_2/12$.

Exercise 1(b). Feil. I stedet for å regne ut samme funksjonsverdi $f(x_i)$ to ganger så kan vi skrive $2f(x_i)$, siden alt skal summeres.

Exercise 1(c). Sant. For ett intervall trengs 3 delepunkter. For hvert intervall vi legger til ytterligere to delepunkter, slik at det totale antallet alltid blir odde.

Exercise 1(d). Sant, men dette bygger på Oppgave 9. Hvis vi bruker bare estimatet fra (4.12) så kan vi ikke konkludere med dette.

Exercise 2(a). Vi har at

$$I_{\text{trap}} = \frac{1}{2} \left(\frac{f(0) + f(1)}{2} + f\left(\frac{1}{2}\right) \right) = \frac{1}{2} \left(\frac{0+1}{2} + \frac{1}{4} \right) = \frac{3}{8},$$

slik at det andre alternativet er riktig.

Exercise 2(b). Alle er eksakte.

Exercise 2(c). Det følger fra (4.12) at Simpsons metod er eksakt for alle polynomer av grad to.

Exercise 2(d). Det siste alternativet er riktig.

Exercise 3. Vi kan godt lage en funksjon her, som vi kan bruke på forskjellige input:

```
function I=trapezoidaln(f, a, b, n)
    h = (b - a)/n;
    I=h*( (f(a) + f(b))/2 + sum( f(a+(1:(n-1))*h ) ) );
```

Program 4. Trapesmetoden

Kode som løser Oppgave 4 og 3 blir da som følger

```
trapezoidaln(@exp,0,1,10)
trapezoidaln(@(x) sin(x)./(1+x.^2), 0, pi/2, 6)
```

Feilen er begrenset av

$$(b-a) \frac{h^2}{12} \max_{x \in [0,1]} |f''(x)| = \frac{h^2}{12} e.$$

Vi har at $\frac{h^2}{12} e \leq 10^{-10}$ når $h \leq \sqrt{12/e} \times 10^{-5} \approx 2.1 \times 10^{-5}$.

Exercise 4(a). Siden feilen i trapesmetoden er begrenset av

$$(b-a) \frac{h^2}{12} \max_{x \in [a,b]} |f''(x)|,$$

og siden $f''(x) = \frac{8}{(1+2x)^3}$ (som får maksimum absolutt verdi for $x = 0$), så trenger vi å velge h slik at

$$(b-a) \frac{h^2}{12} \max_{x \in [a,b]} |f''(x)| = 8 \frac{h^2}{12} \leq 10^{-10},$$

som gir at $h \leq 10^{-5} \sqrt{3/2}$. Antall funksjonsevalueringer er da $\approx 1/h = 10^5 / \sqrt{2/3} \approx 81649.66$, som betyr at minst 81650 evalueringer trengs.

Exercise 4(b). Siden feilen i midtpunktsmetoden er begrenset av

$$(b-a) \frac{h^2}{24} \max_{x \in [a,b]} |f''(x)|,$$

så får vi som i (a) at

$$(b-a) \frac{h^2}{24} \max_{x \in [a,b]} |f''(x)| = 2 \frac{h^2}{6} \leq 10^{-10},$$

som gir at $h \leq 10^{-5} \sqrt{3}$. Antall funksjonsevalueringer er derfor omtrent $1/h = 10^5 / \sqrt{3} \approx 57735.03$. Med andre ord, minst 57736 evalueringer trengs.

Exercise 4(c). Siden feilen i Simpsons metode er begrenset av

$$(b-a) \frac{h^4}{180} \max_{x \in [a,b]} |f^{(iv)}(x)|,$$

og siden $f^{(iv)}(x) = \frac{384}{(1+2x)^5}$ (som får maksimum absolutt verdi for $x = 0$), så trenger vi velger h slik at

$$(b-a) \frac{h^4}{180} \max_{x \in [a,b]} |f^{(iv)}(x)| = 384 \frac{h^4}{180} \leq 10^{-10},$$

som betyr at h er begrenset av $h \leq (180 \cdot 10^{-10} / 384)^{1/4}$. Antall funksjonsevalueringer er da $\approx 1/h = (180 \cdot 10^{-10} / 384)^{-1/4} \approx 382.17$, som betyr at minst 383 evalueringer trengs.

Exercise 5. Vi kan godt lage en funksjon her, som vi kan bruke på forskjellige input:

```
function I=simpsonn(f, a, b, n)
    h = (b - a)/(2*n);
    I = h*(f(a) + f(b) + ...
        2*sum( f(a+2*h*(1:(n-1))) ) + ...
        4*sum( f(a+h+2*h*(0:(n-1)))))/3;
```

Program 5. Simpsons metode

Kode som løser Oppgave 4 og 3 blir da som følger

```
simpsonn(@exp,0,1,10)
simpsonn(@(x) sin(x)./(1+x.^2), 0, pi/2, 6)
```

Exercise 6. Siden $\int_a^b f(x) dx = w_1 f(a) + w_2 f((a+b)/2) + w_3 f(b)$ for $f(x) = x^i$ for $i = 0, 1, 2$, så har vi også at dette holder for polynomet p_2 som interpolerer f i a , $(a+b)/2$, og b . Derfor får vi at

$$\int_a^b p_2(x) dx = w_1 p_2(a) + w_2 p_2((a+b)/2) + w_3 p_2(b) = w_1 f(a) + w_2 f((a+b)/2) + w_3 f(b).$$

Vi har også vist at det første er lik $\frac{b-a}{6}(f(a) + 4f((a+b)/2) + f(b))$, slik at

$$w_1 f(a) + w_2 f((a+b)/2) + w_3 f(b) = \frac{b-a}{6}(f(a) + 4f((a+b)/2) + f(b)).$$

Ved å sette $f(a) = 1$, $f((a+b)/2) = f(b) = 0$ ser vi at $w_1 = \frac{b-a}{6}$, og tilsvarende for w_2 , w_3 . Det følger at metoden er lik Simpsons metode.

Exercise 7(a). Delvis integrasjon på (4.13) med constant C i den antideriverte av $x - m$ gir

$$\begin{aligned} L &= - \left[\left(C + \frac{1}{2}(x-m)^2 \right) f'(x) \right]_a^b + \int_a^b \left(C + \frac{1}{2}(x-m)^2 \right) f''(x) dx \\ &= \int_a^b \left(C + \frac{1}{2}(x-m)^2 \right) f''(x) dx. \end{aligned}$$

som viser (4.14).

Exercise 7(b). For $x \in [a, b]$ har vi at $\frac{1}{2}(x-m)^2 \leq \frac{1}{8}(b-a)^2 = -C$, slik at $-C - \frac{1}{2}(x-m)^2 \geq 0$ for slike x . Tar vi absoluttverdier i (4.14) med dette valget av C får vi

$$\begin{aligned} |L| &= \left| \int_a^b \left(\frac{1}{2}(x-m)^2 - (b-a)^2/8 \right) f''(x) dx \right| \\ &\leq \int_a^b \left| \frac{1}{2}(x-m)^2 - (b-a)^2/8 \right| |f''(x)| dx \\ &\leq \int_a^b \left((b-a)^2/8 - \frac{1}{2}(x-m)^2 \right) M_2 dx \\ &= M_2 \left[x(b-a)^2/8 - \frac{1}{6}(x-m)^3 \right]_a^b \\ &= M_2 \left((b-a)^3/8 - \frac{1}{48}(b-a)^3 - \frac{1}{48}(b-a)^3 \right) \\ &= M_2 \left(\frac{1}{8} - \frac{1}{24} \right) (b-a)^3 = \frac{M_2(b-a)^3}{12}. \end{aligned}$$

Exercise 7(c). Som for midtpunktsmetoden så får man den globale feilen ved å summere de lokale feilene (erstatt $b - a$ med h , og nh med $b - a$):

$$\sum_{k=1}^n \frac{M_2 h^3}{12} = n \frac{M_2 h^3}{12} = (b - a) \frac{h^2}{12} M_2.$$

Dette var det vi skulle vise.

Exercise 8(a). Det er nok å verifisere Simpsons metode på hvert delintervall. Integralene blir

$$\begin{aligned} \int_{a-h}^{a+h} x^3 dx &= \frac{1}{4}((a+h)^4 - (a-h)^4) = 2a^3 h + 2ah^3 \\ \int_{a-h}^{a+h} x^2 dx &= \frac{1}{3}((a+h)^3 - (a-h)^3) = \frac{1}{3}(6a^2 h + 2h^3) = \frac{h}{3}(6a^2 + 2h^2) \\ \int_{a-h}^{a+h} x dx &= \frac{1}{2}((a+h)^2 - (a-h)^2) = \frac{1}{2}4ah = 2ah \\ \int_{a-h}^{a+h} dx &= 2h. \end{aligned}$$

Vi får også at metoden blir

$$\begin{aligned} \frac{h}{3}((a-h)^3 + 4a^3 + (a+h)^3) &= \frac{h}{3}(6a^3 + 6ah^2) = 2a^3 h + 2ah^3 \\ \frac{h}{3}((a-h)^2 + 4a^2 + (a+h)^2) &= \frac{h}{3}(6a^2 + 2h^2) \\ \frac{h}{3}((a-h) + 4a + (a+h)) &= 2ah \\ \frac{h}{3}(1 + 4 + 1) &= 2h. \end{aligned}$$

Dette viser at Simpsons metode er eksakt for de gitte polynomene.

Exercise 8(b). For $f(x) = bx^3 + cx^2 + dx + e$ blir integralet

$$\int_{a-h}^{a+h} f(x) dx = b \int_{a-h}^{a+h} x^3 dx + c \int_{a-h}^{a+h} x^2 dx + d \int_{a-h}^{a+h} x dx + e \int_{a-h}^{a+h} 1 dx.$$

Metoden blir nå

$$\begin{aligned} &\frac{h}{3}(f(a-h) + 4f(a) + f(a+h)) \\ &= b \frac{h}{3}((a-h)^3 + 4a^3 + (a+h)^3) + c \frac{h}{3}((a-h)^2 + 4a^2 + (a+h)^2) \\ &\quad + d \frac{h}{3}((a-h) + 4a + (a+h)) + e \frac{h}{3}(1 + 4 + 1) \end{aligned}$$

Likhet følger nå fra at vi har likhet for x^3 , x^2 , x , og 1 .

Exercise 8(c). Siden $f^{(4)}(x) = 0$ for alle x for ethvert kubisk polynom, så følger det direkte fra feilestimatet at metoden er eksakt for slike funksjoner.

Exercise 8(d). Vi har at

$$\begin{aligned} q(m-y) &= (h-y)(-y)(-h-y) \\ q(m+y) &= (h+y)y(-h+y) = -(h-y)(-y)(-h-y), \end{aligned}$$

slik at $q(m-y) = -q(m+y)$. Det følger at q er anti-symmetrisk om m , slik at $\int_a^b q(x) dx = 0$. Anta nå at f er et polynom av grad 3. Newtonformene er

$$\begin{aligned} p_2(x) &= c_0 + c_1(x-a) + c_2(x-a)(x-m) \\ f(x) &= c_0 + c_1(x-a) + c_2(x-a)(x-m) + c_3(x-a)(x-m)(x-b) \end{aligned}$$

Siden $\int_a^b (x-a)(x-m)(x-b) dx = \int_a^b q(x) dx = 0$ så følger det at $\int_a^b f(x) dx = \int_a^b p_2(x) dx$, slik at Simpsons metode for polynomer av grad 3.

Exercise 9(a). Delvis integrasjon tre ganger gir

$$\begin{aligned} \int_a^b xq(x) dx &= \int_a^b x(x-a)(x-m)(x-b) dx = -\frac{1}{2} \int_a^b (x-a)^2(x-m)(x-b) dx \\ &= \frac{1}{6} \int_a^b (x-a)^3(x-b) dx \\ &= -\frac{1}{24} \int_a^b (x-a)^4 dx \\ &= -\frac{1}{120} (b-a)^5, \end{aligned}$$

som viser første del. At $(xq(x))^{(4)} = 4!$ følger ved å bruke regelen for å derivere et produkt fire ganger.

Exercise 9(b). Vi har at $p_2^{(4)}(x) = q^{(4)}(x) = 0$. Innsetting av $(xq(x))^{(4)} = 4!$, og $\int_a^b xq(x) dx = -\frac{1}{120} (b-a)^5$ fra (a), viser første del. For den andre delen, vi vet fra før at $\int_a^b q(x) dx = 0$. Integrasjon av φ fra a til b gir derfor

$$\int_a^b f(x) dx - \int_a^b p_2(x) dx - E = E - E = 0.$$

Den siste delen følger fra at f og p_2 stemmer overens på a , m , og b , og fra at $q(a) = q(m) = q(b) = 0$.

Exercise 9(c). $\int_a^m \varphi(x) dx = 0$ er det samme som

$$0 = \int_a^m f(x) dx - \int_a^m p_2(x) dx - k \int_a^m q(x) dx - \frac{E}{\int_a^b xq(x) dx} \int_a^m xq(x) dx,$$

som kan løses for k ved dele med $\int_a^m q(x) dx \neq 0$ (som er $\neq 0$ siden q ikke har noe fortegnsskifte på (a, m)).

Exercise 9(d). Siden $\int_a^b \varphi(x) dx = \int_a^m \varphi(x) dx = 0$, så må vi ha at $\int_m^b \varphi(x) dx = 0$. Siden φ er kontinuerlig, så finnes det en $u \in (a, m)$ slik at $\varphi(u) = 0$, og en $v \in (m, b)$ slik at $\varphi(v) = 0$. φ har derfor minst fem forskjellige nullpunkter i $[a, b]$, siden vi også viste at $\varphi(a) = \varphi(m) = \varphi(b) = 0$.

Exercise 9(e). Du kan argumentere som følger

1. Bruker du Rolles teorem på de fire intervallene (a, u) , (u, m) , (m, v) , og (v, b) , så får vi fire punkter $a_1 < a_2 < a_3 < a_4$, der alle $\varphi'(a_i) = 0$.
2. Bruker du Rolles teorem igjen, men nå på de tre intervallene (a_1, a_2) , (a_2, a_3) , og (a_3, a_4) , så får vi tre punkter $b_1 < b_2 < b_3$, der alle $\varphi''(b_i) = 0$.
3. Bruker du Rolles teorem en tredje gang, nå på de to intervallene (b_1, b_2) , (b_2, b_3) , så får vi to punkter $c_1 < c_2$ slik at $\varphi'''(c_1) = \varphi'''(c_2) = 0$.
4. Bruker du Rolles teorem en fjerde og siste gang så får vi en $\xi \in (c_1, c_2)$ slik at $\varphi^{(4)}(\xi) = 0$,

Exercise 9(f). Evaluering av (4.15) i ξ gir

$$0 = f^{(4)}(\xi) + 120 \frac{4!E}{(b-a)^5},$$

slik at $E = -f^{(4)}(\xi)(b-a)^5/2880$. En øvre grense for feilen på ett intervall blir derfor $M_4(b-a)^5/2880$.

Exercise 10(a). Newtonformen får vi ved å løse likningene

$$f(a) = c_0$$

$$f(m) = c_0 + c_1 h$$

$$f(b) = c_0 + 2c_1 h + 2c_2 h^2.$$

Vi får $c_0 = f(a)$, $c_1 = \frac{f(m)-f(a)}{h}$, $c_2 = \frac{f(a)-2f(m)+f(b)}{2h^2}$. Dermed blir det interpolerende polynomet

$$p_2(x) = f(a) + \frac{f(m)-f(a)}{h}(x-a) + \frac{f(a)-2f(m)+f(b)}{2h^2}(x-a)(x-m).$$

Exercise 10(b). Her kan det være til hjelp først å regne ut $\int_a^b (x-a)(x-m) dx$ ved hjelp av delvis integrasjon:

$$\int_a^b (x-a)(x-m) dx = \left[\frac{1}{2}(x-a)^2(x-m) \right]_a^b - \frac{1}{2} \int_a^b (x-a)^2 dx = 2h^3 - \frac{4}{3}h^3 = \frac{2}{3}h^3.$$

Integrerer vi så $p_2(x)$ får vi

$$\begin{aligned} & \left[f(a)(x-a) + \frac{f(m)-f(a)}{2h}(x-a)^2 \right]_a^b + \frac{f(a)-2f(m)+f(b)}{2h^2} \frac{2}{3} h^3 \\ &= 2f(a)h + 2(f(m)-f(a))h + \frac{f(a)-2f(m)+f(b)}{h} \frac{1}{3} h \\ &= h \left(2f(m) + \frac{1}{3}(f(a)-2f(m)+f(b)) \right) \\ &= \frac{h}{3} (6f(m) + f(a) - 2f(m) + f(b)) \\ &= \frac{h}{3} (f(a) + 4f(m) + f(b)). \end{aligned}$$

Exercise 11. Vi kan skrive den om som følger.

$$I_{\text{trap}}(h) = h \left(\frac{f(a)+f(b)}{2} + \sum_{i=1}^{n-1} f(x_i) \right).$$

Seksjon 5.1

Exercise 1. Siden A er inverterbar er alle $\lambda_i \neq 0$. Hvis vi transponerer begge sider i $A^{-1}A = I$ får vi at $A^T(A^{-1})^T = I$. Siden A er symmetrisk betyr dette at $A(A^{-1})^T = I$, slik at $A^{-1} = (A^{-1})^T$. Det følger at A^{-1} også er symmetrisk.

Anta $A\mathbf{v} = \lambda\mathbf{v}$. Da må $\mathbf{v} = A^{-1}(\lambda\mathbf{v})$, slik at $A^{-1}\mathbf{v} = \frac{1}{\lambda}\mathbf{v}$. Det følger at $1/\lambda$ er en egenverdi for A^{-1} når λ er en egenverdi for A . A og A^{-1} har også de samme egenvektorene.

Dette følger også lett fra at A er ortogonalt diagonaliserbar, slik at vi kan skrive $A = PDP^{-1}$, der D er diagonalmatrisen med egenverdiene λ_i på diagonalen. Fra dette følger at $A^{-1} = PD^{-1}P^T$, der D^{-1} er diagonalmatrisen med $1/\lambda_i$ på diagonalen.

Exercise 4. Det er klart at

$$|U\mathbf{x}|^2 = \mathbf{x}^T U^T U \mathbf{x} = |\mathbf{x}|^2.$$

Det følger at, hvis $\mathbf{x}$ er en egenvektor, så er $|U\mathbf{x}| = |\lambda||\mathbf{x}| = |\mathbf{x}|$, slik at $|\lambda| = 1$. Siden alle egenverdiene er like i absoluttverdi følger det at $K(A) = 1$. Merk at vi ikke trenger å gå via ortogonal diagonaliserbarhet her.