MATLAB-OPPGAVER I IGR1601 MATEMATIKK 2
VÅREN 2017

UIT NORGES ARKTISKE UNIVERSITET

Irina Pettersson, UiT.

Innhold

Introduksjon

Dette heftet supplerer pensumet i IGR1601 Matematikk 2 med relevante MATLAB oppgaver. Dette er verken en tekstbok i Matematikk 2 eller i MATLAB som programmeringsspråk. Det er tenkt at man har grunnleggende kunnskap om MATLABs funksjoner og har gått gjennom de tilsvarende delene i Matematikk 2 innen man begynner med disse oppgavene.

Formålet med heftet er å vise eksempler på hvordan man kan bruke MATLAB for å kontrollere svaret eller få en idé hva svaret skal være, samt å gi øvingsoppgaver. Fasit til alle oppgavene er gitt, men vi oppmuntrer alltid studentene å forsøke først selv.

De innebygde funksjonene som er brukt er ikke forklart, men om man klikker på navnet til en funksjon der den er nevnt for første gang, kommer man til en webside på mathworks.com der man kan lese mer om den funksjonen. Det er tenkt at noen av dem er kjent fra før og alle andre skal man lese om i Help.

Det finnes alltid flere måter å løse en oppgave på. For eksempel kan noen oppgaver løses ved hjelp av innebygde funksjoner eller ved manuell programmering med en løkke (som i Eksempel 1.1). De innebygde funksjoner er bra å kunne, men det å kunne lage en løkke er å regne som en naturlig del av programmeringskulturen og er en viktig del av ingeniørfaget generelt.

Kapittel 1.3 handler om numerisk derivasjon. Man kan hoppe over den ved første lesning, men numerisk derivasjon er et veldig viktig tema og er anbefalt å lese om selv om symbolsk derivasjon er enklest å bruke i dette kurset.

Det kan forekomme endringer i innholdet underveis.

Alle kommentarer og forslag som gjelder dette heftet mottas med takk av Irina Pettersson, irina.pettersson@uit.no.

1. Rekker

1.1. Endelige og uendelige summer. Vi begynner ved å gi et enkelt eksempel der vi regner ut en endelig sum.

Eksempel 1.1. Lag en MATLAB-kode som regner ut summen $\sum_{n=1}^{4} n(n+2)$.

Løsning. Vi setter først summen lik null og etterpå regner ut summen i en løkke med hensyn på n, fra $n = 1$ til $n = 4$. Ta bort semikolonet etter i linja 7 om du vil skrive ut delsummene.

```
1  % Compute the sum \sum_{n=1}^{4} n(n+2)
2  close all; clear all;
3  % Initial value for S
4  S = 0;
5  % Compute the sum in a loop
6  for n=1:4
7      S = S + n*(n+2);
8  end
9  % Print the answer
10 fprintf('Sum = %d\n', S);
```

Output:

```
>> Sum = 50
```

□

Samme sum kan vi regne ut ved å bruke `sum` eller `symsum` (en symbolsk sum).

```
>> n=1:4;
>> a_n=n.*(n+2);
>> sum(a_n)

ans = 50
```

Her definerer vi først en vektor n ($n = [1, 2, 3, 4]$), bruker den til å definere det allmenne leddet a_n og etterpå regner ut summen av elementene i vektoren a_n.

Vi kan også regne ut summen symbolsk:

```
>> syms n
>> S=symsum(n*(n+2), n, 1, 4)

S = 50
```

Funksjonen `symsum` regner ut summen av en følge definert ved $n(n+2)$ med hensyn på n fra $n = 1$ til $n = 4$.

Funksjonen `symsum` kan også brukes til å finne uendelige summer. For eksempel, summen

$$\sum_{1}^{\infty} \frac{1}{k^2} = 1 + \frac{1}{4} + \frac{1}{9} + \cdots = \frac{\pi^2}{6}.$$

I MATLAB:

```
>> syms  k;
>> symsum(1/k^2, k, 1, Inf)

ans = pi^2/6
```

1.2. **Taylorpolynomer.** La f være en n ganger deriverbar funksjon for $x = a$. Taylorpolynomet P_n av grad n må tilfredsstille betingelsene

$$P_n(a) = f(a), \quad P_n'(a) = f'(a), \ldots, P_n^{(n)}(a) = f^{(n)}(a). \tag{1}$$

Det finnes bare ett sånt polynom:

$$P_n(x) = f(x_0) + \frac{f'(x_0)}{1!}(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \cdots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n.$$

Det betyr at koeffisientene til $P(x)$ kan beregnes i termer av deriverte av f.

Jo høyre grad et polynom har, desto bedre approksimerer det den gitte funksjonen.

Taylorpolynomer er ikke vanskelig å beregne, men det kan være tidskrevende. I MATLAB finnes det en innebygd funksjon `taylor` som man kan bruke til å finne et taylorpolynom av en gitt orden.

Eksempel 1.2. Finn et taylorpolynom om punktet $x = 1$ til e^x av orden 2.

Løsning.

```
>> syms x
>> f = exp(x);
>> taylor(f,x,'ExpansionPoint', 1, 'Order', 3)

ans = exp(1) + exp(1)*(x - 1) + (exp(1)*(x - 1)^2)/2
```

Legg merke til at vi setter Order lik 3, og ikke 2. Funksjonen `taylor(f,x,'Order',n)` gir et taylorpolynom av orden $n - 1$.

□

For å se hvordan taylorpolynomene approksimerer en gitt funksjon, vil vi tegne grafene av både funksjonen og polynomene. For $y = e^x$ fra Eksempel 1.2 kan vi gjøre følgende:

```
% y=exp(x) and taylor polynomials
close all; clear all;
% Define a symbolic x
syms x
% Define the function and
% Taylor polynomials of order 1, 2 and 3
f = exp(x);
T1 = taylor(f,x,'Order',2)
T2 = taylor(f,x,'Order',3)
T3 = taylor(f,x,'Order',4)
% Plot the function and the polynomials
```

```
fplot([f, T1, T2, T3], [-1, 1], 'LineWidth',2);
legend('show', 'Location', 'best');
```

Output:

```
T1 = x + 1
T2 = x^2/2 + x + 1
T3 = x^3/6 + x^2/2 + x + 1
```

Grafene vises i Figur 1.

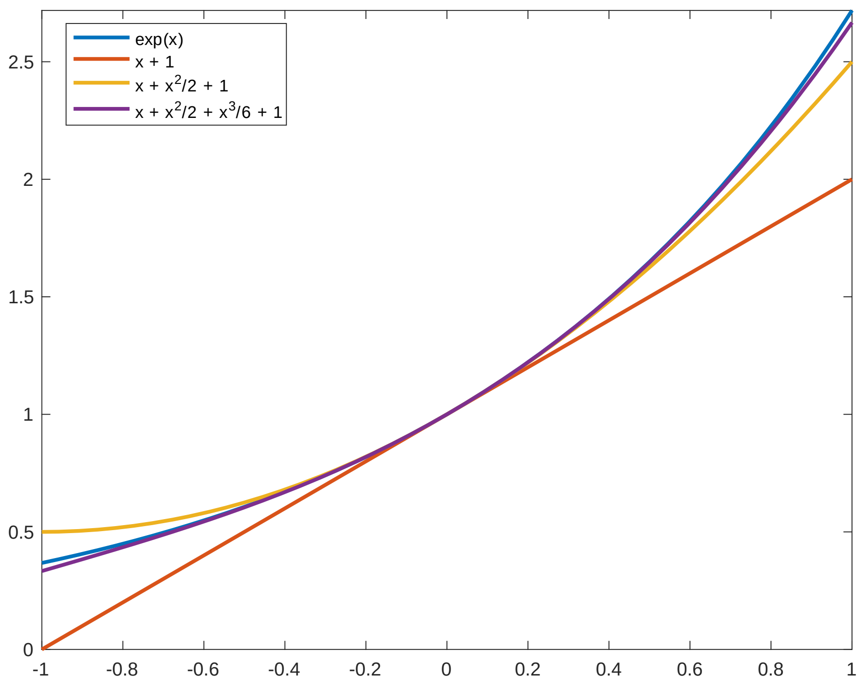

FIGURE 1. Approksimasjon av funksjonen $y = e^x$ med Taylor polynomer.

1.3. **Lite om numerisk derivasjon*.** I MATLAB finnes det muligheter for både numerisk (diff) og symbolsk derivasjon (symbolsk diff).

For eksempel, for å derivere $f(x) = (3x^2 + 5)^4$ symbolsk, definerer vi en symbolsk variabel x og bruker deretter `diff`.

```
>> syms x
>> diff((3*x^2 + 5)^4)

ans =

24*x*(3*x^2 + 5)^3
```

Selvklart kan man (og det er enklest i våre oppgaver) bruke symbolsk derivasjon for å finne et taylorpolynom, men vi skal benytte oss av denne muligheten og lære oss å derivere numerisk. Denne delkapittel kan du hoppe over i første omgang, men numerisk derivasjon er et kraftfullt verktøy som vil være nyttig i mer omfattende problemstillinger.

Eksempel 1.3. La $f(x) = e^{-x}$. Tegn grafen til $y = f(x)$ og grafen til taylorpolynomer P av grad mindre eller lik 2 (det vil si tre polynomer: av grad null, en og to) med

$$P^{(k)}(0) = f^{(k)}(0), \quad k = 0, 1, 2.$$

Dette eksemplet viser hvordan man approksimerer en funksjon med et taylorpolynom i et gitt punkt. Det er artig å se hvordan grafen til polynomet nærmer seg funskjonens graf når vi øker polynomets grad.

Løsning.

```matlab
% Approximate f=exp(-x) by a polynomial at x=0
close all;
h = 0.0001; % step size
x = -5:h:5;
x0 = 0;
f = exp(-x);
plot(x, f, 'LineWidth', 2); % plot f
xlim([-5, 5]); ylim([0, 10]);
hold on; grid on;axis equal;
% Find a point in x close to x0=2
[val, idx] = min(abs(x-x0))
% The value of f at this point
y0 = f(idx)
% Approximation of the derivative f'(0)
df = (f(idx+1)-f(idx))/h;
% Approximation of the 2. derivative f"(0)
ddf = (f(idx+1) - 2*f(idx)+f(idx-1))/h^2;
% Polynomial p=C1x+C0 with p(0)=f(0) and p'(0)=f'(0)
C0 = f(idx); % C0 = closest to f(0)
C1= df; % C1 = approximation of f'(0)
% 1. order polynomial
P1 = C1*x + C0;
% Print P1
fprintf('P1 = %fx + %f', C1, C0);
plot(x, P1, 'Color', 'r', 'LineWidth', 2); % Plot P1
% Polynomial p=C2* x^2 + C1*x+C0 with p(0)=f(0), p'(0)=f'(0)
% and p"(0) = f"(0)
C0 = f(idx); % C0 = approx. f(0)
C1 = df; % C1 = approx. f'(0)
C2 = ddf/2; % C2 = approx. f"(0)
% 2. order polynomial
P2 = C2*x.^2 + C1*x + C0;
% Print P2
fprintf('\nP2 = %f x^2 + %fx + %f', C2, C1, C0);
plot(x, P2, 'Color', 'g', 'LineWidth', 2); % Plot P2
legend('Kurven y=exp(-x)', '1. gradspolynom',...
         '2. gradspolynom','location', 'NorthEastOutside');
```

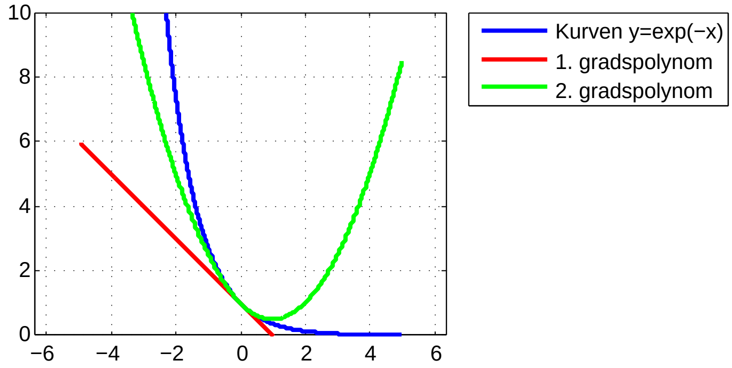

FIGURE 2. Eksempel 1.3.

I dette eksemplet regner vi ut den deriverte numerisk. Utregningen er basert på definisjonen av den deriverte:

$$f'(x_0) \approx \frac{f(x_0+h) - f(x_0)}{h}.$$

Problemet igjen er at x ikke nødvendigvis inneholder $x0 = 0$ (den gjør det likevel i vårt tilfelle pga at vi starter i -5 og velger $h = 0.0001$). Så vi skal finne et punkt i x som ligger nærmest x_0. For å gjøre det, tar vi alle punkter i x og ser på avstanden $abs(x - x_0)$. Avstanden blir minimal for det punktet som ligger nærmest. Variabelen `val` gir verdien av $abs(x - x_0)$, mens `idx` gir posisjonen av det nærmeste til x_0 punktet i x (les mer om funskjonen `min` i Documentation). Som du ser i **Output**, `val`$= 0$ (viser at x_0 er i x), og `idx`$= 50001$ (element nr. 50001 i x). Tenk at du tar utgangspunktet i -5 og går med skritt 0.0001, hvor mange steg skal du ta for å komme til 0?

Nå kan vi regne ut funskjonens verdi $f(idx)$ og approksimere den deriverte:

$$f'(idx) \approx \frac{f(idx+1) - f(idx)}{h}.$$

Husk at f er også en vektor, med samme lengde som x. Når vi skriver $f(idx)$, får vi den komponenten av f som tilsvarer elementet i x med nummer idx (det nærmeste punktet til $x_0 = 0$). Derimot gir $f(idx + 1)$ funskjonens verdi i neste punkt, $x_0 + h$.

Den andre deriverte $f''(x_0)$ approksimerer vi som følge:

$$f''(x_0) \approx \frac{f(x_0+h) - 2f(x_0) + f(x_0-h)}{h^2}.$$

Nå har vi de første to deriverte og kan konstruere polynomene. Som vi vet, et polynom $P(x)$ slik at

$$P(x_0) = f(x_0), P'(x_0) = f'(x_0), P''(x_0) = f''(x_0), \cdots P^{(n)}(x_0) = f^{(n)}(x_0)$$

kalles Taylorpolynomet av orden n. I vårt eksempel er $x_0 = 0$. For 1. gradspolynomet $C1x + C0$, må vi ha

$$C0 = f(0), \quad C1 = f'(0).$$

For 2. gradspolynomet $C2x^2 + C1x + C0$, har vi

$$C0 = f(0), \quad C1 = f'(0), \quad C2 = \frac{f''(0)}{2}.$$

Legg merke til at `df` og `ddf` er skalare (ikke vektorer som `f`), derfor skriver vi ikke noe argument (som i `f(idx)`).

Output:

```
val =
     0
idx =
       50001
y0 =
     1
P1 = -0.999950x + 1.000000
P2 = 0.500000 x^2 + -0.999950x + 1.000000
```

□

1.4. Følger og rekker. Konvergens.

Eksempel 1.4. Bruk MATLAB til å skrive ut de første fem leddene av følgen

$$a_n = \frac{\ln n}{n}.$$

Finn a_{50}, a_{150}.

Denne følgen konvergerer. Bruk flere verdier og prøv å gjette grenseverdien for denne følgen når $n \to \infty$. Regn ut grenseverdien analytisk (eller med MATLAB symbolsk) og sammenlikn resultatene.

Løsning. For å definere følgen lager vi først en vektor n med lengde 5 ved å skrive $n = 1:5$ og etterpå setter $a_n = \ln(n)/n$:

```
>> n=1:5;
>> a_n=log(n)./n

a_n = 0    0.3466    0.3662    0.3466    0.3219
```

For å finne a_{50} og a_{150} må vi da definere om a_n:

```
>> n=1:150;
>> a_n=log(n)./n;
>> a_n(50)
ans = 0.0782

>> a_n(150)
ans = 0.0334
```

Legg merke til at `a_n` er en vektor, og 50. element av denne vektoren er `a_n(50)`.

Vi ser at verdiene blir mindre og mindre og lager en gjetting at grenseverdien er null.

Man kan bruke l'Hôpitals regel for å finne grenseverdien:

$$\lim_{n\to\infty} \frac{\ln n}{n} = \lim_{n\to\infty} \frac{1/n}{1} = 0.$$

En annen mulighet er å bruke funksjonen `limit`:

```
>> limit(log(x)/x,x,inf)

ans = 0
```

Man kan også plotte grafen av følgens allmenne ledd a_n som en funksjon av n for å se om den har noen endelig grenseverdi.

```
>> x = 1:500;
>> plot(log(x)/x, 'LineWidth', 2);
>> grid on;
```

I Figur 4 ser vi at a_n nærmer seg 0 når n øker.

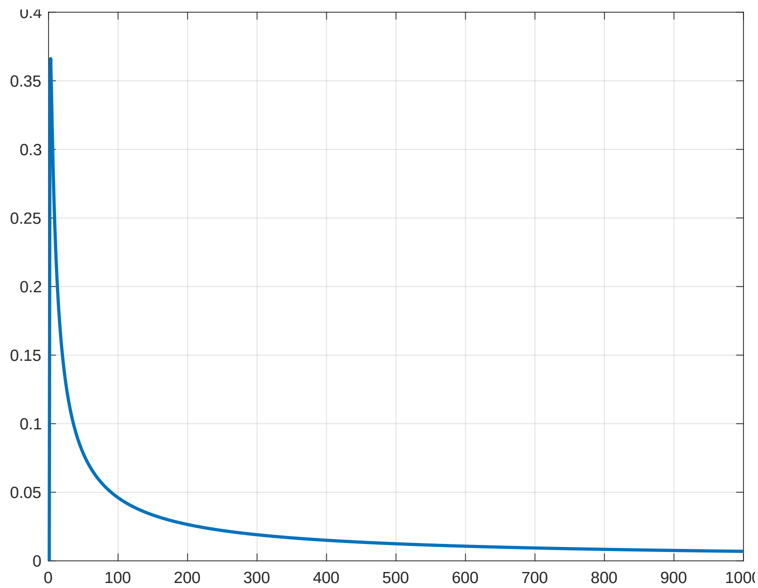

FIGURE 3. Grafen av $a_n = \ln(n)/n$.

□

Vi fortsetter med rekker, og nå skal vi se hvordan man kan bruke MATLAB til å sjekke om en rekke konvergerer eller divergerer og til å visualisere delsummene.

Eksempel 1.5. Bruk MATLAB til å bestemme om rekken

$$\frac{1}{1\cdot 2}+\frac{1}{2\cdot 3}+\frac{1}{3\cdot 4}+\cdots=\sum_{n=1}^{\infty}\frac{1}{n(n+1)}$$

konvergerer eller divergerer. Dersom rekken konvergerer, finn summen.

Løsning. Vi skal sjekke om delsummene har en grenseverdi, som er da summen av rekka i følge definisjonen. Delsummen S_k for den gitte rekka er

$$S_k=\sum_{n=1}^{k}\frac{1}{n(n+1)}.$$

Vi sier at rekka konvergerer dersom delsummene konvergerer (som en tallfølge) når $k \to \infty$. Delsummen S_k finner vi med MATLAB ved hjelp av `symsum`, mens grenseverdien til S_k når $k \to \infty$ kan vi regne ut ved hjelp av `limit`:

```
>> syms n
>> limit(symsum(1/(n*(n+1)), n, 1, x), x, inf)

ans = 1
```

Grenseverdien eksisterer og er lik 1, det vil si at rekka konvergerer og summen er lik 1.

En annen mulighet skulle bli å bruke, for eksempel, forholdstesten. Vi skal se på grenseverdien til a_{n+1}/a_n, der $a_n = 1/n(n+1)$ når $n \to \infty$. Dersom grenseverdien mindre enn 1, konvergerer rekka. I vårt tilfelle

$$\frac{a_{n+1}}{a_n} = \frac{n(n+1)}{(n+1)(n+2)} = \frac{n}{n+2} \to 1, \quad n \to \infty,$$

derfor gir ikke forholdstesten noe svar, og man må bruke andre metoder (for eksempel sammenligningstesten eller integralmetoden).

□

Oppgaver til Kapittel 1

1.1. Lag en MATLAB-kode som regner ut summen

$$\sum_{n=-3}^{1} (-1)^n \frac{3}{n^2+1}$$

og bruk den for å kontrollere svaret i opp. 2.6.2f.

1.2. Gitt funksjonen $f(x) = \ln x$, regn ut de første fem deriverte for hånd eller ved hjelp av MATLAB symbolic. Tegn grafene til funksjonen og taylorpolynomene av orden 1 til 5 om punktet $x = 1$.

1.3. La $f(x) = \cos(x/2)$. Sett opp likningene til taylorpolynom P av grad 1 og 2 med

$$P^{(k)}(1) = f^{(k)}(1), \quad k = 1, 2, 3.$$

Tegn grafene til funskjonen og de to polynomene i ett og samme koordinatsystem.

1.4. Skriv ut de første ti leddene av tallfølgen

$$a_n = \frac{3n^2 + n - 7}{-n^2 + 2}.$$

Konvergerer tallfølgen når $n \to \infty$? Kan du gjette grenseverdien? Regn ut grenseverdien analytisk eller ved hjelp av MATLAB symbolsk og sammenlikn svaret.

1.5. Bruk MATLAB til å gjette grenseverdien av tallfølgen $a_n = -n^2 + 2n - 1$ når $n \to \infty$. Regn ut grenseverdien ved hjelp av `limit`.

1.6. Bestem om rekka

$$\sum_{n=1}^{\infty} \frac{1}{3^{n+1}}$$

konvergerer eller divergerer. Dersom rekken konvergerer finn summen.

1.7. Bestem om rekka

$$\sum_{n=1}^{\infty} \frac{1}{2n+1}$$

konvergerer eller divergerer. Plot delsummene $S_k = \sum_{n=1}^{k} 1/(2n+1)$ som en funskjon av k. Ser det ut som at delsummene konvergerer?

1.8. Bestem om rekka

$$\sum_{n=1}^{\infty} \frac{(-1)^n}{2n+1}$$

konvergerer eller divergerer. Plot delsummene $S_k = \sum_{n=1}^{k} (-1)^n/(2n+1)$ vs k. Ser det ut som at delsummene konvergerer? Dersom rekka konvergerer, finn summen.

2. Laplacetransformasjon

For en stykkevis kontinuerlig funksjon f for $t \geq 0$ er laplacetransformasjonen definert ved

$$\mathcal{L}(f) = F(s) = \int_0^\infty f(t)\, e^{-st}\, dt$$

dersom integralet konvergerer.

Det å regne ut laplacetransformasjon med MATLAB er ganske lett: man kan bruke den innebygde funksjonen `laplace`.

Eksempel 2.1. Finn laplacetransformasjonen til $f(t) = t\, e^{2t}$.

Løsning.
```
>> syms t s;
>> f = t*exp(2*t);
>> laplace(f,t,s)

ans = 1/(s - 2)^2
```

Legg merke til at `t` i `laplace(f,t,s)` er variabelen i selv integralet i laplacetransformasjonen, mens `s` er transformasjonsvariabelen.

For å få et "finere" svar, kan vi bruke `simplify` and `pretty`.

I vårt tilfelle får vi

```
>> pretty(ans)
    1
--------
       2
(s - 2)
```

□

Invers laplacetransformasjonen gis ved `ilaplace`. La oss bruke den for å få den opprinnelige funksjonen $f(t) = te^{2t}$ i Eksempel 2.1.

```
>> syms t s;
>> F = 1/(s-2)^2;
>> ilaplace(F,s,t)

ans = t*exp(2*t)
```

For å beskrive funksjoner som er definert stykkevis kan vi bruke `heaviside` som returnerer 0 for $x < 0$, 1 for $x > 0$, and 1/2 for $x = 0$.

For eksempel, funksjonen

$$f(t) = \begin{cases} f_1, & t < t_1, \\ f_2, & t_1 < t < t_2, \\ f_3, & t > t_2 \end{cases}$$

kan vi skrive om på formen

$$f(t) = f_1(t) + u(t - t_1)(f_2(t) - f_1(t)) + u(t - t_2)(f_3(t) - f_2(t)),$$

(2)
$$u(t-a) = \begin{cases} 0, & t < a, \\ 1, & t \geq a. \end{cases}$$

Eksempel 2.2. Finn laplacetransformasjonen til
$$f(t) = \begin{cases} 3, & 0 \leq t \leq 2, \\ 1, & t \geq 2. \end{cases}$$

Løsning. Vi begynner ved å skrive om funksjonen ved hjelp av enhetssprangfunksjonen $u(t)$ definert ved (2).
$$f(t) = 3u(t) + (1-3)u(t-2) = 3u(t) - 2u(t-2).$$
La oss plotte funksjonen slik at vi ser at denne omskrivningen er riktig.

```
>> syms t;
>> f=3*heaviside(t)+(1-3)*heaviside(t-2);
>> fplot(f,[-1,4],'LineWidth',2);
>> grid on;
>> legend('show');
```

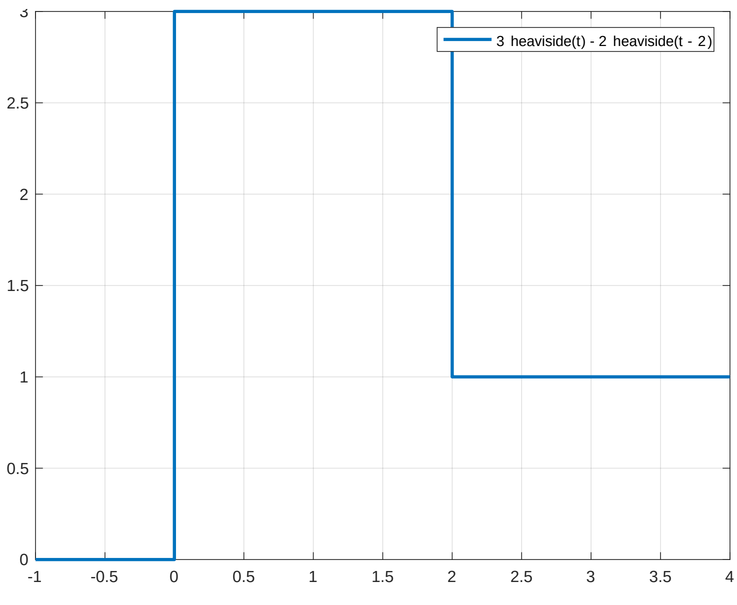

FIGURE 4. Grafen til $f(t) = 3u(t) - 2u(t-2)$.

Nå gjør vi laplacetransformasjonen, analytisk og i MATLAB:
$$F(s) = \int_0^\infty f(t)e^{-st}dt = \int_0^2 3e^{-st}dt + \int_2^\infty e^{-st}dt$$
$$= -\frac{3}{s}e^{-st}\Big|_{t=0}^{t=2} - \lim_{R\to\infty} \frac{1}{s}e^{-st}\Big|_{t=2}^{t=R} = \frac{3}{s}(1-e^{-2s}) + \frac{1}{s}e^{-2s} = \frac{3}{s} - \frac{2e^{-2s}}{s}.$$

```
>> laplace(f,t,s)

ans = 3/s - (2*exp(-2*s))/s
```

$\square$

Eksempel 2.3. Løs initialverdiproblemet ved å bruke laplacetransformasjon både analytisk og ved hjelp av MATLAB

$$y'' - 4y = 12e^t, \quad y(0) = 4, \; y'(0) = 0. \tag{3}$$

Løsning. Vi begynner ved å løse det analytisk. Første steg er å anvende laplacetransformen til begge sider av likningen. La $Y(s)$ være laplacetrasformasjonen til $y(t)$. Laplacetransformasjonen til annenderivert er gitt ved

$$\mathcal{L}[y''] = s^2Y - sy(0) - y'(0) = s^2Y - 4s.$$

Etter anvendelse av $\mathcal{L}$ blir likningen omformet til

$$s^2Y - 4s - 4Y = \frac{12}{s-1}.$$

Her har vi brukt at $\mathcal{L}[e^t] = 1/(s-1)$.

Den siste likningen løses med hensyn på Y:

$$Y(s) = \frac{4s^2 - 4s + 12}{(s-1)(s^2-4)}.$$

Vi bruker delbrøkoppspalting og får

$$Y(s) = \frac{5}{s-2} - \frac{4}{s-1} + \frac{3}{s+2}.$$

Invers laplacetransformasjonen $\mathcal{L}^{-1}[n!/(s-a)^{n+1}] = t^n e^{at}$ gir

$$\boxed{y(t) = 5e^{2t} - 4e^t + 3te^{-2t}.}$$

Nå skal vi løse likningen (3) ved hjelp av MATLAB.

```
% Solve y"-4y=12exp(t), y(0)=4, y'(0)=0,
% by using Laplace transform
close all; clear all;
% Symbolic variables
syms t s Y;
% Define the right-hand side
f = 12*exp(t);
% Laplace transform of f
F = laplace(f,t,s);
% Laplace transform of y": s^2*Y-s*y(0)-y'(0)
Y2 = s^2*Y-4*s;
% Solve Y2 -4Y-F=0 with respect to Y
Solution = solve(Y2-4*Y-F,Y)
% Make invers Laplace transform
solution = ilaplace(Solution,s,t)
% Plot the solution
```

```
fplot(solution,[0,5],'LineWidth',2);
grid on; legend('show');
```

Output:

```
Solution = (4*s + 12/(s - 1))/(s^2 - 4)

solution = 3*exp(-2*t) + 5*exp(2*t) - 4*exp(t)
```

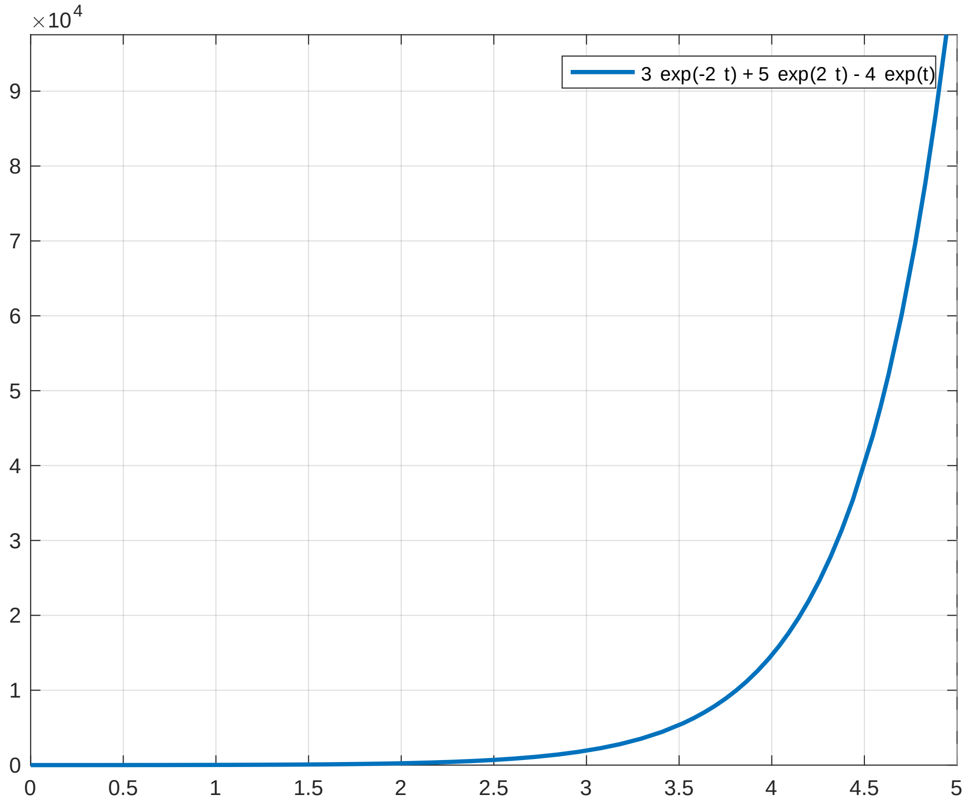

FIGURE 5. Løsningen til $y'' - 4y = 12e^t$ med $y(0) = 4, y'(0) = 0$.

□

OPPGAVER TIL KAPITTEL 2

2.1. Bruk definisjonen (integrer) og MATLAB til å finne laplacetransformen til

$$f(t) = \begin{cases} 2, & 3 \leq t \leq 5, \\ 4, & 0 \leq t < 3, \ t \geq 5. \end{cases}$$

Sammenlikn resultatene.

2.2. Finn laplacetransformen til funksjonen

$$f(t) = -\cos(3t) + 4\sin(\sqrt{2}t), \quad t \geq 0.$$

2.3. Finn funksjonen f gitt ved $f(t) = \mathcal{L}^{-1}[F(s)]$, $t \geq 0$, der
$$F(s) = \frac{se^{-\pi s}}{2s^2+1}.$$

2.4. Løs initialverdiproblemet ved å bruke laplacetransformasjon både analytisk og ved hjelp av MATLAB
$$y'' - 2y' + 5y = 0, \quad y(0) = 1,\ y'(0) = 0.$$

2.5. Løs initialverdiproblemet
$$y'' - y = f(t),\ y(0) = 1,\ y'(0) = 3,$$
der
$$f(t) = \begin{cases} 0, & 0 \leq t \leq \pi,\ t \geq 2\pi, \\ 4\sin t, & \pi \leq t < 2\pi. \end{cases}$$

3. Fourier rekker

Eksempel 3.1. Lag en MATLAB-kode som regner ut summen $\sum_{n=1}^{4} n(n+2)$.

Løsning. Vi setter først summen lik null og etterpå regner ut summen i en løkke med hensyn på n, fra $n = 1$ til $n = 4$. Ta bort semikolonet etter i linja 7 om du vil skrive ut delsummene.

```
1  % Compute the sum \sum_{n=1}^{4} n(n+2)
2  close all; clear all;
3  % Initial value for S
4  S = 0;
5  % Compute the sum in a loop
6  for n=1:4
7      S = S + n*(n+2);
8  end
9  % Print the answer
10 fprintf('Sum = %d\n', S);
```

Output:

```
>> Sum = 50
```

□

4. Fasit

Kapitel 1

1.1. Lag en MATLAB-kode som regner ut summen $\sum_{n=-3}^{1}(-1)^n\frac{3}{n^2+1}$ og bruk den for å kontrollere svaret i opp. 2.6.2f.

Løsning.

```
% Compute the sum \sum_{n=-3}^{1} 3*(-1)^n/(n^2+1)
close all; clear all;
% Initial value for S
S = 0;
% Compute the sum in a loop
for n=-3:1
    S = S + 3*(-1)^n/(n^2+1);
end
% Print the answer
fprintf('Sum = %f\n', S);
```

```
>> Sum = 0.300000
```

1.2. Gitt funksjonen $f(x) = \ln x$, regn ut de første fem deriverte for hånd og sammenlikn svaret med det du får ved hjelp av MATLAB symbolic.

Løsning.
Vi deriverer vi funksjonen fem ganger symbolsk:

```
>> syms x
>> diff(log(x))
ans = 1/x

>> diff(log(x),2)
ans = -1/x^2

>> diff(log(x),3)
ans = 2/x^3

>> diff(log(x),4)
ans = -6/x^4

>> diff(log(x),5)
ans = 24/x^5
```

□

1.3. Tegn grafene til funksjonen og taylorpolynomene av orden 1 til 5 om punktet $x = 1$.

Løsning.

```
% y=ln(x) and Taylor polynomials
close all; clear all;
% Define a symbolic variabel x
syms x
% Define the function and Taylor polynomials
% of order from 1 to 5
f = log(x);
T1 = taylor(f,x,'ExpansionPoint',1,'Order',2)
T2 = taylor(f,x,'ExpansionPoint',1,'Order',3)
T3 = taylor(f,x,'ExpansionPoint',1,'Order',4)
T4 = taylor(f,x,'ExpansionPoint',1,'Order',5)
T5 = taylor(f,x,'ExpansionPoint',1,'Order',6)
% Plot the functions
fplot([f, T1, T2, T3, T4, T5],[0, 2], 'LineWidth',2);
legend('show', 'Location', 'best');
grid on;
```

Figur 6 viser grafene til funksjonen og taylorpolynomene av grad fra 1 til 5.

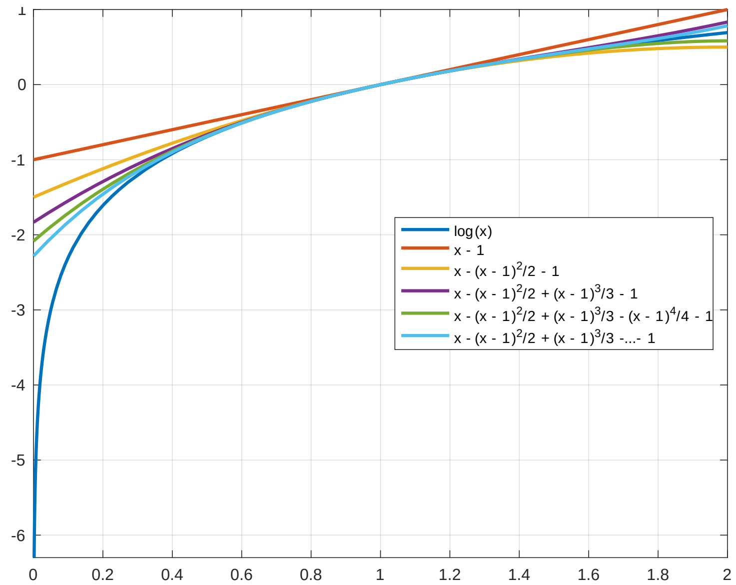

FIGURE 6. $f(x) = \ln x$ og de første fem taylorpolynomene i $x_0 = 1$.

□

1.4. La $f(x) = \cos(x/2)$. Sett opp likningene til polynomene P av grad mindre eller lik n med

$$P^{(k)}(0) = f^{(k)}(0), \quad k = 0, 1, \cdots, n$$

for $n = 2$. Tegn grafene til funskjonen og de tre polynomene i ett og samme koordinatsystem.

Løsning. Det er snakk om Taylor polynomer, og vi skal bruke samme strategy som før.

```
% y=cos(x/2) and Taylor polynomials
```

```
close all; clear all;
% Define a symbolic variable x
syms x
% Define the function and the polynomials
% of order from 1 to 5
f = cos(x/2);
T0 = taylor(f,x,'Order',1)
T1 = taylor(f,x,'Order',2)
T2 = taylor(f,x,'Order',3)
fplot([f, T0, T1, T2],[-1, 1], 'LineWidth',2);
legend('show', 'Location', 'best');
grid on;
```

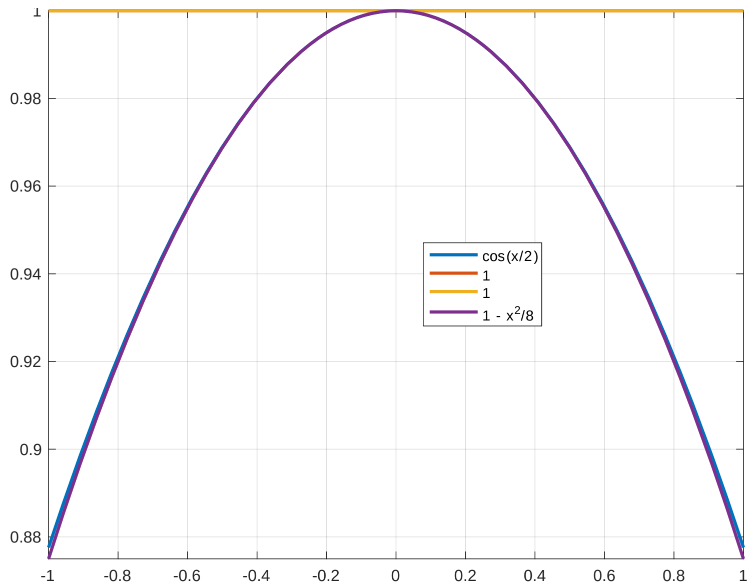

FIGURE 7. $f(x) = \cos(x/2)$ og polynomer av grad 0, 1 og 2 som approximerer $f(x)$ i $x = 0$.

```
T0 = 1

T1 = 1

T2 = 1 - x^2/8
```

□

1.5. Bruk MATLAB til å skrive ut de første ti leddene av tallfølgen

$$a_n = \frac{3n^2 + n - 7}{-n^2 + 2}.$$

Konvergerer tallfølgen når $n \to \infty$? Kan du gjette grenseverdien?

Løsning.
```
>> n=1:10;
>> a_n = (3*n.^2+n-7)./(-n.^2+2)

a_n =
```

```
  Columns 1 through 5

   -3.0000   -3.5000   -3.2857   -3.2143   -3.1739

  Columns 6 through 10

  -3.1471   -3.1277   -3.1129    -3.1013    -3.0918
```

Tar vi større n, vil vi se at a_n konvergerer mot -3. Det kan man regne ut analytisk:

$$\lim_{n\to\infty} \frac{3n^2+n-7}{-n^2+2} = \lim_{n\to\infty} \frac{3+\frac{1}{n}-\frac{7}{n^2}}{-1+\frac{2}{n^2}} = -3.$$

```
>> limit((3*x^2 + x-7)/(-x^2+2), x, inf)

ans = -3
```

□

1.6. Bruk MATLAB til å gjette grenseverdien av tallfølgen $a_n = -n^2 + 2n - 1$ når $n \to \infty$.

Løsning.
```
>> n=1:100;
>> a_n = -n.^2+2.*n-1;
>> a_n(50)
ans =  -2401

>> a_n(80)
ans = -6241

>> a_n(100)
ans = -9801
```

Vi ser at grenseverdien er $-\infty$, som er bekreftet av MATLAB:

```
>> limit(-x^2+2*x-1, x, inf)

ans = -Inf
```

□

1.7. Bruk MATLAB til å bestemme om rekka

$$\sum_{n=1}^{\infty} \frac{1}{3^{n+1}}$$

konvergerer eller divergerer. Dersom rekken konvergerer finn summen.

Løsning. Vi kan bruke definisjonen og sjekke om delsummene $S_k = \sum_{n=1}^{k} 1/3^{n+1}$ har en grenseverdi:

```
>> syms x,n
>> limit(symsum(1/(3^(n+1)), n, 1, x), x, inf)

ans = 1/6
```

Man kan plotte delsummen som en funksjon av k:

```
>> fplot(Sk, [1, 10], 'LineWidth', 2);
>> legend('show');
```

Resultatet vises i Figur 8. Man ser at delsummen nærmer seg grenseverdien $1/6$ når $k \to \infty$.

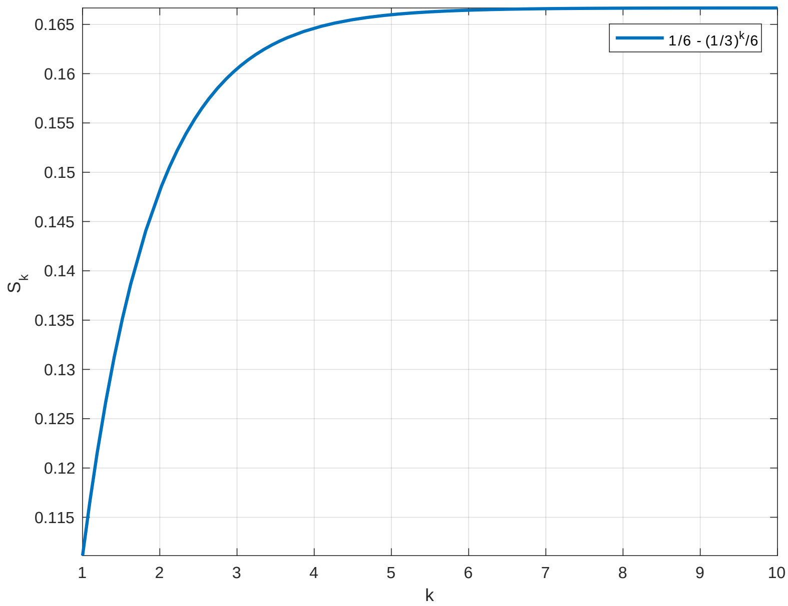

FIGURE 8. Delsummen $S_k = \sum_1^k 1/3^{n+1}$ som en funksjon av k.

Så vi ser at delsummene konvergerer, derfor er den opprinnelige rekka også konvergerer og summen er lik $1/6$. Det kan vi bekrefte analytisk ved å regne ut summen av en geometrisk rekke:

$$\sum_{n=1}^{\infty} \frac{1}{3^{n+1}} = \frac{1}{3^2} \cdot \frac{1}{1 - 1/3} = \frac{1}{9} \cdot \frac{3}{2} = \frac{1}{6}.$$

Man kan også bruke forholdstesten (rekka er positiv). Fordi det allmenne leddet $a_n = 1/3^{n+1}$, skal vi regne ut grenseverdien

$$\lim_{n\to\infty} \frac{a_{n+1}}{a_n} = \lim_{n\to\infty} \frac{3^{n+1}}{3^{n+2}} = \frac{1}{3}.$$

Fordi grenseverdien er mindre enn 1 konvergerer rekka. Samme grenseverdien kan man regne ut ved hjelp av MATLAB:

```
>> limit(3^(n+1)/3^(n+2),n,inf)

ans = 1/3
```

Summen av rekka:

```
>> syms n
>> symsum(1/3^(n+1),n,1,inf)
```

```
ans = 1/6
```

□

1.8. Bruk MATLAB til å bestemme om rekka

$$\sum_{n=1}^{\infty} \frac{1}{2n+1}$$

konvergerer eller divergerer. Plot delsummene $S_k = \sum_{n=1}^{k} 1/(2n+1)$ vs k. Ser det ut som at delsummene konvergerer?

Løsning.
```
>> syms n k;
>> Sk = symsum(1/(2*n+1),n,1,k);
>> fplot(Sk, [1, 100],'LineWidth',2);
>> grid on;
>> xlabel('k'); ylabel('S_k');
```

Som vi ser i Figur 9 konvergerer ikke delsummene til noe endelig verdi, derfor kan vi konkludere at rekka divergerer.

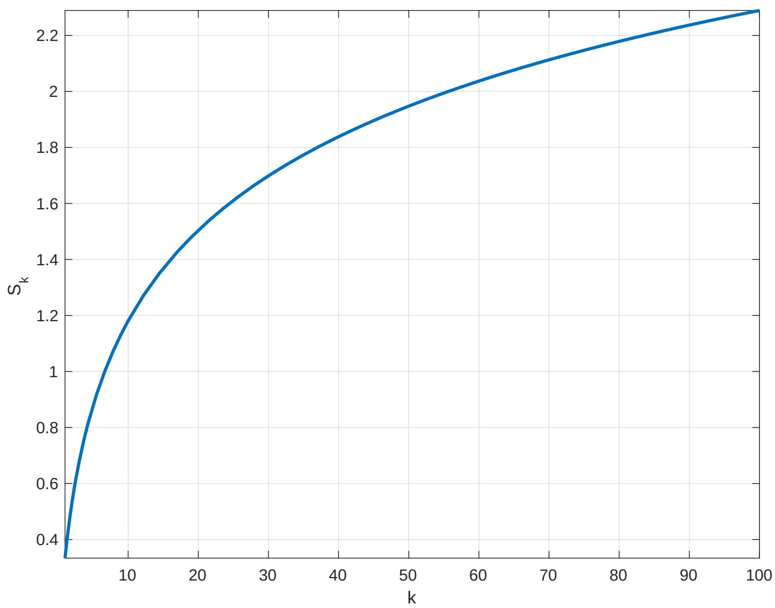

FIGURE 9. Delsummen $S_k = \sum_{1}^{k} 1/(2n+1)$ som en funksjon av k.

□

1.9. Bruk MATLAB til å bestemme om rekka

$$\sum_{n=1}^{\infty} \frac{(-1)^n}{2n+1}$$

konvergerer eller divergerer. Plot delsummene $S_k = \sum_{n=1}^{k} (-1)^n/(2n+1)$ vs k. Ser det ut som at delsummene konvergerer? Dersom rekka konvergerer, finn summen.

Løsning. Denne gangen skal vi bruke en løkke for å regne ut delsummene. Det kunne vi selvklart gjøre i de forrige eksemplene slik at vi ikke er avhengige av de innebygde funksjonene **sum** og **symsum**.

```
% Plot partial sums of \sum_1^infty (-1)^n/(2n+1)
close all; clear all;;
% Number of points in the partial su
N = 100;
% Allocating a vector for partial sums
S = zeros(N,1);
% Computing partial sums
for k=1:N
    for n=1:k
        S(k) = S(k)+(-1)^n/n;
    end
end
% Vector k for the plot
k=1:N;
% Plot partial sums
plot(k, S, 'LineWidth', 2);
grid on;
xlabel('k'); ylabel('S_k');
```

Som vi ser i Figur 10 konvergerer delsummene og, som følge, rekka. På grunn av at rekka er alternerende, er delsummen som en funskjon av k ikke monoton - grafen går opp og ned siden leddene endrer fortegn. Men totalt sett, siden det allmenne leddet går mot null når $n \to \infty$, konvergerer rekka. Sammenlikn gjerne grafen i Figur 10 med den i Figur 9, der delsummene blir større og større selv om det allmenne leddet går mot null når $n \to \infty$.

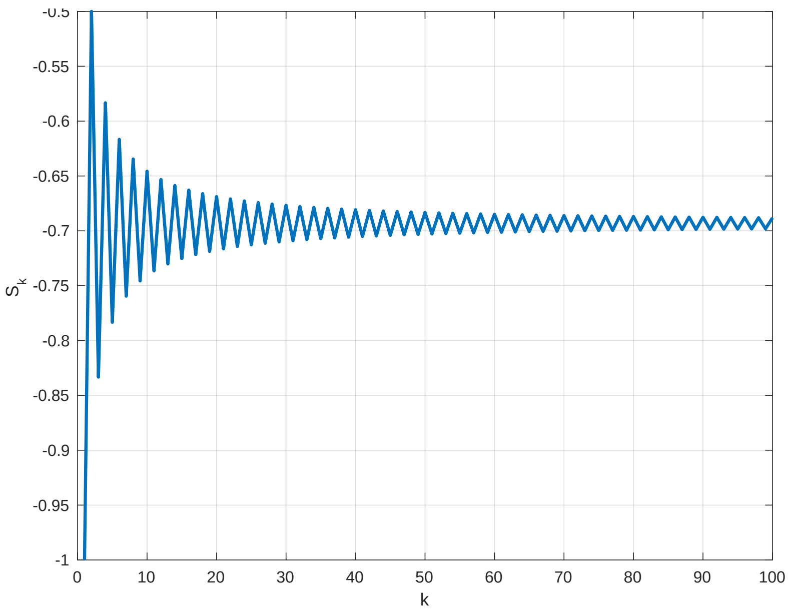

FIGURE 10. Delsummen $S_k = \sum_{n=1}^{k} \frac{(-1)^n}{(2n+1)}$ som en funksjon av k.

□

Kapitel 2

2.1. Bruk definisjonen (integrer) og MATLAB til å finne laplacetransformen til

$$f(t) = \begin{cases} 2, & 3 \le t \le 5, \\ 4, & 0 \le t < 3, \ t \ge 5. \end{cases}$$

Sammenlikn resultatene.

Løsning. Ved å bruke definisjonen:

$$F(s) = \int_0^\infty f(t)e^{-st}dt = \int_0^3 4e^{-st}dt + \int_3^5 2e^{-st}dt + \int_5^\infty 4e^{-st}dt$$
$$= \frac{4}{s} - \frac{2}{s}e^{-3s} + \frac{2}{s}e^{-5s}.$$

Ved hjelp av MATLAB:

```
>> syms t s;
>> f=4*heaviside(t)+(2-4)*heaviside(t-3)+4*heaviside(t-5);
>> laplace(f,t,s)

ans = (4*exp(-5*s))/s - (2*exp(-3*s))/s + 4/s
```

□

2.2. Finn laplacetransformen til funksjonen

$$f(t) = -\cos(3t) + 4\sin(\sqrt{2}t), \quad t \ge 0.$$

Løsning.

```
>> laplace(-cos(3*t)+4*sin(sqrt(2)*t))

ans = (4*2^(1/2))/(s^2 + 2) - s/(s^2 + 9)
```

□

2.3. Finn funksjonen f gitt ved $f(t) = \mathcal{L}^{-1}[F(s)]$, $t \ge 0$, der

$$F(s) = \frac{se^{-\pi s}}{2s^2+1}.$$

Løsning.

```
>> ilaplace(s*exp(-pi*s)/(2*s^2+1))

ans = (cos((2^(1/2)*(t - pi))/2)*heaviside(t - pi))/2
```

□

2.4. Løs initialverdiproblemet ved å bruke laplacetransformasjon både analytisk og ved hjelp av MATLAB

(4) $$y'' - 2y' + 5y = 0, \quad y(0) = 1, \ y'(0) = 0.$$

Løsning.
Vi begynner med analytisk løsning. La $Y = \mathcal{L}[y]$. Laplacetransformen til de første to deriverte er

$$\mathcal{L}[y'] = sY - y(0), \quad \mathcal{L}[y''] = s^2Y - sy(0) - y'(0).$$

La oss anvende laplacetransformasjonen på begge sider av likningen (4):

$$(s^2Y - s) - 2(sY - 1) + 5Y = 0 \quad \Rightarrow Y = \frac{s-2}{s^2 - 2s + 5} = \frac{s-2}{(s-1)^2 + 4}$$

Den siste brøken kan man skrive om på formen

$$\boxed{Y = \frac{s-1}{(s-1)^2 + 2^2} - \frac{1}{(s-1)^2 + 2^2}}.$$

Invers laplacetransfromasjonen til Y gir

$$\boxed{y(t) = e^t \cos(2t) - \frac{1}{2}e^t \sin(2t).}$$

Ved hjelp av MATLAB:

```
% Solve y"-2y'+5y=0, y(0)=1, y'(0)=0,
% by Laplace transform
close all; clear all;
% Symbolic variables
syms t s Y;
% Laplace transform of y': s*Y -y(0)
Y1 = s*Y - 1;
% Laplace transform of y": s^2*Y-s*y(0)-y'(0)
Y2 = s^2*Y-s;
% Solve Y2-2Y1+5Y=0 with respect to Y
Solution = solve(Y2-2*Y1+5*Y,Y)
% Make the inverse Laplace transform
solution = ilaplace(Solution,s,t)
% Plot the solution
fplot(solution,[0,5],'LineWidth',2);
grid on; legend('show');
```

Output:

```
Solution =(s - 2)/(s^2 - 2*s + 5)


solution =exp(t)*(cos(2*t) - sin(2*t)/2)
```

□

2.5. Løs initialverdiproblemet

$$y'' - y = f(t), \; y(0) = 1, \; y'(0) = 3,$$

der

$$f(t) = \begin{cases} 0, & 0 \leq t \leq \pi, \; t \geq 2\pi, \\ 4\sin t, & \pi \leq t < 2\pi. \end{cases} \tag{5}$$

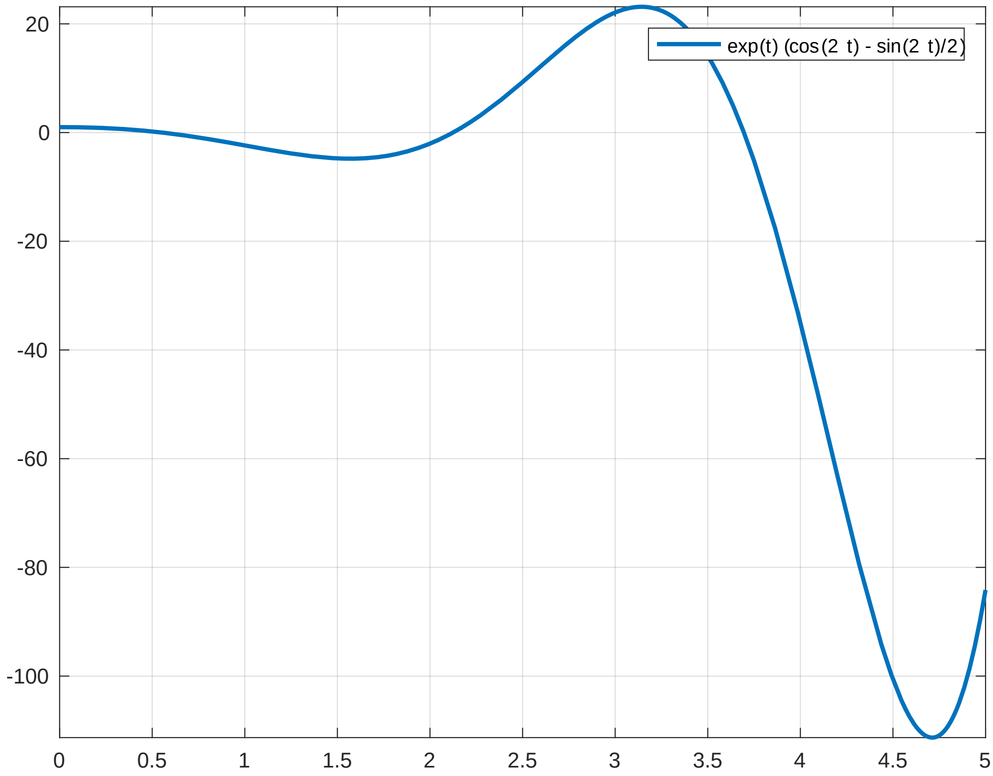

FIGURE 11. Løsningen til $y'' - 2y' + 5y = 0$ med $y(0) = 1, y'(0) = 0$.

Løsning.

```
% Solve y"-y=f(t), y(0)=1, y'(0)=3,
% f=0 for 1<t<=pi and t>=2pi
% f=4sin t for pi<=t>2pi
% by using Laplace transform
close all; clear all;
% Symbolic variables
syms t s Y;
% Define the right-hand side
f = 4*sin(t)*heaviside(t-pi)-4*sin(t)*heaviside(t-2*pi);
% Laplace transform of f
F = laplace(f,t,s);
% Laplace transform of y': s*Y-y(0)
Y1 = s*Y-1;
% Laplace transform of y": s^2*Y-s*y(0)-y'(0)
Y2 = s^2*Y-s-3;
% Solve Y2-Y-F=0 with respect to Y
Solution = solve(Y2-Y-F,Y)
% Make invers Laplace transform
solution = ilaplace(Solution,s,t)
% Plot the solution
```

```
fplot(solution,[0,10],'LineWidth',2);
grid on; legend('show');
```

Output:

```
Solution =
 (s - (4*exp(-pi*s))/(s^2 + 1)
 - (4*exp(-2*pi*s))/(s^2 + 1) + 3)/(s^2 - 1)


solution =
 2*exp(t) - exp(-t) - 4*heaviside(t - pi)*(sin(t)/2
 + exp(t - pi)/4 - exp(pi - t)/4)
 + 4*heaviside(t - 2*pi)*(exp(2*pi - t)/4 + sin(t)/2
 - exp(t - 2*pi)/4)
```

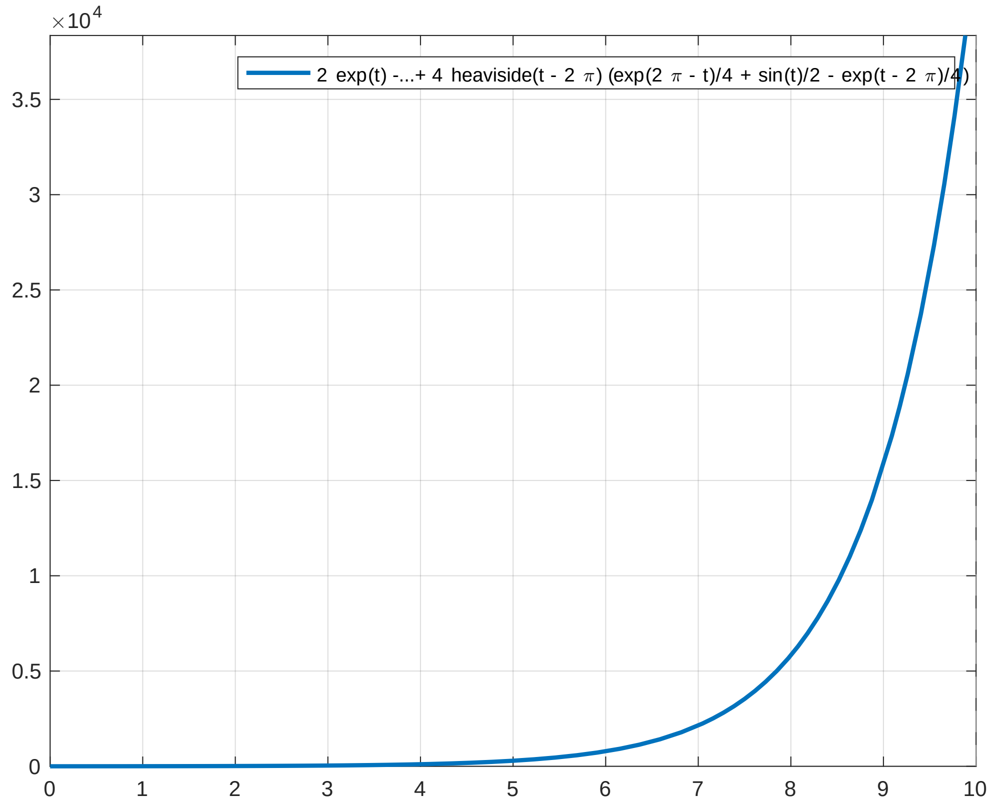

FIGURE 12. Løsningen til $y'' - 4y = f(t)$ med $y(0) = 1, y'(0) = 3$ og f definert i (5).

□