

Masteroppgave

Visualisering av funksjonsfølger og -rekker

av

Eirik Sørstrand

Masteroppgaven er gjennomført som et ledd i utdanningen ved Universitetet i Agder og er godkjent som sådan. Denne godkjenningen innebærer ikke at universitetet inntår for de metoder som er anvendt og de konklusjoner som er trukket.

Veileder:

Olav Nygaard

Universitetet i Agder, Kristiansand

Juni 2009

Forord

Jeg vil først og fremst takke min veileder Olav Nygaard for konstruktive innspill og tilbakemeldinger underveis i arbeidet mitt og for at døren alltid var åpen.

Arbeidet har medført mye frustrasjon og glede som har gitt mange opp- og nedturer. Noen løsninger medførte nye problemer, men totalt førte disse løsningene til et bedre sluttprodukt. Arbeidsmengden har til tider vært stor, men likevel overkommelig. Sett i en helhetlig sammenheng har det vært mange spennende og positive utfordringer.

Arbeidsmiljøet på rommet med de andre mastergradstudentene har vært trivelig og atmosfæren har vært en god motivasjonsfaktor under arbeidet. Tusen takk til mine medstudenter for en trivelig tid.

Til sist, men ikke minst, vil jeg takke min kone, Sara, som har støttet og oppmuntret meg med oppgaven når utfordringene var store, dagene var lange og kvelden satte inn. Takk for tålmodigheten din når jeg har levd i min egen verden foran pc-skjermen og vandret i mine egne tanker for å løse utfordringene som har stått foran meg.

Nå returnerer jeg tilbake til deg og samfunnet.

1.6.2009.

Universitetet i Agder

Institutt for matematiske fag, 2009.

Eirik Sørstrand

Sammendrag

Denne oppgaven handler om å visualisere funksjonsfølger og funksjonsrekkeutviklinger.

Først er det en innledning om hvordan Kalkuluskurset MA-100 ved Universitetet i Agder introduserer rekker for studenter og hva slags utfordringer jeg selv møtte i forbindelse med rekkekonseptet. Deretter kommer en del om hva som oppleves vanskelig med grenseverdibegrepet og en definisjon av punktvis- og uniform konvergens.

Hoveddelen av denne oppgaven er et egenprodusert JAVA dataprogram i form av Applet slik at det kan publiseres på Internett. Det blir forklart hvordan selve designet er bygget opp samt utfordringer under utviklingen. Hele kildekoden ligger som vedlegg.

Tilslutt kommer det 3 eksempler på hvordan programmet kan brukes i en undervisningssammenheng.

Eksempelene gjelder

- Punktvis konvergens.
- Uniform konvergens.
- En rekkeutvikling for å vise konvergens mot en kjent elementærfunksjon.

Universitetet i Agder
Institutt for matematiske fag, 2009.

Eirik Sørstrand

Abstract

This thesis is about visualizing sequences and series of functions.

First comes an introduction on how the course in Calculus, MA-100 at the University of Agder, introduces the concept of series to the students and the challenges I met in this subject. Then comes a part where I discuss difficulties in connection with the limit concept together with a definition of pointwise and uniform convergence.

The main part of this thesis is a self produced JAVA computer program in form of an Applet to be public on the Internet. The design will be explained and challenges during the development are discussed. The source code is given in the Appendix.

Finally 3 examples will be given on how the program can be used in an educational context.

The examples concern

- Pointwise convergence.
- Uniform convergence.
- The development of a series converging to a known elementary function.

University of Agder
Department of Mathematics, 2009.

Eirik Sørstrand

Innholdsfortegnelse

1	Møte med rekker i Kalkulus	1
2	Litteraturbakgrunn.....	5
2.1	Læren om uendelighetsbegrepet	5
2.2	Funksjonsfølge.....	6
2.3	Punktvis konvergens.....	6
2.4	Uniform konvergens.....	7
2.5	Innføring av nye funksjoner på lavere nivå	7
3	Layout og design av programmet	9
3.1	Utdropstegnet ved avslutt knappen.....	9
3.2	Koordinatsystemet	9
3.3	Zoom inn ved hjelp av klikk, dra og slipp.....	10
3.4	Navnet til programmerer.....	10
3.5	Innstillingsmuligheter	11
3.5.1	Zoom inn x, Zoom ut x, Zoom inn y og Zoom ut y.....	11
3.5.2	Slå av/på rutenett	11
3.5.3	Fjerne/tilføy talle.....	11
3.5.4	Still vindu	12
3.5.5	Ferdige $f(x)$	13
3.5.6	Hjelp	14
3.5.7	N-verdier	14
3.5.7.1	Funksjonsfølger	14
3.5.7.2	Rekker.....	15
3.5.8	Fjern F_n ?.....	15
3.5.8.1	Funksjonsfølger	15
3.5.8.2	Rekker.....	16
3.5.9	Nullstill.....	16
3.6	Funksjons- og rekkeutviklingsfeltet.....	16
3.6.1	Tegn graf / Øk verdien med * / Start på nytt	17
3.7	Gjett grensefunksjon	17
3.8	Ulovlige verdier i tekstfelt	18
3.9	PI og det irrasjonelle tallet e	18
3.10	Informasjonsfelt	18

4	Utviklingsfasen	21
4.1	Programvaren	21
4.2	Java utfordringer	21
4.2.1	Programmeringsspråket JAVA.....	21
4.2.2	Panelløsningene	22
4.2.3	Plassering av origo.....	22
4.2.4	Analysering av teksten	22
4.2.5	Skjulte tegn.....	23
4.2.6	Automatisk innlesning fra fil	23
4.3	Kalkulasjonsutfordringer	24
4.3.1	Rekursive rutiner (Rutiner kaller opp seg selv)	24
4.3.2	Potenser	24
4.3.3	Uendelig og ulovlige verdier	24
4.3.4	Standardform	25
4.3.5	Hacking ved mange n – ledd	26
4.3.6	Fakultet utregninger.....	26
4.4	Forbedringsmuligheter til programmet.....	26
5	Eksempler på bruk av programmet	27
5.1	Eksempel 1: Punktvis konvergens	27
5.2	Eksempel 2: Uniform konvergens.....	29
5.3	Eksempel 3: Visualisere en rekkeutvikling som konvergerer	34
6	Pedagogiske implikasjoner.....	43
7	Referanser.....	45
8	Vedlegg: Kildekode.....	47
8.1	Like klasser i programkoden til funksjonsfølger og rekkeutvikling.	47
8.1.1	Klassen: Advarsel.....	47
8.1.2	Klassen: Navnepanel	48
8.1.3	Klassen: KnappPanel	49
8.1.4	Klassen: FerdigeFunksjoner.....	53
8.1.5	Klassen: HjelpVindu.....	55
8.1.6	Klassen: InnstillingsVindu	57
8.1.7	Klassen: DelingsVindu	61
8.1.8	Klassen: Hovedvindu	63
8.1.9	Klassen: GjettGrenseFunksjonsVindu	65

8.2	Funksjonsfølger	68
8.2.1	Klassen: FjernFnVindu	68
8.2.2	Klassen: TegneVindu	71
8.2.3	Klassen: StillNVerdiVindu	106
8.2.4	Klassen: UtviklingPanel	109
8.2.5	Klassen: funksjonsfølger.....	130
8.3	Rekkeutviklingen	133
8.3.1	Klassen: FjernFnVindu	133
8.3.2	Klassen: StillNVerdiVindu	135
8.3.3	Klassen: TegneVindu	137
8.3.4	Klassen: UtviklingPanel	138
8.3.5	Klassen: rekkeutvikling.....	138

1 Møte med rekker i Kalkulus

Kalkuluskurset MA-100 ved Universitetet i Agder inneholder et kapittel om funksjonsrekker, spesielt potensrekker. Emnet om funksjonsrekker skapte mye frustrasjon for meg, siden jeg ikke hadde tidligere kunnskap om emnet. Her måtte jeg bygge opp et helt nytt område innenfor matematikken, og kapitlet med funksjonsrekker var det siste som ble undervist før eksamen. Dette ga dårlig tid til å fordøye emnet.

Utdrag fra boken "Kalkulus - med én og flere variable"

(Lorentzen, Hole, & Lindstrøm, 2003):

Vi skal innføre et hav av nye funksjoner. Med disse funksjonene i bakhånd kan vi i prinsippet finne alle antideriverte av våre funksjoner. Funksjonene ser ut som "polynomer av uendelig grad", og det er jo praktisk, for polynomer er enkle både å derivere og integrere. Det er bare en hake: et "polynom av uendelig grad" inneholder uendelig mange ledd $a_k x^k$ som må adderes.

Å addere n tall er enkelt. Men hvordan vil du addere uendelig mange tall?

I dette kapitlet skal vi se hvordan vi kan gjøre det i praksis, både eksakt og tilnærmet.

Vi kaller en slik uendelig lang addisjon for en rekke.

(Lorentzen, et al., 2003, p. 305)

Som vi ser her utdyper boken "Kalkulus - med én og flere variable" behovet for å innføre rekker. På neste side utdyper de nytteverdiene for å innføre rekker:

Approksimere tall og funksjoner

- *Skrive elementære funksjoner på en måte som ofte gjør det enklere å arbeide med dem, spesielt om de inngår i integraler eller differensiallikninger*
- *Beskrive enkelte fysiske fenomener*
- *Behandle diskrete sannsynlighetsfordelinger*

(Lorentzen, et al., 2003, p. 306)

For å få en oppbygning til selve rekkekonseptet starter boken med følger for så å bevege seg inn på emneområdet rekker.

Problemstillingen jeg ønsker å gripe tak i er å gjøre selve rekkeutviklingen mer visuelt.

Boken "Kalkulus - med én og flere variable" illustrerer ved hjelp av kaker på side 313, og de skriver:

Eksempel 7.2.1. Summer rekken

$$\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots + \frac{1}{2^n} = \sum_{n=1}^{\infty} \frac{1}{2^n}$$

Løsning: For å være konkrete, tenker vi oss en bløtekake.

Du forsyner deg med halvparten. Da har du tatt $\frac{1}{2}$ kake. Nestemann som kommer, forsyner seg med halvparten av dette. Det vil si, hun tar $\frac{1}{4}$ kake. Da er det totalt sett tatt $(\frac{1}{2} + \frac{1}{4})$ kake, og det er $\frac{1}{4}$ kake igjen på fatet. Nestemann tar halvparten av dette igjen, altså $\frac{1}{8}$ kake.

Da er det totalt sett tatt $(\frac{1}{2} + \frac{1}{4} + \frac{1}{8})$ kake, og det er $\frac{1}{8}$ kake igjen på fatet. Se figur 7.2.1.

Fortsetter vi denne prosessen i det uendelige, er det tatt $\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots + \frac{1}{2^n} = \sum_{n=1}^{\infty} \frac{1}{2^n}$ kake og fatet er tomt. Dermed bør summen av rekken være lik 1.

Vi kan skrive dette matematisk.

La S_N betegne hvor stor andel av kaken som er tatt etter N gangers forsyning. Da er

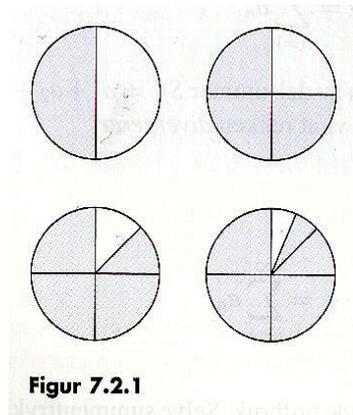
$$S_1 = \frac{1}{2} = 1 - \frac{1}{2}$$

$$S_2 = \frac{1}{2} + \frac{1}{4} = 1 - \frac{1}{4}$$

$$S_3 = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} = 1 - \frac{1}{8}$$

...

$$S_N = \frac{1}{2} + \frac{1}{4} + \dots + \frac{1}{2^N} = \sum_{n=1}^N \frac{1}{2^n} = 1 - \frac{1}{2^N}$$



Figur 1: Kakeillustrasjonen figur 7.2.1 fra (Lorentzen, et al., 2003, p 313)

Som vi ser vokser S_N når N øker, og S_N blir mer og mer lik 1. (Lorentzen, et al., 2003, p. 313)

Mye av kapitlet med rekker blir veldig lite visuelt og mesteparten er algebraiske kalkulasjoner. Jeg husker selv at jeg bare godtok det boken skrev uten selv å kunne se utviklingen i mine egne tanker og refleksjoner. Selve regneprosessen tok litt tid å bli vant med, men når jeg først fikk "forståelsen" var det enklere algebraisk, men fremdeles vanskelig å se visuelt.

Hvordan kan vi se at rekkeutviklingen $\sum_{n=0}^{\infty} \frac{x^n}{n!}$ er helt identisk med e^x ?

Jeg klarer ennå ikke den dag i dag å se dette i mine tanker eller refleksjon, men har bare valgt å godta det. Dermed kan man diskutere om den algebraiske regnemethoden er en automatisk prosess uten fullstendig forståelse, eller om studenten går på "autopilot" om man vil bruke et slikt uttrykk.

Det er her programmet kommer inn for å illustrere funksjonsfølger og rekkeutviklinger. Et av hovedfokusene med rekkekonseptet er å avgjøre når rekken konvergerer eller divergerer. Programmet er tiltenkt å kunne visualisere funksjonsfølger eller rekkeutviklinger for enkelt å kunne gi et bilde av om de konvergerer eller divergerer.

Videre skriver boken:

Følgende Taylorrekker om $a = 0$ (Maclaurinrekker) konvergerer mot funksjonen de representerer i det gitte intervallet. (Lorentzen, et al., 2003, p. 322)

Jeg velger her bare å bruke funksjonen e^x siden boken oppgir dette som et konkret eksempel på side 323.

$$B. \quad e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots = \sum_{n=0}^{\infty} \frac{x^n}{n!} \text{ for alle } x.$$

Ved første øyekast ser dette lite lurt ut. Vi har skrevet våre kjente, kjære elementærfunksjoner på en skikkelig komplisert måte, nemlig som rekker.

Men det er faktisk kjempesmart.

...

Setter vi for eksempel $x=1$ i eksemplet over får vi at

$$B. \quad e^x = 1 + \frac{1}{1!} + \frac{1^2}{2!} + \frac{1^3}{3!} + \dots = \sum_{n=0}^{\infty} \frac{1}{n!} = e \text{ (Legg merke til at det står likhet)}$$

Tallet e er helt nøyaktig lik summen av rekken, en rekke som er lett å skrive opp siden vi vet hvordan alle leddene ser ut. Det er mer enn vi kan si om desimaltallet for det irrasjonale tallet e . (Lorentzen, et al., 2003, p. 323)

Jeg vil bruke programmet til å vise at $\sum_{n=0}^{\infty} \frac{x^n}{n!}$ faktisk blir e^x under **punkt 5.3 Eksempel 3:**

Visualisere en rekkeutvikling som konvergerer.

2 Litteraturbakgrunn

2.1 Læren om uendelighetsbegrepet

Mange elever og studenter har vanskeligheter med å lære å se for seg uendeligheten. Siden vi lever i en verden som er endelig og mesteparten av all matematisk regning og matematiske problemer har vært veldig ”konkret”, blir overgangen veldig stor når emner plutselig har uendelighetsbegrepet inkludert. Det er vanskelig å tenke at noe aldri stopper, det bare fortsetter og fortsetter og fortsetter og ...

Singer og Voica (2008) viser til forskning gjort av Fischbein, Tirosh, & Hess (1979) der det skrives at elever ser på uendelig som en egenskap i en prosess i stedet for et nummerliknende objekt som har en størrelsesorden. Det blir påpekt misoppfattelser ved bruk av reproduisering med en endelig regnemåte for å løse problemer med grenseverdier og konvergensområder.

Hvorfor er det så vanskelig å tenke seg at noe er uendelig?

Fischbein (1979) mener at noen grunner for dette vanskelige konseptet er våre forbindelser med tv-serier som har reist i rom og tid. Uendelighetskonseptet er ofte forbundet med en selvforklarende erkjennelse av at det aldri stopper. Elever og studenter (personer generelt) lever i en verden som er endelig. Vi ”vet” hvor stor jorda er, vi vet hvor langt det er fra Oslo til Kristiansand og de fleste av oss har begreper om endelige konsepter. Uendelig blir derfor vanskelig å konseptualisere eller diskutere.

Jirotková og Littler (2003) påpeker at elever og studenter ofte bruker uendelig og uendelighet om hverandre og ikke vet helt forskjellen mellom ordene. Vanskeligheten for elevene og studentene blir å skille mellom en uendelig prosess som ender med grenseverdi eller at grenseverdien er et resultat av et uendelig forløp.

Mange matematikkonsepter og begreper er avhengig av grenseverdi, som for eksempel derivert, integraler, kontinuitet av funksjoner samt sum av uendelige rekker. Elever og studenter møter tidlig slike eksempler, kanskje uten å være klar over det.

La meg vise et slikt eksempel:

Er 0,99999999... lik 1 eller nesten lik 1? Intuitivt vil nok mange svare ”nesten lik 1”.

Men ser vi på $\frac{1}{3}$ så er ”alle” enige om at det er lik 0,333333333....

$$\frac{1}{3} = 0.333333333...$$

ganger vi så med 3 på begge sider får vi

$$1 = 0.999999999...$$

Så hvor mye er 0,99999999...?

Ser vi videre på dette tallet, kan det skrives

$$0 \cdot 10^0 + 9 \cdot 10^{-1} + 9 \cdot 10^{-2} + 9 \cdot 10^{-3} + 9 \cdot 10^{-4} + 9 \cdot 10^{-5} + 9 \cdot 10^{-6} + \dots$$

som igjen kan skrives som rekken $\sum_{n=1}^{\infty} 9 \cdot 10^{-n}$.

Dette er en geometrisk rekke som konvergerer til verdien 1.

Dersom uendelighetsbegrepet blir sterkere forklart for elever og studenter kan det hjelpe de matematisk på mange områder.

En av vanskelighetene med uendelighetsbegrepet er at det er vanskelig å visualisere. Det er store pedagogiske fordeler å få muligheten til å utforske og analysere utviklingen visuelt i stedet for ved abstrakte tanker.

Det er her programmet skal hjelpe oss til å visualisere rekkeutvikling og til å forstå begrepene punktvis konvergens og uniform konvergens, som blir beskrevet under punktene **2.3**

Punktvis konvergens og **2.4 Uniform konvergens**.

Roh (2008) skriver at forskning gjort av Ferrini-Mundy and Lauten (1993); Tall (1992) understreker at konseptet med grenseverdi er en av de mest fundamentale ideer, ikke bare for å forstå Kalkulus, men også for å utvikle en matematisk tenkning utenfor Kalkulus og forfølgelsen av matematikk på et dypere nivå.

For å få en bedre forståelse av grenseverdibegrepet er dynamisk visning en positiv angrepsvinkel. Dette blir støttet opp med forskning gjort av Kidron og Zehavi (2002) som fant at det er stor nytteeffekt med å bruke dynamiske illustrasjoner i læren om grenseverdier. Pinto og Tall (2002) avslørte at visuelle bilder har en positiv rolle i læring og forståelse av reell analyse. Har vi en visuell instruksjonsmetode kan det hjelpe studenter til å bevege seg mot en mer korrekt matematisk definisjon mener Navarro og Carreras (2006).

De fleste forelesere ønsker ikke å gjennomføre et kurs i reel analyse uten bruk av visuelle bildefremstillinger etter Alcock og Simpson (2004). Det blir her påpekt at det er både negative og positive sider med bruk av bilder. Men konsentrerer vi oss om de positive sidene og bruker en visuell tilnærming, kan vi knytte sterkere bånd mellom det visuelle bildet og den matematiske definisjonen, som også samsvarer med Navarro og Carreras (2006).

2.2 Funksjonsfølge

En følge er en indeksert liste av objekter som enten er av endelig eller tellbar uendelighet. Dermed blir en funksjonsfølge en indeksert liste av funksjoner med endelig eller tellbar uendelig lengde.

2.3 Punktvis konvergens

Hva skal til for at en funksjonsfølge konvergerer?

Punktvis konvergens defineres slik:

La (f_n) være ei følge av reelle funksjoner, der hver f_n er definert på samme mengde A .

Hvis vi for hver $x \in A$ har at tallfølga $(f_n(x))$ konvergerer, definerer vi den punktvis grensa $g = \lim f_n$ til funksjonsfølga (f_n) ved at $g(x) = \lim_n f_n(x)$. (Nygaard, 2007, p. 78)

g blir nå en funksjon med definisjonsmengde A .

Idéen med programmet jeg har utviklet er at man enkelt skal kunne tegne opp grafene til $f_n(x)$ til funksjonsfølgene eller delsummene til en rekkeutvikling. I en undervisningssituasjon kan det ta lang tid å tegne opp en funksjons- eller rekkeutvikling dersom man må tegne opp mange grafer før selve konvergensområdet blir tydelig. Mange kalkulatorer mister også koordinatsystemet når det skal legges til en ny graf, men dette programmet beholder alltid koordinatsystemet synlig. Pedagogisk lønner det seg å beholde koordinatsystemets visuelle effekt når man analyserer.

2.4 Uniform konvergens

Hva skal til for at vi får uniform konvergens?

Definisjonen på uniform konvergens:

La (f_n) være ei følge av funksjoner definert på samme mengde A .

(f_n) sies å konvergere uniformt til funksjonen g dersom det for alle $\varepsilon > 0$ finnes $N \in \mathbb{N}$

slik at når $n > N$ så er $d_\infty(f_n, g) < \varepsilon$, der d_∞ er metrikken definert ved

$d_\infty(\phi, \psi) = \sup_{x \in A} |\phi(x) - \psi(x)|$ (Nygaard, 2007, p. 79)

Dette betyr intuitivt at det finnes en x der konvergensens av $(f_n(x))$ ”går tregest”.

Her er idéen til programmet de samme som punktene beskrevet under punkt **2.3 Punktvis konvergens**.

2.5 Innføring av nye funksjoner på lavere nivå

Kunnskapsløftet sier om de matematiske fagene:

R1:

- *gjøre rede for begrepene grenseverdi, kontinuitet og deriverbarhet, og gi eksempler på funksjoner som ikke er kontinuerte eller deriverbare*

R2:

- *summere endelige rekker med og uten digitale hjelpemidler, utlede og bruke formelene for summen av de n første leddene i aritmetiske og geometriske rekker, og bruke dette til å løse praktiske problemer*
- *regne med uendelige geometriske rekker med konstante og variable kvotienter, bestemme konvergensområdet for disse rekkene og presentere resultatene (Utdanningsdirektoratet)*

Som vi ser skal elevene allerede i R1 bli kjent med begrepet grenseverdi.

Læreplanmålene i R2 påpeker at elevene skal bestemme konvergensområdet til rekker og presentere resultatene. Ved å bruke dette programmet får de en mulighet til å fremstille rekkene på en fin og oversiktlig måte. Når konvergensområdet skal bestemmes, kan det være en stor hjelp å ha en visuell fremstilling av rekken når analysen skal finne sted.

Elevene får her altså muligheten til å lære om rekker, som medfører at ”verktøykassen med funksjoner” blir større, og mer potensielle matematiske muligheter byr seg frem.

3 Layout og design av programmet

3.1 Utdropstegnet ved avslutt knappen



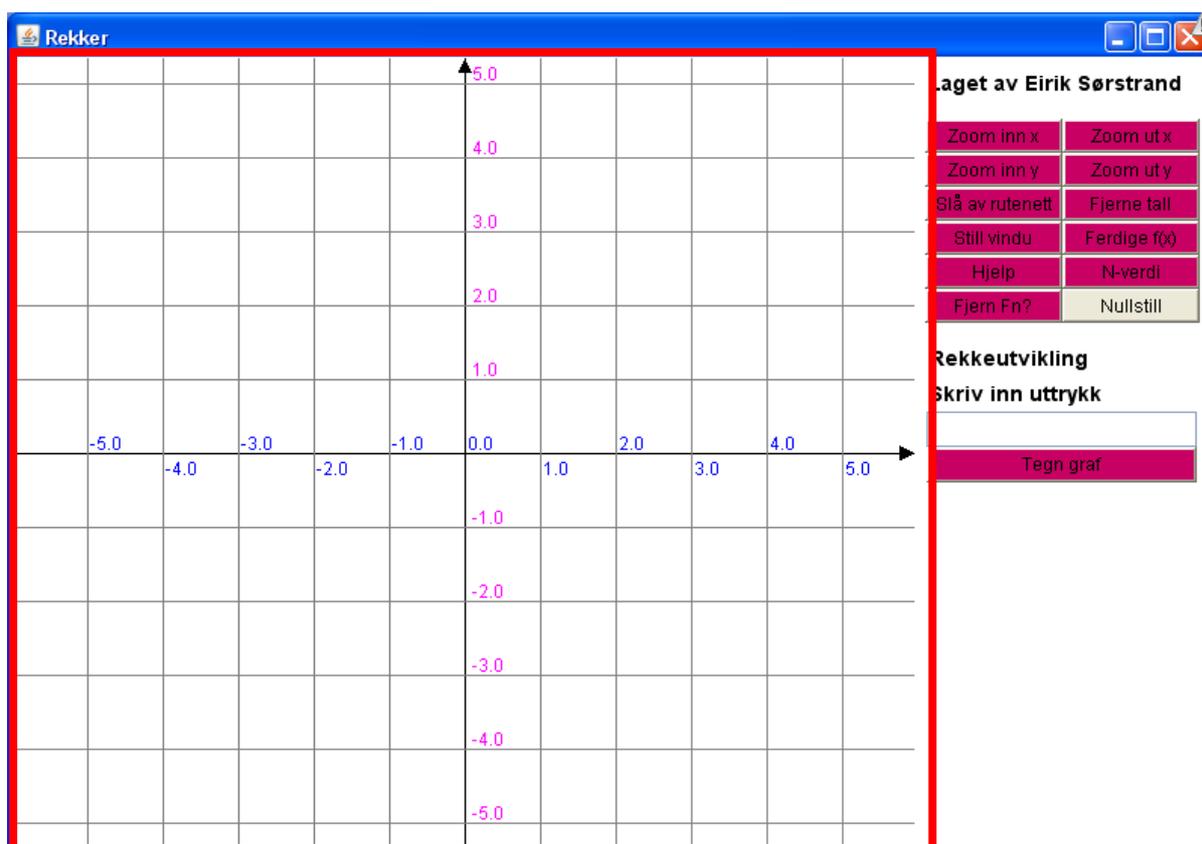
Figur 2:
Utdropstegn over
avslutt knappen

Utdropstegnet som blir liggende litt over avslutt knappen er en konsekvens av at programmet er en Applet som bruker en funksjon i Java som heter Frame. Utdropstegnet er en advarsel om at Appleten har startet et nytt vindu. Dette er ikke farlig, men bare en liten påminnelse om at det er et ”sprett opp vindu”.

3.2 Koordinatsystemet

Koordinatsystemet (se markering med rød ramme på figur 3) ble prioritert slik at det skulle få en størst mulig utstrekning, slik at det skal bli enkelt å bruke det i en undervisningssammenheng. Det er nøytral bakgrunn for ikke å stjele fokus fra analysen som skal forekomme i læreprosessen.

Som standard er origo satt i midten av selve koordinatsystemvinduet.



Figur 3: Bilde fra programdelen som viser rekkeutvikling

Hvis man skalerer vinduet vil origo automatisk plassere seg i forhold til den nye størrelsen av vinduet. Dette ble gjort for å få brukeren til å oppleve minimalt med forandringer av funksjonene som er tegnet. Grafene blir tegnet inn med trippel strek for å gjøre det lettere for

brukere å observere den i koordinatsystemet. Samtidig vil grafene ha forskjellig farge som samsvarer med fargen på uttrykket som kommer opp i det store hvite feltet. Dette feltet kaller jeg for utviklingsfeltet og blir beskrevet under punkt **3.6 Funksjons- og rekkeutviklingsfeltet**.

Det var to muligheter til å tegne opp grafene. Enten å la brukeren ha kontroll over når neste graf skulle tegnes eller å ha et automatisk tidsintervall. Valget ble at brukeren selv bestemmer, for i en undervisningssammenheng er ikke automatisk tidsintervall alltid å foretrekke. Det kan nemlig oppstå spørsmål eller andre hendelser som må diskuteres eller forklares. Dermed kan det være vanskelig å foreta en nødvendig analyse i det bestemte tidsintervallet før den neste grafen blir tegnet opp.

3.3 Zoome inn ved hjelp av klikk, dra og slipp

For å få full kontroll over det ønskede analyseområde har jeg programmert en funksjon som gjør det mulig å zoome med musen. Dersom man ønsker et nærmere fokus på et område, tar man musen og trykker ned med venstre musetast. Hold inne knappen og dra over det området du ønsker det skal zoomes på, området blir markert i koordinatsystemet. Når man slipper musetasten blir koordinatsystemet automatisk stilt inn til det ønskede området.

Koordinatsystemet blir transformert og skalert slik at origo blir balansert korrekt ut fra vektleggingen til det ønskede området. Her kan det oppstå tallverdier på aksene som ikke er heltall, siden man med stor sannsynlighet ikke treffer direkte på heltallsverdier. Men heltallsverdier kan stilles under innstillingsvalget ”Still vindu”, se punkt **3.5.4 Still vindu**. Det er også lagt inn en sperre slik at dersom du bare trykker uten å dra musen, blir det ikke foretatt en skalering eller transformering.

3.4 Navnet til programmerer

Mitt eget navn ble lagt øverst i vinduet for å få litt mellomrom ned til knappene.



Figur 4: Navnet til programmerer

3.5 Innstillingsmuligheter

For å få en fullstendig mulighet til å kontrollere området man vil analysere, har jeg lagt til en del innstillingsmuligheter.

Valgene er vurdert ut fra en standard kalkulator for å få et kjent brukergrensesnitt samt noen nye innstillinger spesielt egnet for bruksområdet til dette programmet.

3.5.1 Zoom inn x, Zoom ut x, Zoom inn y og Zoom ut y

Disse knappene gjør det mulig å skalere enten x - eller y -verdiene.

Først hadde jeg bare valget mellom å skalere i x og y retning, men jeg valgte fort å utvide til muligheten for å skalere i hver enkelt retning. Grunnen var at det ikke alltid er ønskelig å skalere i begge retninger samtidig. Skaleringen foregår med en faktor på 2 ved forstørring og en faktor på 0,5 ved forminsking.

Begge disse skaleringsmulighetene hjelper brukeren til å stille inn til ønsket verdi på aksene, som gir en bedre mulighet for analysere.

Dersom det er ønskelig å stille inn nøyaktige verdier se punkt **3.5.4 Still vindu**.

3.5.2 Slå av/på rutenett

Denne funksjonaliteten gir brukeren valget til å fjerne eller tilbakeføre rutenettet og ble implementert for at brukeren kan få større renhet i bilde og dermed fjerne bakgrunnsstøy. Et slikt valg har man også på kalkulator, men dette valget er ofte skjult i menyer som kan være vanskelig å finne.

"Slå på rutenett" kommer som en konsekvens av at man har valgt "Slå av rutenett".

Knappen vil bytte skrift, men stå på samme sted for ikke å forvirre brukeren.

3.5.3 Fjerne/tilføy tall

Denne funksjonaliteten gir brukeren valget mellom å fjerne eller tilbakeføre tallene på koordinataksene og ble implementert for at brukeren kan få større renhet i bilde og dermed fjerne bakgrunnsstøy. Men uten tallverdier kan analysen bli vanskeligere.

"Tilføy tall" kommer som en konsekvens av at man har valgt "Fjerne tall".

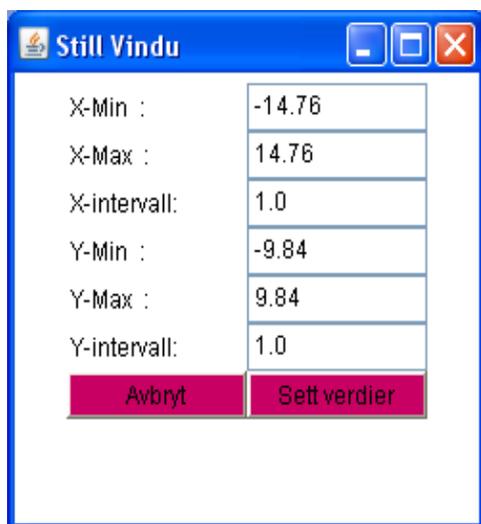
Knappen vil da bytte skrift, men står på samme sted for ikke å forvirre brukeren.

Tallene på x -aksen skapte problemer hvis tallene hadde for mange siffer og dermed kunne de overlappe hverandre. Løsningen jeg valgte var å sette annethvert tall over og under x -aksen. Y – verdiene står over hverandre og medførte ikke noe problem.

Dersom man har foretatt en skalering eller transformering kan tallene til x -verdiene ofte kollidere med y -verdiene. Løsningen på dette problemet er at tallene til x og y verdiene har en forskjellig farge. X - verdiene har fått en svak blålig farge, mens y - verdiene har fått en svak magenta farge.

3.5.4 Still vindu

Denne funksjonaliteten gir brukeren valget til selv å stille inn ønskede verdier på koordinatsystemet. Brukeren kan skrive inn verdier på:



X-Min :	-14.76
X-Max :	14.76
X-intervall:	1.0
Y-Min :	-9.84
Y-Max :	9.84
Y-intervall:	1.0

Avbryt Sett verdier

- *X*-min: den laveste verdien på *x*-aksen.
- *X*-max: den høyeste verdien på *x*-aksen.
- *X*-intervall: tallverdien mellom to rutenettlinjer på *x*-aksen.
- *Y*-min: den laveste verdien på *y*-aksen.
- *Y*-max: den høyeste verdien på *y*-aksen.
- *Y*-intervall: tallverdien mellom to rutenettlinjer på *y*-aksen.

Figur 5: Still verdiene til koordinatsystemet

Når verdiene er stilt inn kan bruker velge å akseptere innstillingene med å trykke på knappen "Sett verdier" eller velge å forkaste innstillingene med å trykke på knappen "Avbryt".

Dersom man aksepterer innstillingene blir koordinatsystemet skalert og transformert etter de angitte verdiene. Har brukeren skrevet logiske feil i verdiene, som for eksempel *X*-min verdien er større enn *X*-max verdien, kommer det opp en advarsel (se figur 6 og 7) som påpeker dette. Koordinatsystemet vil da ikke bli skalert eller transformert.



Advarsel!

X-max mindre enn X-min

OK

Avbryt

Figur 6: Advarsel når *X*-max er mindre enn *X*-min



Advarsel!

Y-max mindre enn Y-min

OK

Avbryt

Figur 7: Advarsel når *Y*-max er mindre enn *Y*-min

3.5.5 Ferdige f(x)

Her kan læreren legge inn ferdige uttrykk slik at brukeren kan velge mellom de forskjellige funksjonsfølgene eller rekkeuttrykkene som er lagt til. Da minsker man situasjoner hvor feil uttrykk blir skrevet inn som kan medføre at feil konklusjoner og/eller analyser kan oppstå.

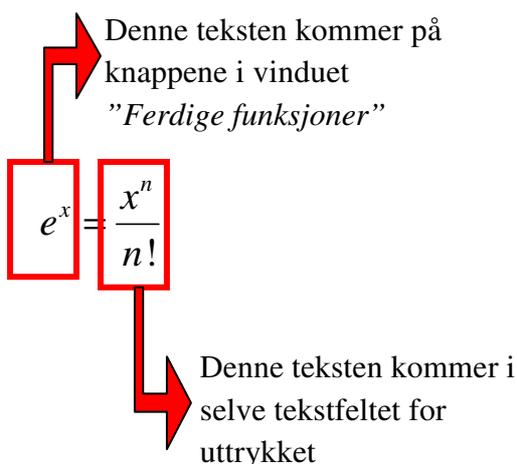
Denne funksjonaliteten er laget med en automatisk generering som medfører at programmet ikke trenger å kompiles hver gang nye uttrykk skal tilføyes.

Dette øker funksjonaliteten til programmet og gjør det lettere for lærere å ta det i bruk.

Filen som leses heter "ferdigeFunksjoner.txt" og er bygd opp på følgende metode:

La meg forklare med uttrykket $e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$

Måten man legger inn i filen "ferdigeFunksjoner.txt" er:



Figur 8: Uttrykk lagt til av læreren



Figur 9: Uttrykket blir automatisk overført til tekstfeltet

Merk at $\sum_{n=0}^{\infty}$ skrives ikke siden det er implementert i programkoden og brukeren velger selv startverdi på n, se punkt 3.5.7 N-Verdi.

Det som avgjør hvor den forskjellige tekst skal plasseres er altså '=' tegnet.

Teksten før likhetstegnet kommer på knappene i vinduet "Ferdig funksjoner" og teksten etter likhetstegnet blir plassert i uttrykksfeltet.

Merk at eksemplet over vil bli sendt slik ut i tekstform: $e^x = (x^n)/n!$

Det kan anbefales at det settes "Oppgave1" på navnet til knappen dersom man skal gjette en grensefunksjon, som vil bli vist under punkt **5.3 Eksempel 3: Visualisere en rekkeutvikling som konvergerer**, for ikke å avsløre grensefunksjonens identitet.

Dersom det inntreffer en feil ved innlesning av de ferdige funksjonene fra filen, kommer det opp en feilmelding med en forklarende tekst på hvorfor feilen oppstod.

Et eksempel her er hvis programmet ikke finner filen. Feilmeldingen som kommer opp er:
... **ferdigeFunksjoner.txt (Systemet finner ikke angitt fil)** eller
... **ferdigeFunksjoner.txt (The system cannot find the file specified).**

Teksten på feilmeldingen styres av operativsystemet og medfører at tilbakemeldingen kan være på engelsk.

3.5.6 Hjelp

Her kan læreren skrive inn en hjelpetekst slik at brukeren kan se hvordan programmet skal brukes eller hvordan oppgavene skal løses. Dersom det inntreffer en feil ved den automatiske innlesning, kommer det opp forklarende feilmelding til brukeren som ble forklart i punkt **3.5.5 Ferdige f(x)**.

Hjelpen leses fra filen "hjelp.txt" og er bygd opp på følgende måte:

Høyden på vinduet blir stilt automatisk etter hvor mange linjer det er, og bredden på vinduet blir stilt automatisk etter hvor mange tegn den lengste linjen inneholder.

All tekst har fått lik formatering, samme teksttype og størrelse, siden jeg har valgt å ikke prioritere tiden på denne funksjonaliteten.

3.5.7 N-verdier

Her kan brukeren stille startverdien til n slik at han/hun har kontrollen på funksjonsfølgene eller rekkeutviklingen. Det er jo slik at funksjons- eller rekkeutvikling ikke alltid starter med verdien $n = 0$ og derfor er det blitt implementert en mulighet for at brukeren kan velge startverdi på n . Her er det også et valg for å sette verdien til hvor mye n skal økes per steg, dersom man ønsker å se utviklingen til funksjonsfølgen eller legge til en ny delsum i rekkeutviklingen. Det er litt forskjellig utseende på de to programdelene, la meg forklare.

3.5.7.1 Funksjonsfølger

Her falt valget på å skrive inn egne verdier til n i form av tekst. Valget var veldig logisk siden vi kan ende opp med en veldig stor verdi på n når funksjonsfølger utvikler seg, og da ville rullefeltet være upraktisk.



Figur 10: Forandrer verdier til n under funksjonsfølger

3.5.7.2 Rekker

For rekker er det implementert et rullefelt slik at brukeren ikke skal skrive inn verdien på n . Herved utelukker programmet feilinnskrivinger av n verdien siden brukeren blir tvunget til å velge ferdige verdier. I rekkeutviklinger dannes ofte mønsteret fort samtidig som det ofte er lav endring i n verdien.

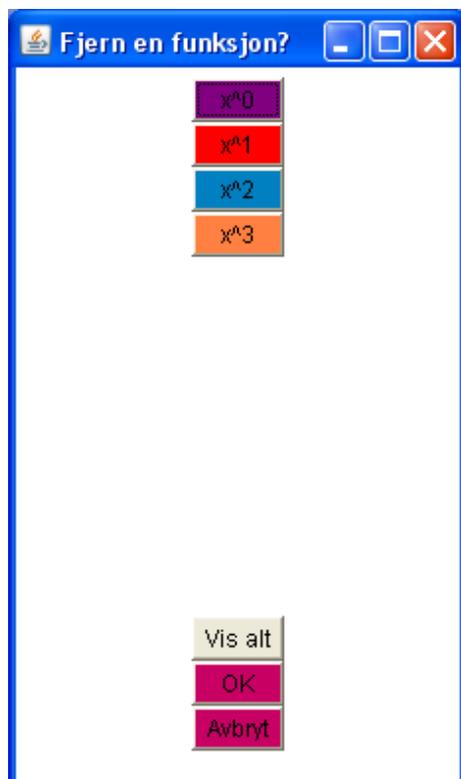


Figur 11: Forandrer verdier til n under rekkeutvikling

3.5.8 Fjern Fn?

Her har brukeren muligheten til å fjerne en eller flere funksjonsfølger eller rekkeutviklinger. Denne muligheten medfører at vi kan få en bedre mulighet til å analysere det ønskede området, uten bakgrunnsstøy. Det er to forskjellige funksjonaliteter på de to programdelene, la meg forklare.

3.5.8.1 Funksjonsfølger



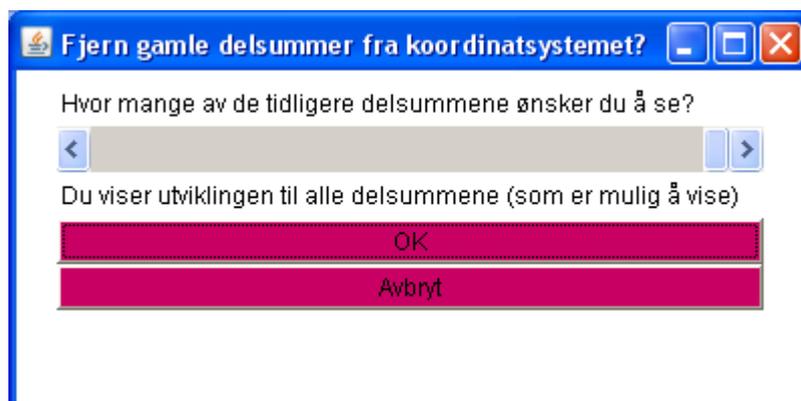
Her kan brukeren selv fjerne noen av de tidligere funksjonene i funksjonsfølgen. Trykker man på en av knappene vil den valgte funksjonen automatisk forsvinne fra koordinatsystemet. Dersom noen av de tidligere funksjonene er skjult kan brukeren velge "Vis alt" og dermed blir alle funksjonene synlige igjen.

Figur 12: Fjern noen av funksjonsfølgene

3.5.8.2 Rekker

I programdelen som visualiserer rekker har jeg implementert et rullefelt hvor brukeren avgjør hvor mange av de siste tidligere delsummene han/hun ønsker å fremstille i koordinatsystemet. Den siste delsummen blir alltid vist, men brukeren kan avgjøre selv om de ønsker å visualisere de to siste, de tre siste ... eller alle utviklingene.

Ved bruk av rullefeltet blir teksten (under rullefeltet) oppdatert med hvor mange av de tidligere visualiseringene som blir vist.



Figur 13: Fjern gamle visualiseringer til rekkeutviklingen

3.5.9 Nullstill

Nullstill er intuitivt slik at alt stilles tilbake til det opprinnelige designet som programmet starter opp med.

3.6 Funksjons- og rekkeutviklingsfeltet

I tekstfeltet kan man skrive inn ønsket funksjonsfølge eller rekkeutvikling.



Figur 14: Feltet brukeren skriver inn en funksjonsfølge eller et rekkeuttrykk



Figur 15: Følgen eller delsummene utvikler seg og får en fargekode som er lik den aktuelle visualiseringen i koordinatsystemet

Når man har trykket på knappen for å få tegnet en funksjon, får funksjonsuttrykket eller delsummen en bestemt farge i feltet under knappen "Tegn graf" (og påfølgende utskiftninger av teksten etterpå). Fargekoden blir den samme som funksjonsfølgen eller delsummen til rekkeuttrykket i koordinatsystemet, slik at brukeren skal kunne se hvem som tilhører hverandre.

Merk: I funksjonsfølger er uttrykkene individuelle, mens under rekkevisualiseringen blir uttrykkene addert.

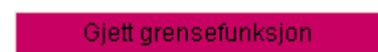
3.6.1 Tegn graf / Øk verdien med * / Start på nytt

Denne knappen medfører at funksjonsfølgen eller rekkeuttrykket blir tegnet opp i koordinatsystemet. Når du har trykket på "tegn graf" kommer teksten "Øk verdien med *", hvor * betyr økningen i verdien til n . Denne teksten blir automatisk oppdatert dersom brukeren velger å forandre n verdien. Teksten er dermed en indikator som gir brukeren kontroll over hvor stor økningen er på n .

Dersom brukeren har kommet til det maksimale antall funksjonsfølger eller delsummer i rekkeutviklingen, som er mulig å visualisere, blir teksten byttet fra "Øk verdien med *" til "Start på nytt". Det er dermed ikke mulighet til å tegne flere utviklinger grunnet plassmangel til nye uttrykk og det maksimale antallet er 25. Antallet som er mulig å vise kan variere og blir bestemt ut fra høyden på skjermen til brukeren.

3.7 Gjett grensefunksjon

Her har brukeren en mulighet til å gjette på grensefunksjonen til funksjonsfølgen eller rekkeuttrykket.



Figur 16: Brukeren kan velge å gjette på en grensefunksjon

Dersom brukeren velger å gjette på en grensefunksjon kommer følgende vindu opp.



Figur 17: Vinduet hvor brukeren kan gjette grensefunksjonen og sette en epsilon verdi

Her kan brukeren skrive inn sitt forslag til hvilken grensefunksjon funksjonsfølgen eller rekken konvergerer mot, dersom den konvergerer. Programmet tegner opp grensefunksjonen med fargen svart. Jeg vil vise et konkret eksempel på denne funksjonaliteten under punktet

5.3 Eksempel 3: Visualisere en rekkeutvikling som konvergerer.

Feltet "Sett epsilon verdi" er ment til å sette en minste avstand fra grensefunksjonen. Dersom brukeren skriver inn for eksempel 0,1 i dette feltet vil det bli tegnet opp to ekstra funksjoner til grensefunksjonen som vil ha verdier lik grensefunksjonen \pm epsilonverdien og de har fargekoden rød.

3.8 Ulovlige verdier i tekstfelt

Programmet har implementert, i de forskjellige delene hvor brukeren kan skrive inn tekst, en rutine for å prøve å unngå skrivefeil. Det sjekkes etter ulovlige verdier i den innskrevne teksten og dersom det finnes ulovlige tegn fjernes de. Ulovlige verdier er for eksempel bokstaver som r , ϕ , α , \hat{a} , q , w ..., men også ”, @, #, \Re , % og lignende tegn som ikke er ansett til å være i brukerområdet for dette programmet. Mellomrom blir fjernet, samt ”..” kombinasjonen blir erstattet med ”.” og ”,” blir erstattet med ”.” siden datamaskiner opererer med punktum som kommaoperator. Dette er implementert for å prøve å fjerne feilkilder.

3.9 PI og det irrasjonelle tallet e

Brukeren kan skrive inn PI eller pi, dersom han/hun ønsker den matematiske verdien for π . Det er også implementert mulighet for at brukeren kan skrive $3\pi/2$ og lignende. Denne funksjonaliteten er ikke case sensitiv. Det irrasjonelle tallet e blir erstattet med 2.718281828, men her må det merkes at funksjonaliteten er case sensitiv. Stor E er nemlig skrivemåte for standardform.

3.10 Informasjonsfelt

Informasjonsfeltet gir brukeren tilbakemelding underveis. De forskjellige tilbakemeldingene er:

<p>Informasjonsfelt</p> <p>Figur 18: Informasjonsfelt</p>	<p>Programmet starter med denne teksten slik at brukeren kan se at det er et informasjonsfelt.</p>
<p>Tegner opp graf</p> <p>Figur 19: Utregning skjer og grafen blir snart tegnet opp</p>	<p>Når brukeren har valgt å tegne opp en ny $f_n(x)$ eller legge til en ny delsum blir feltet oppdatert til ”Tegner opp graf”. Nå ser brukeren at datamaskinen jobber med å regne ut verdiene og gjør seg klar til å visualisere. Når utregningen er ferdig blir feltet helt hvitt.</p>
<p></p> <p>Figur 20: Ingen informasjon</p>	<p>Ingen informasjon på det nåværende tidspunkt.</p>
<p>Ekstremt høye verdier</p> <p>Figur 21: Noen utregninger ender i standardform</p>	<p>Utregningen til en x-verdi har en funksjonsverdi som er et ekstremt høyt tall. Noen (x, y) verdier blir dermed ikke tegnet opp. Grunnen for denne informasjonen er at programmet får vanskeligheter med å tegne opp verdien når de blir skrevet på standardform. Denne begrensningen er i grunnet datatypen <i>double</i> som har sine begrensninger og problemet ble løst med dette alternativet (Se også punkt 4.3.4 Standardform).</p>

Divisjon med null!

Figur 22: Programmet jukser litt med divisjon med 0

Her oppstår det en divisjon med 0, men programmet jukser dette til med å dividere med 0,00000001 i stedet for. Dermed kan det være feilmarginer rundt området som gir 0 i nevner. Denne muligheten ble innført for å gjøre det mulig å analysere for eksempel:

$$\lim_{x \rightarrow 0} \frac{\sin\left(\frac{x}{n}\right)}{\left(\frac{x}{n}\right)}$$

Fakultet over 100

Figur 23: Høy fakultet

Utrengningen prøver å ta fakultet over 100. Dette tallet er ekstremt høyt og programmet er satt til å avslutte fakultet utregningen og gale funksjonsverdier kan oppstå. Dermed er 100! det høyeste fakultetet som er mulig i dette programmet.

Negativ ln eller log

Figur 24: ln eller log til et tall som er 0 eller negativt

Utrengningen inneholder et uttrykk med ln eller log som er negativ eller 0. Denne utregningen er umulig, men likevel satt til gul fargekode. Skal man for eksempel vise $(\ln x)^n$ er denne funksjonen bare definert for $x > 0$, men koordinatsystemet kan være innstilt slik at man ser negative x -verdier og dermed vil programmet prøve å regne ut noen negative x -verdier. Disse negative x -verdiene blir dermed ikke regnet ut.

Feil med parenter

Figur 25: Logisk parentesfeil

Det har blitt implementert en rutine som gir brukeren tilbakemelding dersom det er en logisk feilskrivning i parentesbruk. Merk at den ikke sjekker om parentesene er brukt riktig matematisk, men bare om det er logisk korrekt oppsett.

To eksempler på logiske feil:

1. $x^n*(3/x)$ Logisk feil! Bare en (men to)
2. $x^{(n+1)}$ Logisk feil! Bare en (men ingen)

Ingen gyldig n

Figur 26: Ingen gyldig n verdi

Ingen gyldig n variabel ble funnet og dermed er uttrykket som er skrevet inn hverken en funksjonsfølge eller et rekkeuttrykk.

En uventet feil oppstod

Figur 27: En feil som ikke kan gjøres rede for

Her har det oppstått en regnefeil som ikke kan gjøres rede for. Bruker kan ha skrevet feil som for eksempel $\sin(xn)$.

Oppstår det et informasjonsfelt med rød farge, kan ikke brukeren fortsette visualiseringen. Da må man dessverre starte på nytt. Her ble fargen på informasjonsfeltet satt som et trafikklys. Rød betyr stopp, gul betyr varsel og grønn ble ikke brukt. Var inne på tanken om å ha grønn på informasjonen *"Tegner opp graf"*, men valgte å beholde denne hvit for ikke å miste fokuset fra koordinatsystemet. Dersom informasjonsfeltet blinket med grønn farge for hver visualisering ville dette vært et forstyrrende moment.

4 Utviklingsfasen

Under utviklingen av programmet oppsto det noen komplikasjoner som medførte mye tanker og frustrasjon men også glede. Jeg vil her beskrive problemene samt hvordan jeg løste dem, eller hva som ikke er løst og dermed hva som kan forbedres til en nyere versjon.

4.1 Programvaren

Jeg brukte programmet JCreator Pro versjon 4.00.028 og JCreator LE (Lite edition) versjon 3.50.013 som begge er utviklet av Xinox Software. JCreator LE ble lastet ned fra <http://download.cnet.com/windows/>, mens i ettertid fant jeg JCreator Pro på <http://www.jcreator.com/>. JCreator Pro er bare en 30 dagers prøveevaluering, dersom man ikke ønsket å kjøpe programmet, noe som medførte kort tidsbruk. Etter denne tidsbegrensningen brukte jeg JCreator LE som er et gratis (freeware) program. Men siden JCreator Pro har en mye bedre brukervennlighet, installerte jeg dette flere ganger på min egen datamaskinen. På skolen lot ikke dette seg gjøre og følgelig ble JCreator LE brukt her. Dette programmet trengte tilleggspakken JDK og jeg brukte Java SE Development Kit (JDK) 6 Update 11 og 12 for å kompilere kildekoden. (Det kom ut en oppdatering under utviklingen av programmet.) Samtidig ble Java SE Development Kit Documentation 6 lastet ned som ekstra hjelpestøtte. Begge disse tilleggene ble hentet fra <http://java.sun.com/>.

Jeg har brukt følgende lærebøker i JAVA under min utvikling:

- Beginning Programming with Java™ For Dummies®, 2nd Edition. (Burd, 2005)
- Core Java™ 2 Volume I - Fundamentals, Seventh Edition. (Horstmann & Cornell, 2004)
- Learning Java, 3rd Edition. (Knudsen & Niemeyer, 2005)

I tillegg har jeg brukt noen Internettssider til veiledning og de er:

- <http://java.sun.com/docs/books/tutorial/>
- <http://www.abbeyworkshop.com/howto/java/>
- <http://www.roseindia.net/java/beginners/>

4.2 Java utfordringer

4.2.1 Programmeringsspråket JAVA

Fra før har jeg vært mest borti Flash, Lingo (Shockwave) og noe C++ programmering. Derfor var det en del utfordringer for å sette seg inn i notasjonen til JAVA, men dette løste seg. Men det skal sies at jeg valgte å forandre deler av datakoden mot slutten av programmets ferdigstillelse for å gjøre kildekoden bedre. Jeg har lært masse nye metoder for å løse problemer innen JAVA teknologien. La meg illustrere:

Først brukte jeg casting for å gjøre bokstaver til tall ved å bruke CharAt(). Tallet "13.45" er i programmet ikke definert som et tall, men som tekst siden man skriver inn uttrykket i et tekstfelt. Jeg lagde en rutine som fant ut hvilken plass kommategnet var plassert, for så å multiplisere med hvilken posisjon i tallsystemet tallet stod på. Dette var både komplisert og tidkrevende og medførte litt unøyaktighet grunnet avrundingsfeil som blir beskrevet under punkt **4.3.4 Standardform**. Men mot slutten av implementeringsfasen fant jeg en metode

som var mer effektiv. Klassen Double har en egenrutine, kalt parcing, for å konvertere tekst til tall. Jeg leitet godt etter en slik rutine i begynnelsen, men fant det dessverre ikke, noe som medførte masse ekstraarbeid for min egen del. Det var i boken Learning Java, 3rd Edition (Knudsen & Niemeyer, 2005) at jeg fant parcing metoden.

4.2.2 Panelløsningene

Inndelingen til de forskjellige delene av designet er gjort ved hjelp av panelløsninger. Disse hadde jeg problemer med i begynnelsen, men de løste seg etter en del jobbing. Fordelen med denne løsningen er at vinduene (koordinatsystemet, navnet, knappene og utviklingsfeltet) stiller seg inn automatisk og fordeler seg ut fra bredde og høyde på skjermen til brukeren.

4.2.3 Plassering av origo

Plasseringen av origo var i startfasen av utviklingen ikke noe problem. Valget falt naturlig og strategisk på midtpunktet av høyden og bredden til koordinatsystempanelet. Dette ble gjort for å få symmetri for brukeren. Mennesker bruker selv ofte symmetri, og designmessig ser det pent ut ved bruk av symmetriegenskaper.

Problemet oppstod når innstillinger av de forskjellige koordinatverdiene skulle transformeres og det ikke oppstod symmetri rundt origo. Hele mitt første design var bygd opp rundt symmetri. Dermed måtte jeg forandre deler av oppbygningen. Dette var en verdifull erfaring for meg og i ettertid tilsier dette at idéskissene burde vært mer gjennomtenkt før jeg startet på implementeringsfasen. Med mer rutine hadde jeg nok unngått dette problemet. Metodene for utregningen fungerte likevel bra, men problemet var ombygging av selve tegnerutinene av funksjonene inkludert opptegning av tall, akser og rutenett.

Siden plasseringen av origo blir stilt automatisk ved hjelp av verdiene på x -aksen og y -aksen måtte jeg gjøre bruk av transformasjonsregning. Origo til koordinatsystemet blir plassert i et tilfeldig (x, y) punkt på datamaskinens koordinatsystem og så måtte jeg regne på en transformering $(x \pm a, y \pm b)$. Deretter måtte x -verdien og funksjonsverdien bli transformert til $(x \pm a \pm x\text{-verdi}, y \pm b \pm \text{funksjonsverdi})$.

Et lite problem ble også at en datamaskin definerer sitt origopunkt øverst til venstre på skjermen. Positiv retning til x er mot høyre og positiv retning til y er nedover.

Siden man oftest regner med positiv retning oppover på y verdiene, ble det litt tanker som lett ble snudd på hodet. Men heldigvis hjalp det å bruke tegninger for å få visualisert transformasjonen sammen med egne utregninger på papir.

4.2.4 Analysering av teksten

Brukeren skriver inn uttrykkene som tekst, og utfordringen for meg som programmerer, var at jeg måtte analysere teksten og så bryte det ned til tall og matematiske operasjoner.

Her hadde jeg en del problemer med å gjøre teksten om til tall som beskrevet under punkt

4.2.1 Programmeringsspråket JAVA.

4.2.5 Skjulte tegn

I matematikken er det mange skjulte tegn. Her måtte jeg analysere det brukeren hadde skrevet inn og legge til de skjulte tegnene. Med skjulte tegn mener jeg for eksempel at $3x$ betyr egentlig $3*x$. Datamaskinen sitt tegn for multiplisering er $*$. Slike tegn tar man ofte for gitt i en matematisk situasjon, men i en algebraisk situasjon i den digitale verden må man legge til slike tegn for å kunne utføre den aktuelle operasjonen. Her måtte jeg også tenke på at det er ikke bare $3x$, men også kombinasjoner med parenteser, opphøyd, pluss-minus operasjoner... etc. Her må jeg påpeke at jeg kan se vanskeligheten for mange elever. For å utføre mange matematiske operasjoner må man kjenne til de skjulte tegnene. På dette området fikk jeg en oppfrisking på at slike tegn må presiseres ekstra godt for elever i en undervisningssituasjon.

Jeg har også valgt å kutte vekk skrivemåten $\sin^3 nx$ for det er vanskelig å vite om brukeren ønsker å skrive inn $(\sin nx)^3$ eller $(\sin x)^3 n$ eller liknende.

Det måtte også sjekkes om n var en variabel i uttrykket eller en del av en matematisk operasjon som for eksempel $\sin/\tan/\ln$.

4.2.6 Automatisk innlesning fra fil

Innholdet til funksjonaliteten ”hjelp” og ”ferdige $f(x)$ ” blir lest automatisk fra filer.

Først brukte jeg en metode som ser ut slik (utdrag):

```
File f = new File(filnavn);
FileInputStream fis = new FileInputStream(f);
BufferedInputStream bis = new BufferedInputStream(fis);
dis = new DataInputStream(bis);
```

Alt fungerte bra, helt til jeg kjørte programmet fra en webserver. I første stund skjønnte jeg ikke hvorfor det fungerte lokalt på maskinen og ikke fra en webserver. Men etter en liten stund fant jeg ut at når man kjører programmet fra en webserver, har man selvsagt ikke rettigheter til å lese filer på denne måten. Dette er en ekstra sikkerhet for brukeren som medfører at ingen personer kan utføre uønskede handlinger på maskinen, som for eksempel å lese personlige filer. Dermed måtte jeg løse dette problemet ved hjelp av en annen metode.

Løsningen ble som følger (utdrag):

```
url = new URL(getCodeBase(), fileToRead);
InputStream in = url.openStream();
BufferedReader bf = new BufferedReader(new InputStreamReader(in));
stringBuff = new StringBuffer();
```

Med denne metoden har programmet muligheten til å lese fra sitt eget webområde og følgelig kunne de aktuelle tekstfilene leses. Denne automatiske innlesningen medfører en enkel oppdateringsmulighet for læreren som ønsker å bruke programmet og er blitt beskrevet i punkt **3.5.5 Ferdige $f(x)$** og punkt **3.5.6 Hjelp**.

4.3 Kalkulasjonsutfordringer

4.3.1 Rekursive rutiner (Rutiner kaller opp seg selv)

I programmet brukte jeg først en metode som gjør at en rutine kaller opp seg selv.

For eksempel $1 + (((3x) + 2)n)^n$.

Etter den matematiske prioriteten skal parenteser regnes ut først og jeg har implementert en rutine som regner ut parenteser. Her ser vi at det er flere parenteser som skal regnes ut men inne i parentesen $((3x) + 2)n$, er det flere parenteser. Programmeringen vil da "leite" etter flere parenteser og den vil prøve å regne ut $(3x)$ deretter (svaret fra $3x + 2$) ... osv.

Parentesrutinen bruker her rekursiv metode for å kalle opp seg selv.

Dette medfører at jeg som programmerer slipper å tenke på alle mulige kombinasjoner av matematiske uttrykk, men igjen medfører dette at programmet må foreta noen flere kalkulasjoner som kan ta litt lenger tid. Fordelen er at man trenger bare en rutine for hver matematisk operasjon. Når den matematiske operasjonen er gjennomført, vil metoden sjekke seg selv igjen. Dersom det ikke er noe av den aktuelle prioriteten går den videre til neste matematiske prioritet. Men etter min oppdagelse av parcing fra tekst til tall som medførte til dels store endringer i kildekoden, lot jeg noen av rutinene bruke for-løkker og noen rutiner brukte rekursive funksjonskall.

4.3.2 Potenser

Problemet med potenser er hvordan programmet skal vite om det er $-x^2$ eller $(-x)^2$. Programmet er bygd opp etter matematiske prinsipper, slik at parenteser har høyest prioritet. Derfor blir $(-x)^2$ regnet ut ved at parentesen $(-x)$ blir regnet ut først.

Når dette uttrykket er regnet ut i rutinen for parenteser vil det i tekstform stå $-x^2$ (x 'en vil være byttet ut med et tall) og problemet oppstår. Her måtte jeg sjekke om parentesen som ble regnet ut var en potensregning som skulle regnes ut i etterkant. Hvis den var det sendte jeg inn parentesutregningen og eksponenten inn til potensrutinen. Men nå måtte potensrutinen vite at det var et negativt tall som skulle opphøyas. Grunnen for at jeg valgte å si at det var et negativt tall som skulle opphøyas, er at programmet må klare å skille mellom et negativt tall og subtraksjonsoperatoren.

En annen utfordring var om potensverdien er en lovlig verdi i de reelle tallene, for som vi vet

er ikke $(-2)^{\frac{1}{2}} \notin \mathbb{R}$. Jeg forklarer mer under punktet **4.3.3 Uendelig og ulovlige verdier**.

En annen utfordring var når funksjonsverdien ble for høy eller for lav, som blir beskrevet under punktet **4.3.4 Standardform**.

4.3.3 Uendelig og ulovlige verdier

Det finnes noen matematiske uttrykk som medfører at funksjonsverdien blir uendelig, for

eksempel $\lim_{x \rightarrow 0} \frac{1}{x}$, eller at funksjonsverdien ikke er element i de reelle tallene, for eksempel

$(-2)^{\frac{1}{2}} \notin \mathbb{R}$.

Java sine metoder for de matematiske utregningene gir forskjellige svar.

Er det en uendelig verdi som oppstår blir svaret ”– *Infinity*” returnert og følgelig er en funksjonsverdi ikke mulig å bestemme og det medfører at punktet (x, y) ikke er mulig å tegne opp.

Det er også mulig å få ulovlige verdier som ikke er element i de reelle tallene. I dette tilfellet returnerer Java et svar som er ”*NaN*”.

Dette medfører også at den aktuelle (x, y) verdien ikke blir tegnet opp.

Det er en boolsk variabel i programmet som styrer om punktet (x, y) skal bli tegnet opp. Navnet på denne variabelen valgte jeg til *Infinity*. Hvis det blir noen kalkulasjoner som tilsier at det ikke skal tegnes et punkt (x, y) , blir verdien til den boolske variabelen satt til sann (dataverdi *true*), og programmet utelukker punktet i opptegningsrutinen.

4.3.4 Standardform

Den digitale verden får et ”problem” når tallene blir veldig små eller veldig store. Når man som programmerer i JAVA skal definere variabler må man bestemme hva slags datatype variabelen skal være.

Man kan velge mellom

- Integer (heltall)
- Double (kommatal)
- Long (større heltall)
- Char (bokstav)
- String (tekst)
- med flere...

Problematikken er at integer og double har begrensninger i antall siffer. Blir verdien for høy eller for lav blir den skrevet på standardform. Dette medførte mye frustrasjon under utviklingen av programmet. Først laget jeg en metode for å runde av tallet som var skrevet på standardform. Dette medførte en unøyaktighet i utregningen som jeg ikke ønsket.

Deretter falt valget på å gjøre tallene som var skrevet på standardform om til datatypen *BigDecimal* og gjøre det om til tekst i form av tall med mange siffer. Her oppstod problemet at det ble ekstremt mange desimaler etter komma som igjen medførte at jeg måtte ta avrundinger. Her oppstod det av og til noen rare utregninger som for eksempel $2+1$ kunne bli 3.0000000000000001 som gav en unøyaktighet samt medførte andre komplikasjoner.

Men mot slutten av utviklingen løste jeg dette problemet på en annerledes måte, noe som var et resultat av at min forståelse i JAVA ble bedre, beskrevet under punkt **4.2.1**

Programmeringsspråket JAVA.

Det skal være sagt at dersom funksjonsverdien til en x -verdi blir et tall skrevet på formen $E+*$ eller E^* , hvor $*$ er et heltall, blir det av og til vanskeligheter med å tegne opp punktet (x, y) . (Av og til skrev JAVA standardformen som $E+*$ eller E^* , hvor $*$ er et heltall.)

4.3.5 Hakking ved mange n - ledd

Under den delen av programmet som tar for seg rekkeutvikling kan det ved mange delsummer, oppstå mange kalkuleringer. Dette kan medføre litt tid før visualiseringen av delsummene blir tegnet opp, så ikke bli utålmodig. Når det er mange delsummer til rekkeutviklingen medfører det mange utregninger. Dersom du ønsker å zoome med musen, spesielt under rekkeutvikling med mange delsummer, kan det oppstå litt hakking.

Det anbefales derfor at man stiller inn det ønskede vinduet ved innstillingsvalget ”*Still vindu*”, eller at man fjerner noen av de tidligere visualiserte delsummene med innstillingsvalget ”*Fjern Fn*”.

Det samme problemet kan oppstå dersom man flytter vinduer.

Grunnen er at koordinatsystemet er satt til å tegne opp alle funksjonsfølger eller rekkeutviklinger ved forandring i vinduet. For å få grafene så glatte som mulig, har jeg valgt å gjøre en utregning av funksjonsverdien for hver piksel. Dette medfører svært mange kalkulasjoner, men dette gir en bedre og glattere visualisering.

4.3.6 Fakultet utregninger

Under utregningen av faktulteter brukte jeg først datatypen long for å regne ut verdien av fakultetet. Long ble valgt siden faktulteter alltid vil forbli et heltall og datatypen long har muligheten til å ha høyere verdier enn en integer. Ved valg av long datatypen oppstod det et problem ved fakultet over 20. Datatypen har nemlig ikke mulighet til å ha så store verdier, og det var en stund ingen løsning på dette problemet. Men når jeg fikk utbedret programmet til å håndtere standardform, valgte jeg fakultetutregningen som en integer. Dermed kan programmet nå ta fakultet over verdien 20, men det er satt sperre på maksimalt 100 fakultet.

4.4 Forbedringsmuligheter til programmet

Her vil jeg skrive noen punkter som kan gjøre programmet bedre, men som dessverre ikke er implementert per dags dato grunnet begrensninger i både tid og studiepoeng.

- Lage en bedre metode for å tegne opp en ny funksjon.
Nå tegnes alle funksjonene opp for hver gang, dersom de skal vises, noe som medfører mange utregninger, som igjen fører til litt hakking beskrevet under punkt **4.3.5 Hakking ved mange n - ledd**.
- Implementere en mulighet som avgjør når en funksjon krysser epsilonintervallet.
- Legge til mulighet for å skrive inn absoluttverdi tegn.
- Legge til mulighet for å tegne opp flere uttrykk enn det utviklingsfeltet har ”plass” til. Kan kanskje bruke modulusoperatoren, slik at når utviklingsfeltet er ”fult” kan det begynne på nytt igjen. Dette medfører at uendelig mange funksjonsfølger eller rekkerdelsummer kan visualiseres.
Per dags dato er det bare et endelig antall rekkeledd som kan tegnes, maks 25 utviklinger.
- Implementere at brukeren kan skrive inn $\sin^{(3)}x$ eller liknende og få svaret som $(\sin x^3)$.
- Legge inn muligheten til å se derivert i punkt (x,y) .

5 Eksempler på bruk av programmet

5.1 Eksempel 1: Punktvis konvergens

Jeg vil nå vise en oppgave der jeg bruker programmet til å visualisere punktvis konvergens.

Oppgave 1.

La $f_n(x) = \sin^n(x)$, der hver f_n er definert på intervallet $[0, \pi]$.

(a) Tegn grafene til f_1, f_2 og f_3 .

(b) Tegn grafen til f_{100} .

(c) Finn et uttrykk for grensefunksjonen. Tegn også grafen til grensefunksjonen.

(Nygaard, 2007, p. 87)

Oppgaven går ut på å tegne opp grafene til

$f_1(x) = \sin^1(x)$, $f_2(x) = \sin^2(x)$, $f_3(x) = \sin^3(x)$ og $f_{100}(x) = \sin^{100}(x)$.

Vi skriver inn funksjonsfølgen.

Legg merke til at programmet må skrive uttrykket $\sin^n(x)$ som $(\sin x)^n$. Grunnen er beskrevet under punkt **4.2.5 Skjulte tegn**.

Skriv inn uttrykk

<input type="text" value="(sinx)^n"/>
Tegn graf

Figur 28: Skriv inn funksjonsfølgen som er gitt i oppgaveteksten

Vi stiller inn vinduet slik i det oppgitte intervallet $[0, \pi]$ og fornuftige verdier på y .



X-Min :	-0.1
X-Max :	pi
X-intervall:	0.2
Y-Min :	-0.1
Y-Max :	1.1
Y-intervall:	1.0
Avbryt Sett verdier	

Figur 29: Stiller inn vinduet

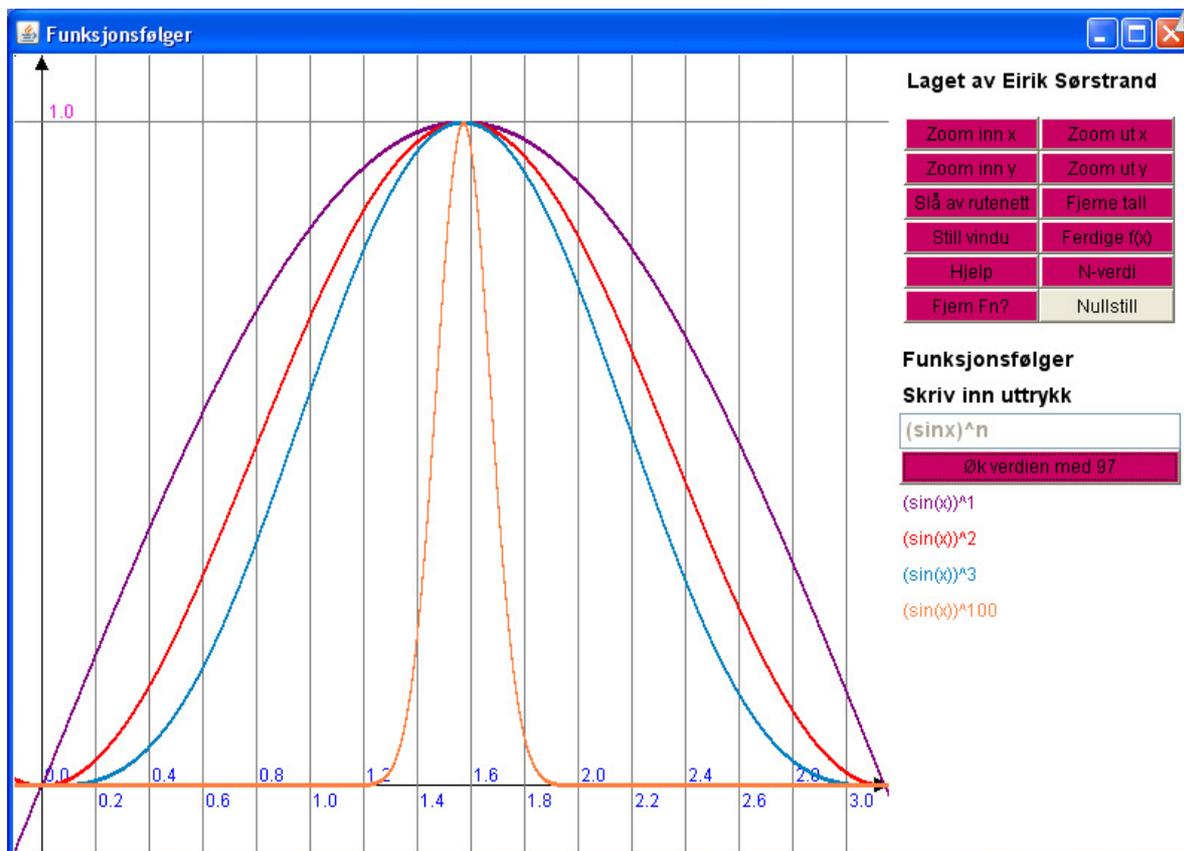
Legg også merke til at n begynner på 1 og ikke 0. Programmet har en løsning på dette problemet, og du kan selv velge ønsket verdi i økningen og startverdi av n . (se punkt 3.5.7 N-verdier.)



Figur 30: Vi setter korrekt verdier på n

Vi tegner de 3 første leddene av funksjonsfølgen, deretter må vi forandre økningen av n leddet med 97 for å få den siste funksjonsfølgen til å bli $\sin^{100}(x)$.

Da ser utviklingen slik ut:



Figur 31: Visualisering av $f_1(x)$, $f_2(x)$, $f_3(x)$ og $f_{100}(x)$

Her tegner programmet opp funksjonsfølgene og gir oss en mulighet til å vurdere og analysere området for punktvis konvergens, i dette tilfellet over intervallet $[0, \pi]$. Skal man prøve å tenke på denne analysen ved kun å se for seg funksjonsfølgene i hodet, medfører det en veldig vanskelig situasjon.

Tiden man vil bruke på å tegne opp disse grafene manuelt er veldig stor i en undervisningssammenheng og stjeler mye tid fra det som er hovedfokuset, nemlig analysen av konvergensintervallet.

La meg illustrere.

Vi kan observere, hvis vi bruker definisjonen på punktvis konvergens, at $g(x)$ blir følgende:

$$g(x) = \begin{cases} 1, & x = \frac{\pi}{2} \\ 0, & x \in [0, \pi] \setminus \{\frac{\pi}{2}\} \end{cases}$$

Vi ser her et tydelig eksempel hvor g ikke er kontinuerlig, selv om hver $f_n(x)$ er uendelig mange ganger deriverbar. Dersom vi ikke hadde hatt et slikt visualiseringsverktøy ville utfordringen vært å løse $\lim_{n \rightarrow \infty} f_n(x)$ manuelt for hver x .

I eksemplet over ser man at jo nærmere x -verdien blir $\frac{\pi}{2}$, desto større verdi må n ha for at funksjonsfølgen skal komme nær 0.

5.2 Eksempel 2: Uniform konvergens

Jeg vil nå bruke programmet til å analysere om oppgaven har uniform konvergens.

Vi setter grensefunksjonen $g(x) = 0$ med et epsilonintervall lik 0,1.



Figur 32: Vi gjetter en grensefunksjon

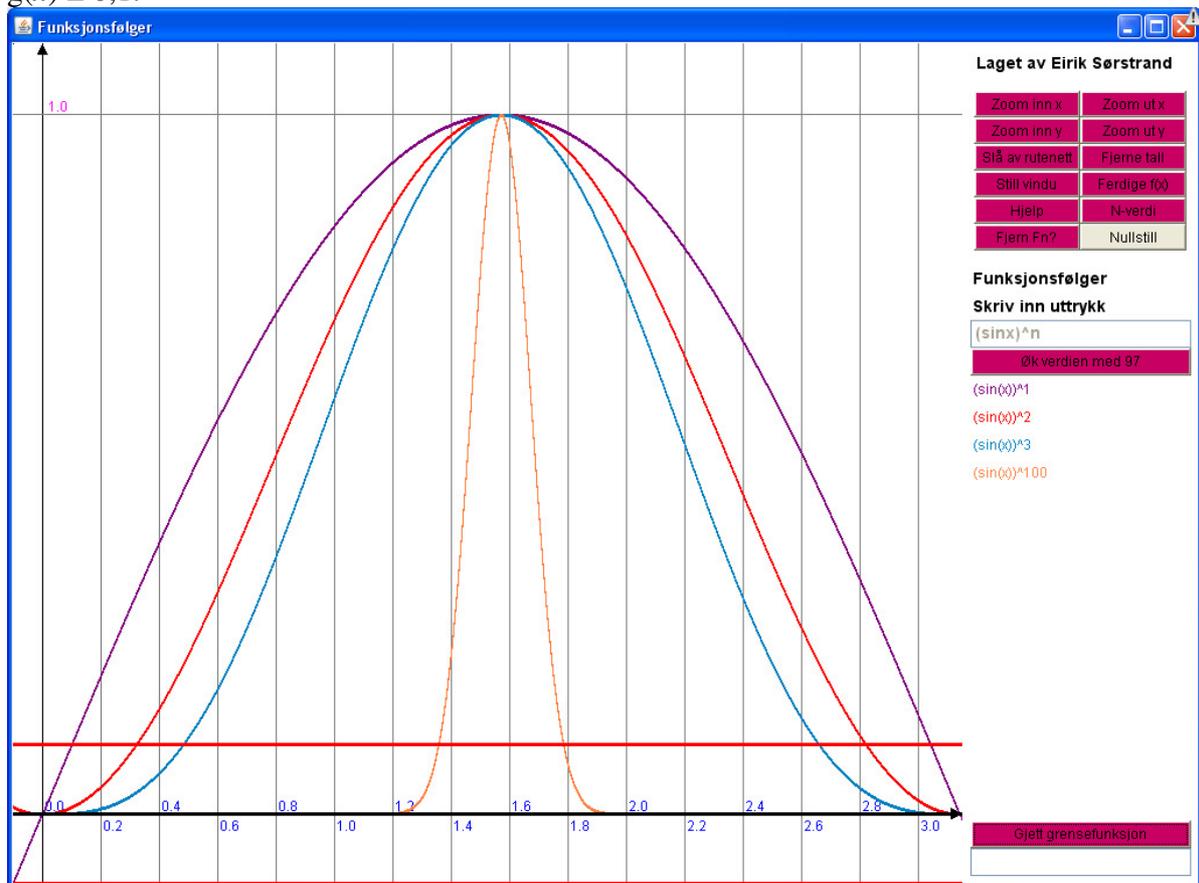
Vi skal observere i hvilket område følgen går utenfor epsilonintervallet.

Målet med analysen er å se hvilken n som kreves for å komme under epsilonintervallet til de følgende x verdier.

$x = 1.4$		$x = 1.5$		$x = 1.55$	
ϵ	$n(x)$	ϵ	$n(x)$	ϵ	$n(x)$
0.1		0.1		0.1	

Tabell 1: Hvilke verdier av n medfører funksjonsverdi lavere enn epsilon?

Legg merke til de røde horisontale linjene som er epsilonintervallet og de har verdi lik $g(x) \pm 0,1$.



Figur 33: Utviklingen av følgen når $n=100$

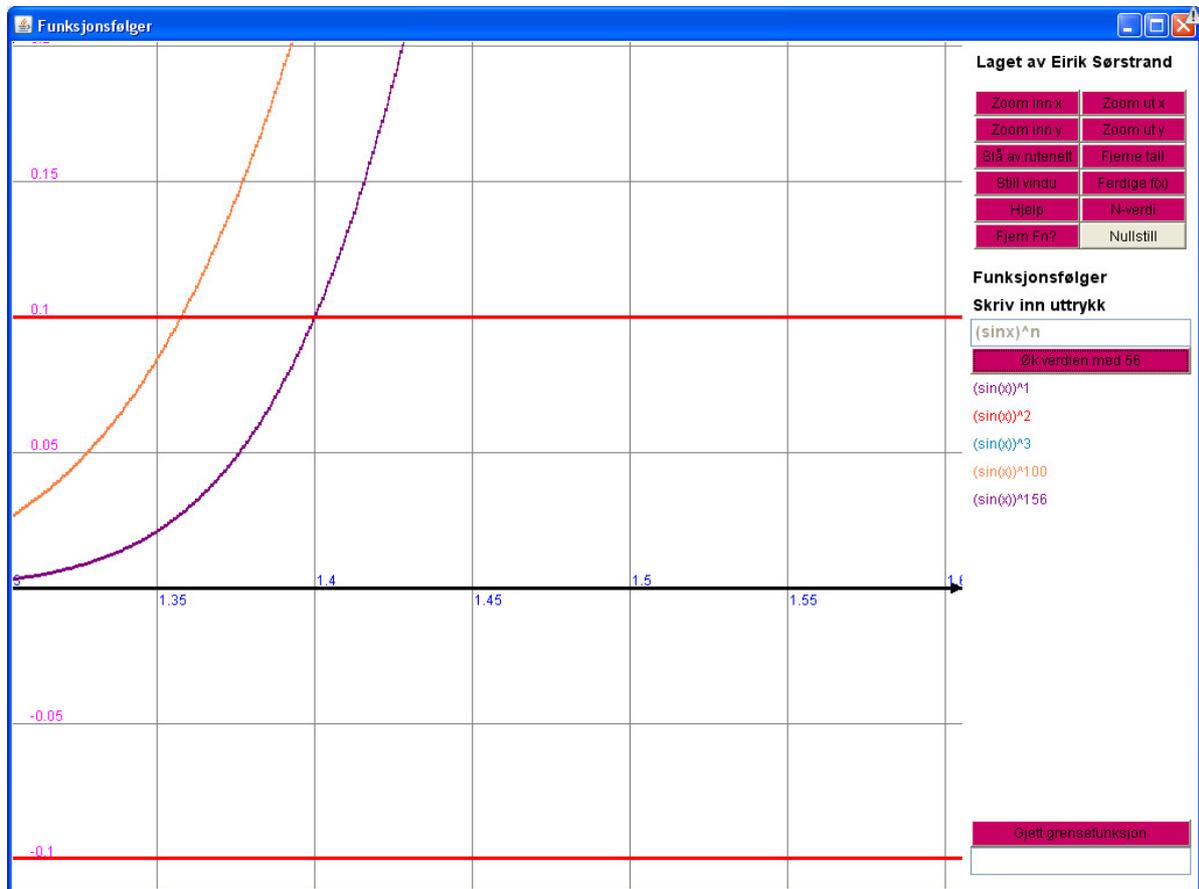
Siden vi skal avgjøre når funksjonen kommer under epsilonintervallet i området fra 1.4 til 1.55 må vi stille på koordinatsystemet.

Området som velges er:



Figur 34: Vi stiller koordinatsystemet

Vi prøver oss frem til en n verdi som medfører at funksjonsverdien kommer under epsilonintervallet. Merk at jeg ikke viser hvordan vi forandrer n verdien, siden dette er vist tidligere under punkt **5.1 Eksempel 1: Punktvis konvergens**.



Figur 35: funksjonsfølgen ved $n = 156$

Som vi kan observere må vi ha en $n = 156$ for at $x = 1.4$ får en funksjonsverdi som ligger på epsilonintervallet.

$x = 1.4$	
ϵ	$n(x)$
0.1	156

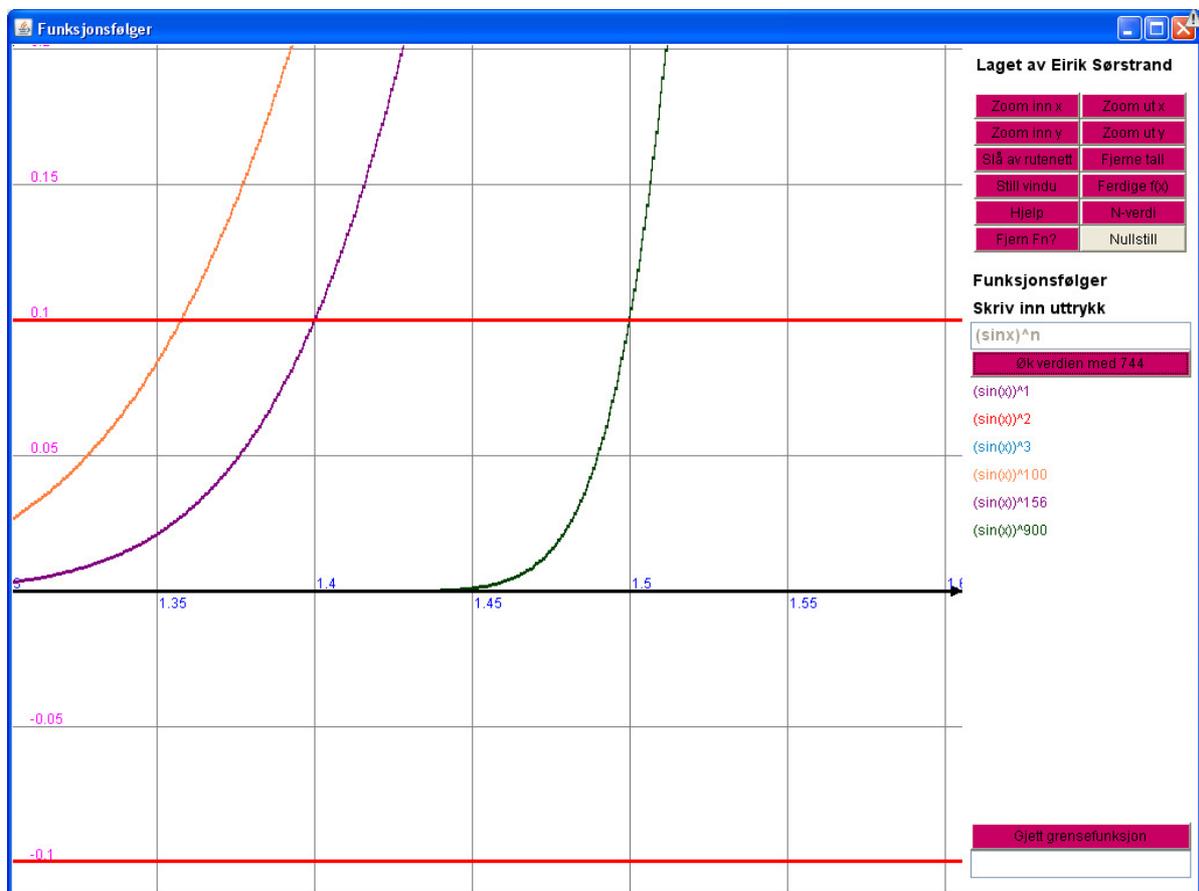
$x = 1.5$	
ϵ	$n(x)$
0.1	

$x = 1.55$	
ϵ	$n(x)$
0.1	

Tabell 2: Funnet n verdien for $x=1.4$

Nå gjenstår bare å finne en n slik at $x = 1.5$ og $x = 1.55$ kommer under epsilonintervallet.

Etter et par forsøk finner vi at $n = 900$ gir oss:
 (Brukt programmet med mange forskjellige verdier til n for å finne krysningspunktet, men viser ikke dette)



Figur 36: Utviklingen av følgen når $n=900$

Som vi kan observere må vi ha en $n = 900$ for at $x = 1.5$ får en funksjonsverdi som ligger på epsilonintervallet.

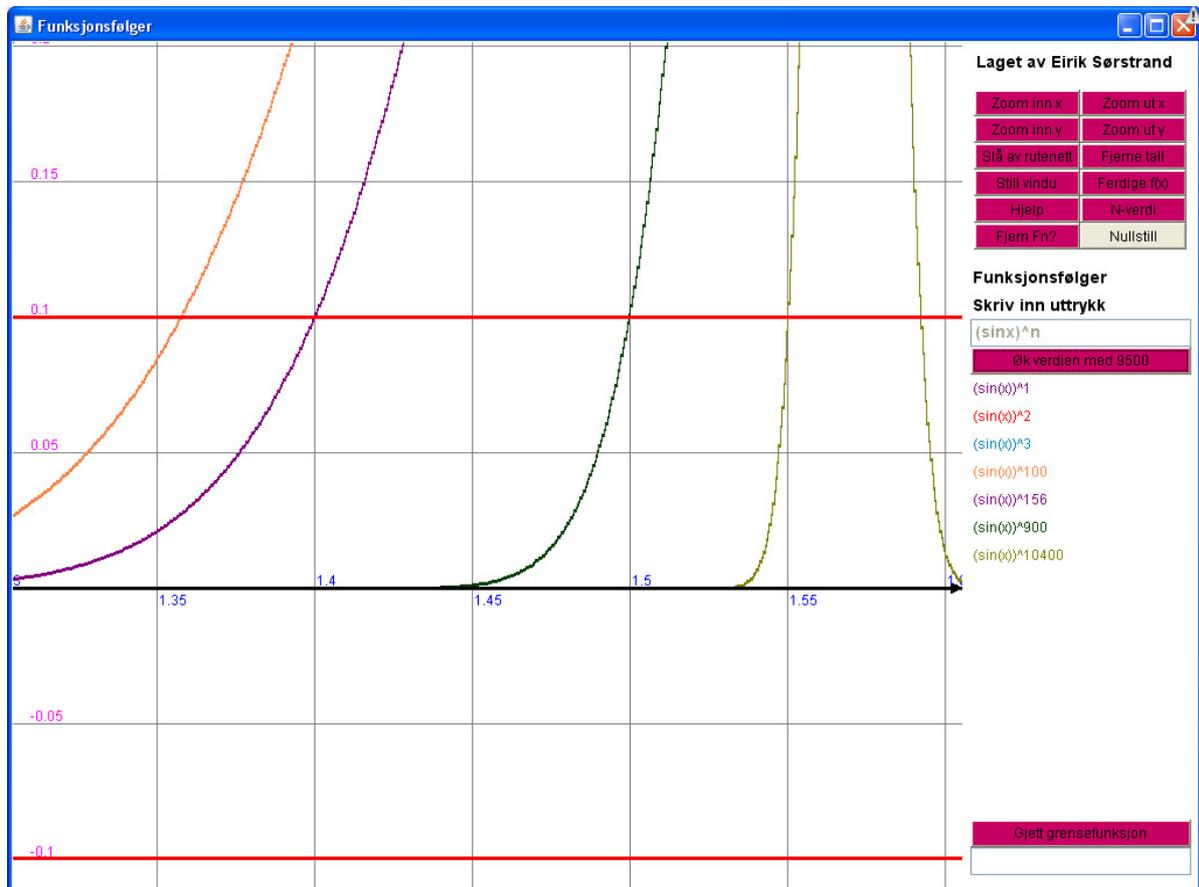
$x = 1.4$	
ϵ	$n(x)$
0.1	156

$x = 1.5$	
ϵ	$n(x)$
0.1	900

$x = 1.55$	
ϵ	$n(x)$
0.1	

Tabell 3: Funnet n verdien for $x=1.5$

Vi leter etter den siste n verdien som får funksjonsverdien i punktet $x = 1,55$ til å komme under epsilonintervallet.



Figur 37: Utviklingen av følgen når $n=10400$

Etter et par forsøk med ”prøv og feil” metoden, finner vi at $n = 10400$ i punktet $x = 1.55$ gir en funksjonsverdi som ligger på epsilonintervallet.

$x = 1.4$	
ϵ	$n(x)$
0.1	156

$x = 1.5$	
ϵ	$n(x)$
0.1	900

$x = 1.55$	
ϵ	$n(x)$
0.1	10400

Tabell 4: Funnet n verdien for $x=1.55$

Som vi ser er n verdien økende til de forskjellige x verdiene.

Jo nærmere vi kommer $\frac{\pi}{2}$, desto større n verdi må vi ha for å få den faste epsilonverdien.

Dersom vi skal ha uniform konvergens må enhver x verdi ha samme verdi av n for å komme under epsilonintervallet.

Dermed kan vi konkludere med at det ikke er uniform konvergens i denne funksjonsfølgen.

5.3 Eksempel 3: Visualisere en rekkeutvikling som konvergerer

Som beskrevet under punkt 1 **Møte med rekker i Kalkulus**, er det ikke enkelt å se at

$$1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots = \sum_{n=0}^{\infty} \frac{x^n}{n!} \text{ konvergerer mot } e^x \text{ for alle } x.$$

Jeg vil gjerne vise at med programmet jeg har utviklet, er det enkelt å illustrere at rekkeutviklingen faktisk går mot e^x .

Som eksempel har jeg valgt oppgaven som boken i Kalkulus bruker og som jeg henviste til under punkt 1 **Møte med rekker i Kalkulus**.

Jeg vil nå visualisere at rekken $\sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$

konvergerer mot e^x for alle x , ved å bruke programmet.

1. Først skriver vi inn rekkeuttrykket.

Rekkeutvikling

Skriv inn rekkeuttrykk

Figur 38: Skriv inn rekkeuttrykket uten sigmategnet

2. Sjekk at verdiene til n er riktig i forhold til rekkeutvikling. Disse innstillingene finner du under knappen "N-verdi". Godkjenn verdiene med valget "Sett verdier".

N-verdi

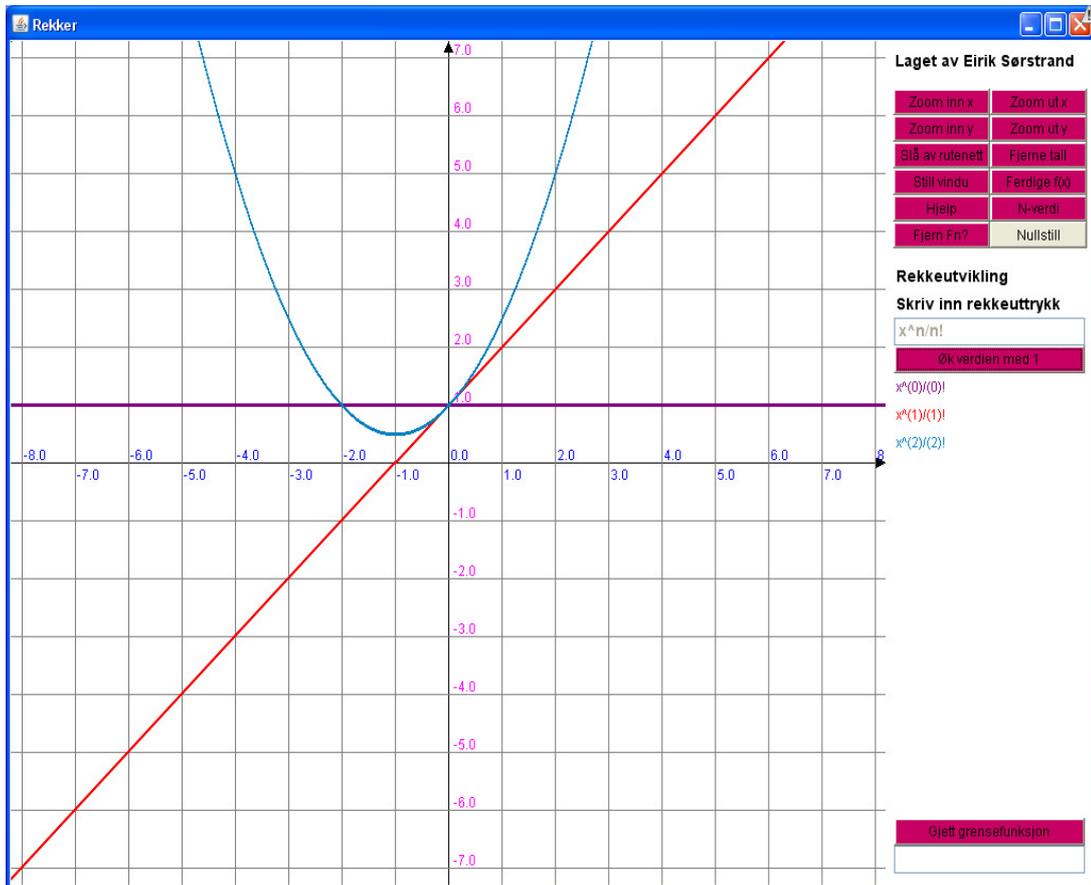
Startverdi til n
< [] >
Startverden til n er: 0

Øk verdien til n med:
< [] >
n verdien økes med: 1

Sett verdier
Avbryt

Figur 39: n verdiene stilles

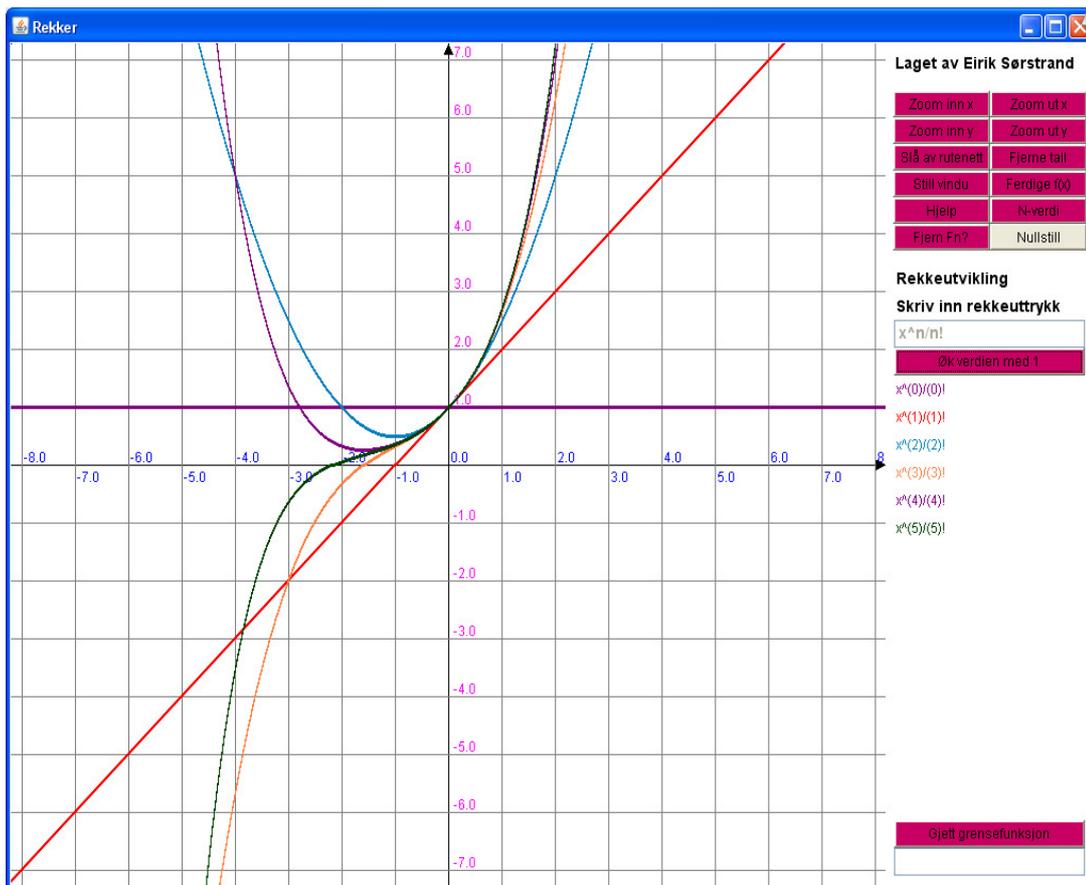
3. Vi adderer et par delsummer i rekkeutviklingen med valget "Tegn graf" og deretter "Øk verdien med *", hvor * er økningen i n -verdi.
Jeg velger å vise fram til delsum $n = 2$.



Figur 40: Utviklingen av rekken slik den ser ut frem til delsum $n=2$

Som vi ser er starten av delsummene veldig lite lik sluttresultatet som er antatt å være e^x , men vi fortsetter med å addere noen delsummer til rekkeutviklingen.

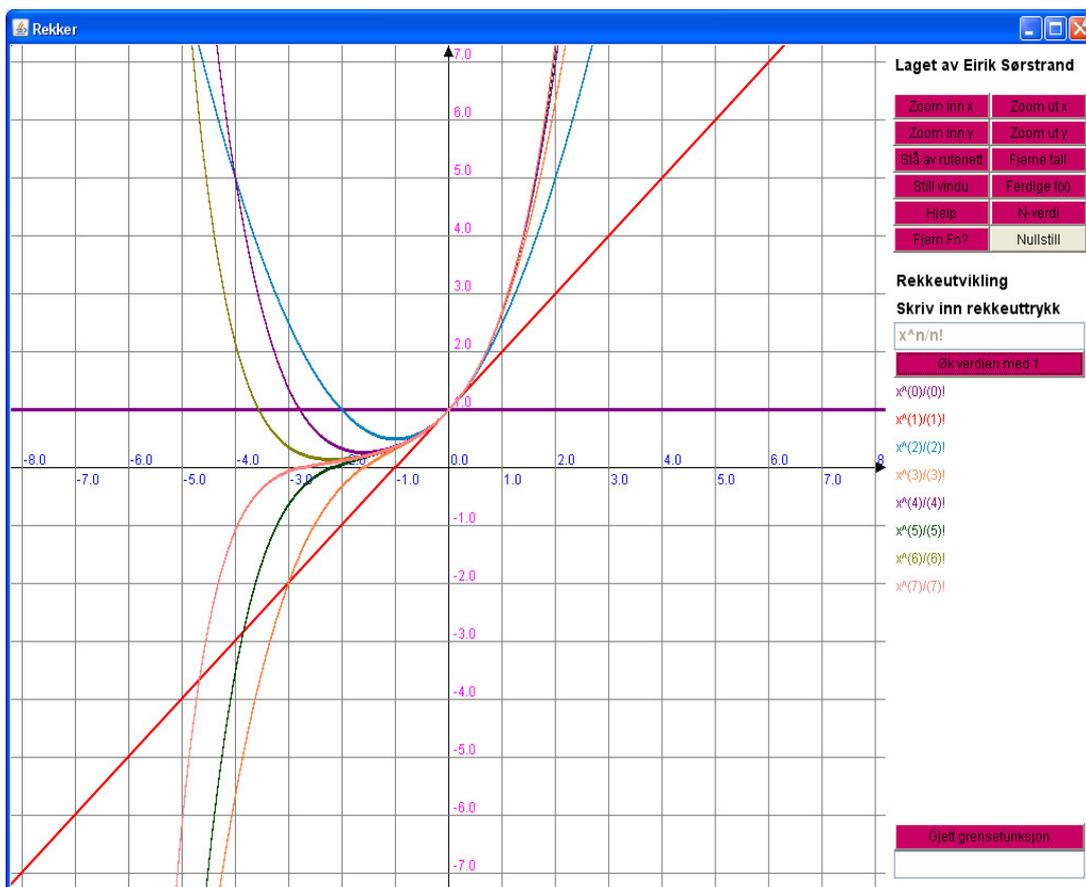
4. Vi fortsetter å addere flere delsummer slik at $n = 5$:



Figur 41: Utviklingen av rekken slik den ser ut frem til delsum $n=5$

Rekkeutviklingen begynner å ligne noe kjent, men det er enda litt usikkerhet med hensyn til konvergeringen.

5. Vi fortsetter å addere flere delsummer slik at vi får $n = 7$ og dermed har vi fått et ”bedre” bilde.

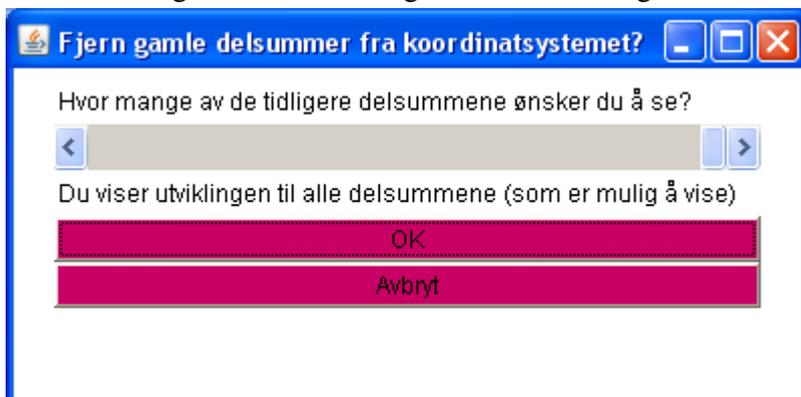


Figur 42: Utviklingen av rekken slik den ser ut frem til delsum $n=7$

Legg merke til at fargene på grafene er lik fargene på n stegene til høyre. Men det er en del støy i bildet vi skal analysere og derfor bør vi fjerne et par grafer. Vi fjerner alle grafene bortsett fra de tre siste for å gjøre det enklere å sammenlikne mot e^x .

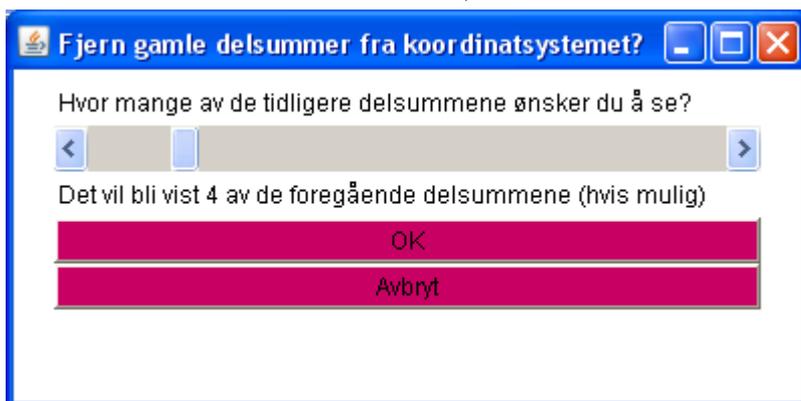
Men det er viktig å se helheten, noe vi ser tydelig når vi adderer flere delsummer til rekkeutviklingen. Dette vil vi se tydeligere fra punkt 7 i dette eksemplet. Først vil jeg fjerne litt ”støy” for å få en bedre oversikt.

6. Fjerning av grafer gjøres under valget "Fjern Fn?".
Brukeren velger selv hvor mange av de siste tidligere delsummene de ønsker å se.



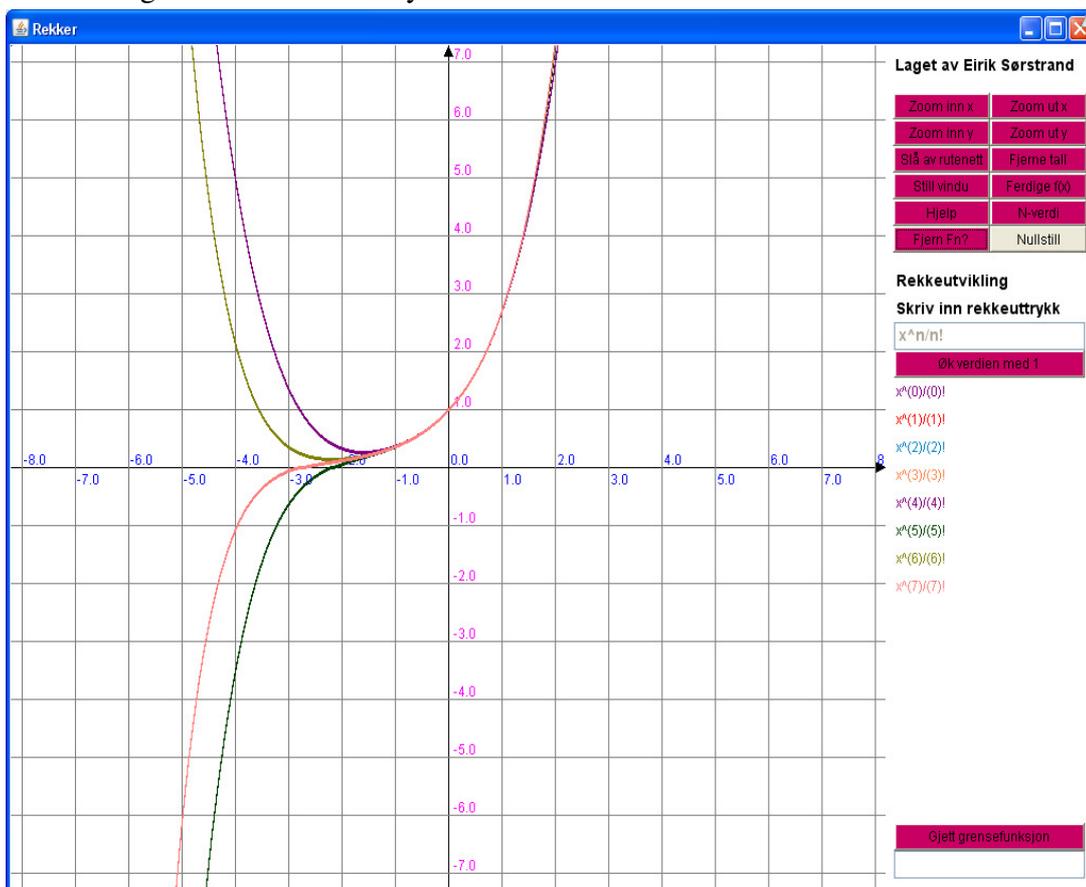
Figur 43: Alle tidligere delsummer vises

Jeg velger at vi skal se de 4 siste utviklingene.
Dersom vi adderer flere delsummer, vil bare de 4 siste utviklingene være visualisert.



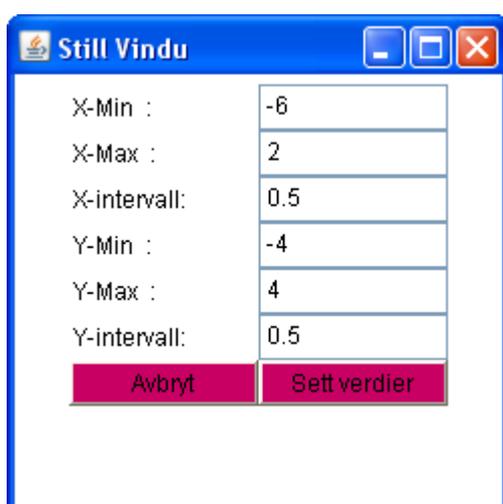
Figur 44: Vi velger antall visualiseringer

7. Et renere koordinatsystem blir synlig og det blir enklere å se hvordan rekkeuttrykket utvikler seg når vi adderer en ny delsum.



Figur 45: Vi ser de 4 siste delsummene

Som vi ser er de positive x – verdiene relativt stabile, men det er store forandringer på de negative x -verdiene.

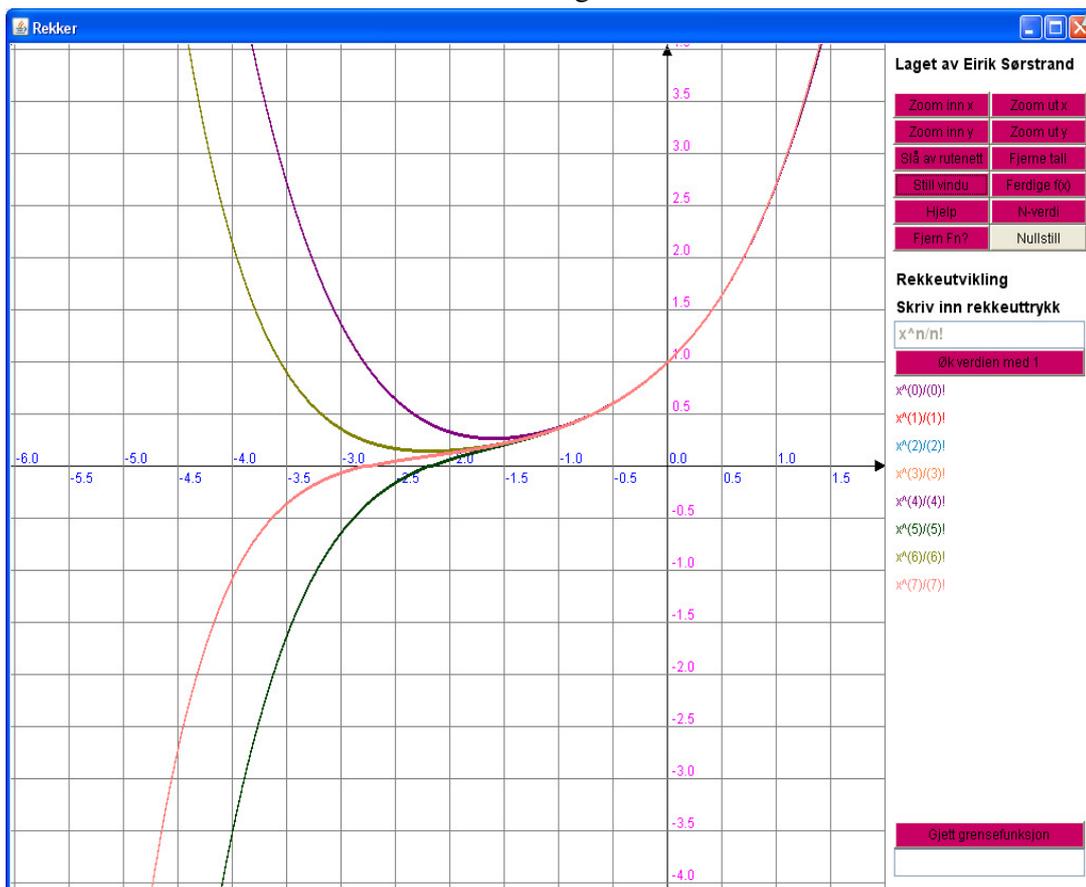


Figur 46: Vi stiller innstillingene til koordinatsystemet

Jeg velger å stille verdiene til koordinatsystemet for å få et bedre bilde og dermed et bedre utgangspunkt for en analyse.

Valget faller ”tilfeldig” på disse verdiene:

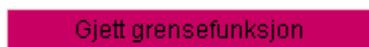
Ved et nærmere blikk vil vi se rekkeutviklingen slik:



Figur 47: Zoomet inn på det aktuelle området

Som vi ser blir området rundt 0 større for hver delsum vi adderer, men så stiger eller synker den plutselig raskt. Dette kommer av at x -verdien er en potens og følgelig vil partallpotenser gi positiv funksjonsverdi og oddetallspotens vil gi en negativ funksjonsverdi. Ved addere en ny delsum ser vi at rekkeutviklingen nærmer seg mer og mer vårt antatte konvergensområde e^x .

- Nå kommer selve gullkornet. Vi ønsker å se om denne rekken konvergerer mot e^x og dermed bruker vi funksjonaliteten "Gjett grensefunksjon".



Figur 48: Knappen "Gjett grensefunksjon"

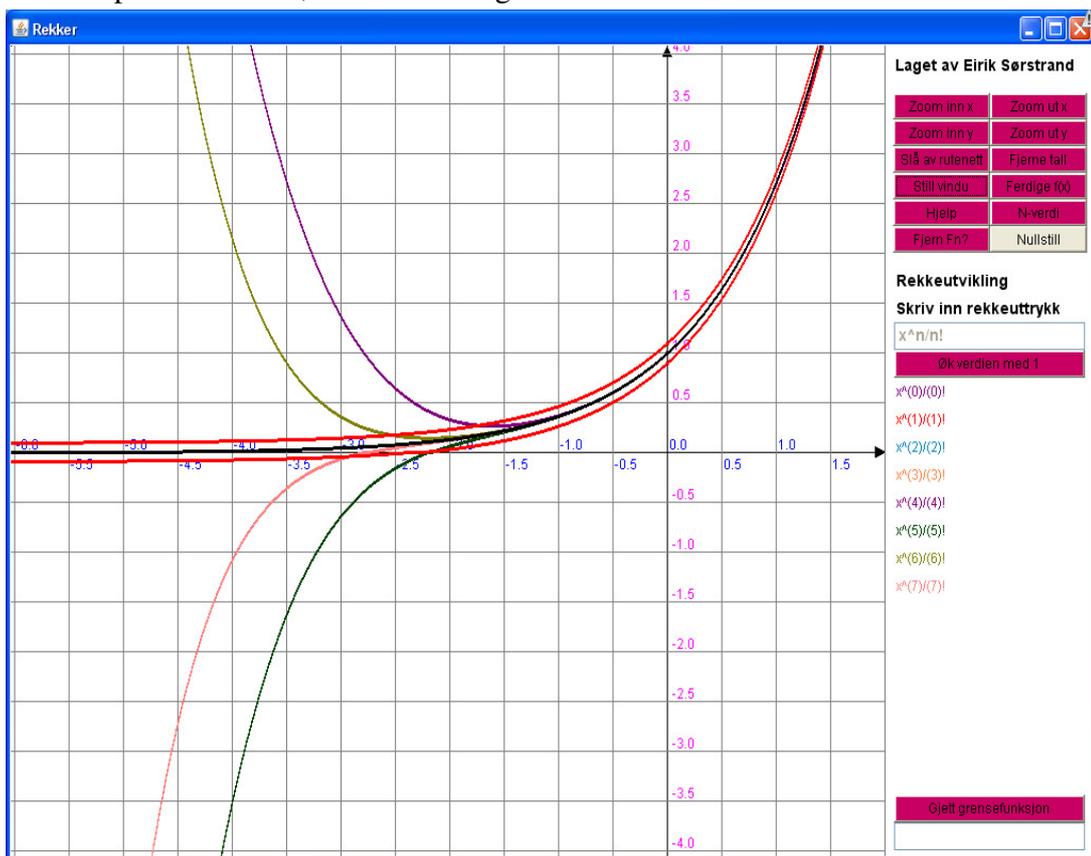
Vi får opp vinduet med "Gjett grensefunksjon" og vi tar en kvalifisert gjetning på at grensefunksjonen er e^x .

Epsilonintervallet lar vi få verdien 0.1 slik at vi kan analysere utviklingen når vi adderer flere delsummer.



Figur 49: Vi har gjettet grensefunksjonen lik e^x innenfor et intervall 0,1

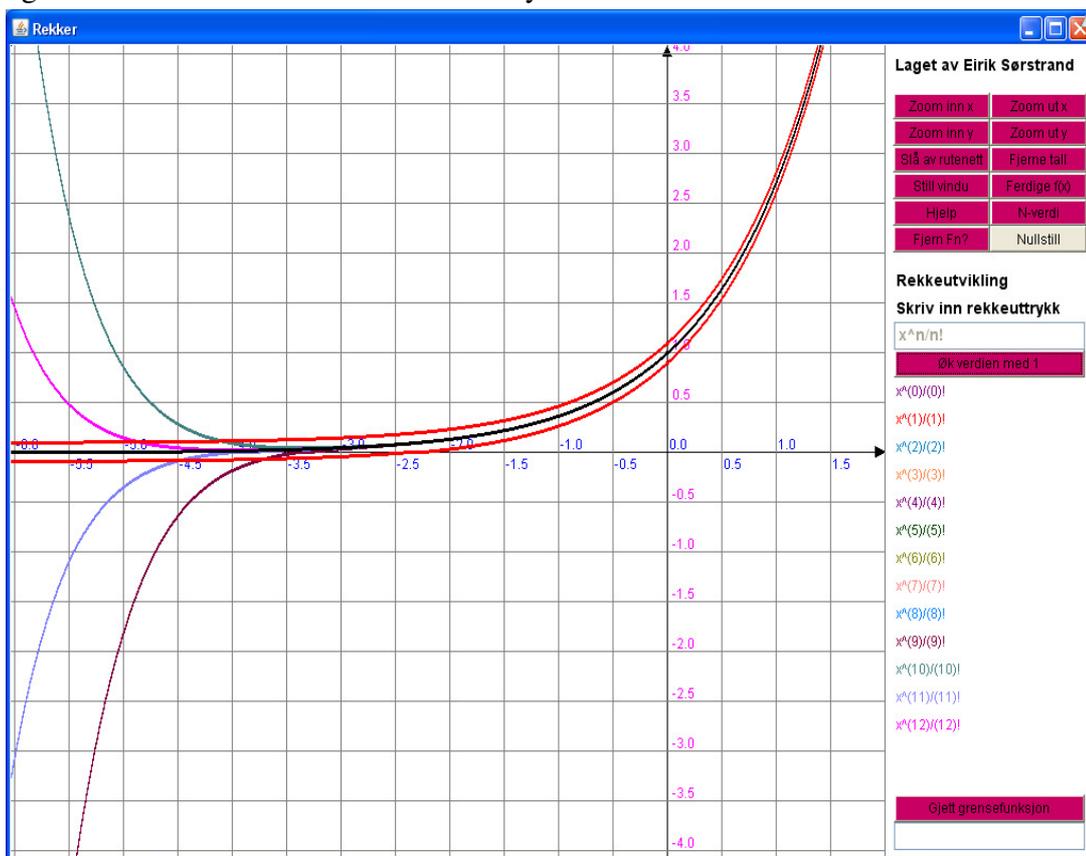
Vi godkjenner grensefunksjonen med å trykke "OK" og får tegnet opp e^x funksjonen, med et epsilonintervall, sammen med grafene av delsummene til rekken.



Figur 50: Grensefunksjonen er tegnet inn

Dersom vi utvider rekken med å addere flere delsummer vil vi se at rekken konvergerer mer og mer mot grensefunksjonen (på de negative x – verdiene) innenfor epsilonintervallet.

9. Vi adderer flere delsummer og observerer at konvergeringen mot e^x blir mer tydelig og som vi vet kan det alltid adderes en ny delsum.



Figur 51: Rekken utvikler seg med flere delsummer

Nå ser vi visuelt at rekkeuttrykket $\sum_{n=0}^{\infty} \frac{x^n}{n!}$ konvergerer mot e^x for alle x og forståelsen for hvordan rekken konvergerer er forhåpentligvis blitt større.

6 Pedagogiske implikasjoner

Nå som vi har et visualiseringsverktøy for funksjonsfølger og rekker har vi kanskje muligheten til å utvide våre kunnskaper på dette området på et tidligere tidspunkt. Vi kan illustrere og analysere om funksjonsfølger eller rekker konvergerer mot en ny funksjon g med definisjonsmengde A . Dette gjør at vi muligens kan innføre nye funksjoner, som ikke er elementære, og dermed åpne for flere områder innenfor matematikken på et tidligere stadium. Her er mulighetene mange.

Jeg mener at med dette programmet har vi et godt utgangspunkt for å forbedre forståelsen av grenser og konvergens hos studenter og elever. Det er mulig å se hva som skjer når vi adderer en ny delsum til en rekke. Dette kan hjelpe med at rekkeutviklinger blir mer forståelig som igjen kan bidra til en bedre forståelse av rekkekonseptet og hvorfor innføringen av rekker er nødvendig og praktisk.

Programmet gir også muligheten for at læreren selv kan legge inn funksjonsfølger eller rekkeuttrykk, som medfører en minst mulig risiko for feil i læringsobjektet som skal analyseres og forhåpentligvis gjør denne funksjonaliteten at det blir lettere å innføre programmet.

7 Referanser

- Alcock, L., & Simpson, A. (2004). Convergence of sequences and series: Interactions between visual reasoning and the learner's beliefs about their own role. *Educational Studies in Mathematics*, 57(1), 1-32.
- Alcock, L., & Simpson, A. (2005). Convergence of Sequences and Series 2: Interactions between visual reasoning and the learner's beliefs about their own role. *Educational Studies in Mathematics*, 58(1), 77-100.
- Beeson, M., & Wiedijk, F. (2005). The meaning of infinity in calculus and computer algebra systems. *Journal of Symbolic Computation*, 39(5), 523-538.
- Bezuidenhout, J. (2001). Limits and continuity: some conceptions of first-year students. *International Journal of Mathematical Education in Science & Technology*, 32(4), 487-500.
- Burd, B. (2005). *Beginning programming with Java™ for dummies®*, 2nd Edition. Indianapolis: Wiley Publishing, Inc.
- Horstmann, C. S., & Cornell, G. (2004). *Core Java™ 2 Volume I - Fundamentals, Seventh Edition*. San Francisco: Prentice Hall PTR.
- Jirotkova, D., & Littler, G. (2004). *Insight into pupils' understanding of infinity in a geometrical context*: International Group for the Psychology of Mathematics Education.
- Kidron, I., & Zehavi, N. (2002). The role of animation in teaching the limit concept. *The International Journal of Computer Algebra in Mathematics Education*, 9(205-227).
- Knudsen, J., & Niemeyer, P. (2005). *Learning Java, 3rd Edition*. Sebastopol: O'Reilly.
- Lorentzen, L., Hole, A., & Lindstrøm, T. L. (2003). *Kalkulus - med én og flere variable*. Oslo: Universitetsforl.
- Navarro, M. Á., & Carreras, P. P. (2006). Constructing a concept image of convergence of sequences in the van Hiele framework. *Research in Collegiate Mathematics Education*, VI, 61-98.
- Nygaard, O. (2007). Begreper i analyse. Kompendium Universitetet i Agder, from <http://home.uia.no/olavn/analysebegr.pdf>
- Pinto, M., & Tall, D. (2002). Building formal mathematics on visual imagery: A case study and a theory. *For the Learning of Mathematics*, 22(1), 2-10.
- Roh, K. H. (2008). Students' images and their understanding of definitions of the limit of a sequence. *Educational Studies in Mathematics*, 69(3), 217-233.
- Singer, F. M., & Voica, C. (2008). Between perception and intuition: Learning about infinity. *Journal of Mathematical Behavior*, 27(3), 188-205.
- Utdanningsdirektoratet. Læreplan i matematikk for realfag - programfag i studiespesialiserende utdanningsprogram, from <http://www.udir.no/grep/Utdanningsprogram/?utdprogrid=111534>

8 Vedlegg: Kildekode

Deler av kildekoden er lik for funksjonsfølger og for rekkeutvikling.

Vedlegg 8.1.1 til 8.1.9 er identiske for de to forskjellige programdelene.

8.1 Like klasser i programkoden til funksjonsfølger og rekkeutvikling.

8.1.1 Klassen: Advarsel

```
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

class Advarsel extends Frame implements ActionListener{

    //variable...
    HovedVindu hovedvindu;

    public Advarsel(String s, boolean helevindu){
        super("Advarsel!"); //navn
        if(helevindu){
            //advarselen går over hele vinduet
            Toolkit toolkit = Toolkit.getDefaultToolkit();
            Dimension dimension = toolkit.getScreenSize();
            setSize((int)dimension.getWidth(),(int)dimension.getHeight());
        } //end if
        else{
            //advarselen popper opp som et lite vindu
            Toolkit toolkit = Toolkit.getDefaultToolkit();
            Dimension dimension = toolkit.getScreenSize();
            setSize(200,150);
            setLocation((int)dimension.getWidth()-480,50);
        }

        //legger til panel for å få fekst
        Panel grunnPanel = new Panel();
        setBackground(new Color(255,255,255));
        add(grunnPanel);
        Panel panel = new Panel();
        panel.setLayout(new GridLayout(8, 2));
        setBackground(new Color(255,255,255));
        panel.add(new Label(s));

        Button godkjennVerdier = new Button("OK");
        Button avbryt = new Button("Avbryt");
        Color fargePaKnapp = new Color(200, 0, 100);
        godkjennVerdier.setBackground(fargePaKnapp);
        avbryt.setBackground(fargePaKnapp);
    }
}
```

```

panel.add(godkjennVerdier);
panel.add(avbryt);
    godkjennVerdier.addActionListener(this);
avbryt.addActionListener(this);
    grunnPanel.add(panel);
// grunnPanel.add(panel);

    //legger til vinduslister
HovedVindu hovedvindu = new HovedVindu(this);
addWindowListener(hovedvindu);
setVisible(true);

} //end DelingsVindu

    public void actionPerformed(ActionEvent actionevent){
//variable
Object obj = actionevent.getActionCommand();

if(obj=="OK"){
    dispose();
} //end if sett verdier
else if(obj=="Avbryt"){
    dispose();
} //end if Avbryt
else{
    //ingenting
} //end else
} //end actionPerformed

} //end

```

8.1.2 Klassen: Navnepanel

```

import java.awt.*;

class NavnPanel extends Panel{
    public NavnPanel(){
        //legger til navn mitt på høyresiden av designet...
        Panel grunnPanel = new Panel();
        grunnPanel.setLayout(new GridLayout(1,1));
        add(grunnPanel);
        Font skriftType = new Font("Verdana", 1, 14);
        Label lagetAv = new Label("Laget av Eirik Sørstrand", 0);
        lagetAv.setFont(skriftType);
        grunnPanel.add(lagetAv);
    } //end KnappPanel

} //end

```

8.1.3 Klassen: KnappPanel

```
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.EventObject;

class KnappPanel extends Panel implements ActionListener{

    //deklarerer variable
    private Button zoomInX;
    private Button zoomUtX;
    private Button zoomInY;
    private Button zoomUtY;
    private Button stillWindow;
    private Button rutenett;
    private Button tall;
    private Button hjelp;
    private Button nullstill;
    private Button ferdige;
    private Button nVerdiLik;
    private Button fjernFn;
    private Color fargePaKnapp;
    private String[] FunksjonsTekst;
    private String[] HjelpeTekst;
    private String[] funksjonsTekst;
    private String[] hjelpeTekst;
    protected InstillingsVindu instillingsVindu;
    protected HjelpVindu hjelpVindu;
    protected FerdigeFunksjoner ferdigeFunksjoner;
    protected StillNVerdiVindu stillNverdiVindu;
    protected FjernFnVindu fjernFnVindu;

    public KnappPanel(DelingsVindu delingsVindu,String[]FunksjonsTekst,String[]HjelpeTekst){
        //variable settes
        funksjonsTekst=FunksjonsTekst;
        hjelpeTekst=HjelpeTekst;
        zoomInX = new Button("Zoom inn x");
        zoomUtX = new Button("Zoom ut x");
        zoomInY = new Button("Zoom inn y");
        zoomUtY = new Button("Zoom ut y");
        stillWindow = new Button("Still vindu");
        rutenett = new Button("Slå av rutenett");
        tall = new Button("Fjerne tall");
        hjelp = new Button("Hjelp");
        nullstill = new Button("Nullstill");
        ferdige = new Button("Ferdige f(x)");
        nVerdiLik = new Button("N-verdi");
```

```

fjernFn = new Button("Fjern Fn?");
fargePaKnapp = new Color(200, 0, 100);

//grunnpanelet legges på
Panel grunnPanel = new Panel();
grunnPanel.setLayout(new GridLayout(6,2));
add(grunnPanel);

//farger settes på knapper
zoomInX.setBackground(fargePaKnapp);
zoomUtX.setBackground(fargePaKnapp);
zoomInY.setBackground(fargePaKnapp);
zoomUtY.setBackground(fargePaKnapp);
stillWindow.setBackground(fargePaKnapp);
ferdige.setBackground(fargePaKnapp);
rutenett.setBackground(fargePaKnapp);
hjelp.setBackground(fargePaKnapp);
tall.setBackground(fargePaKnapp);
nVerdiLik.setBackground(fargePaKnapp);
    fjernFn.setBackground(fargePaKnapp);

    //adder knappene til grunnpanelet
grunnPanel.add(zoomInX);
grunnPanel.add(zoomUtX);
grunnPanel.add(zoomInY);
grunnPanel.add(zoomUtY);
grunnPanel.add(rutenett);
grunnPanel.add(tall);
grunnPanel.add(stillWindow);
grunnPanel.add(ferdige);
grunnPanel.add(hjelp);
grunnPanel.add(nVerdiLik);
grunnPanel.add(fjernFn);
grunnPanel.add(nullstill);

//legger til ActionListener på knappene
zoomInX.addActionListener(this);
zoomUtX.addActionListener(this);
zoomInY.addActionListener(this);
zoomUtY.addActionListener(this);
rutenett.addActionListener(this);
tall.addActionListener(this);
hjelp.addActionListener(this);
nullstill.addActionListener(this);
stillWindow.addActionListener(this);
ferdige.addActionListener(this);
nVerdiLik.addActionListener(this);
fjernFn.addActionListener(this);
} //end KnappPanel

```

```

public void ferdige(){ //ferdige funksjoner vindu
    ferdigeFunksjoner = new FerdigeFunksjoner("Ferdige funksjoner",funksjonsTekst);
}

//end ferdige

public void stillNverdi(String valg){ //still n-verdi-vindu
    stillNverdiVindu = new StillNVerdiVindu("N-verdi");
}

//end stillNVerdi

    public void hjelp(String valg){ //hjelp knapp
        hjelpVindu = new HjelpVindu("Hjelp",hjelpetekst);
}

//end hjelp

    public void vindu(String valg){ //sette x-min, x-max o.s.v
instillingsVindu = new InstillingsVindu("Still Vindu",250,250);
}

//end vindu

    public void fjern(){ //fjerne funksjonen fn...
fjernFnVindu = new FjernFnVindu("Fjern gamle delsummer fra koordinatsystemet?");
}

//end vindu

    public void tall(String valg){ //slå av tall knapp
        if(valg == "Fjerne tall"){
            tall.setLabel("Tilføy tall");
            DelingsVindu.tegneVindu.tallVekk(true);
}

//end if
else{
    tall.setLabel("Fjerne tall");
    DelingsVindu.tegneVindu.tallVekk(false);
}

//end else if rutenett
}

//end tall

    public void rutenett(String valg){ //slå av rutenett knapp
        if(valg == "Slå av rutenett"){
            rutenett.setLabel("Slå på rutenett");
            DelingsVindu.tegneVindu.rutenettVekk(true);
}

//end if
else{
    rutenett.setLabel("Slå av rutenett");
    DelingsVindu.tegneVindu.rutenettVekk(false);
}

//end else if rutenett
}

//end rutenett

    public void zoom(ActionEvent actionevent){ //zoomknapper
        //variable
        Object obj = actionevent.getSource();

        if(obj == zoomInX){
            DelingsVindu.tegneVindu.xskala *=0.5;

```

```

        DelingsVindu.tegneVindu.repaint();
    }//end if
    else if(obj == zoomUtX){
        DelingsVindu.tegneVindu.xskala *=2;
        DelingsVindu.tegneVindu.repaint();
    }//end if
    if(obj == zoomInY){
        DelingsVindu.tegneVindu.yskala *=0.5;
        DelingsVindu.tegneVindu.repaint();//
    }//end if
        if(obj == zoomUtY){
            DelingsVindu.tegneVindu.yskala *=2;
            DelingsVindu.tegneVindu.repaint();//
        }//end if
    }//end rutenett

    public void nullstill(){
        DelingsVindu.utviklingPanel.nullstill();
        DelingsVindu.tegneVindu.nullstill();

    }//end nullstill

public void actionPerformed(ActionEvent actionevent){
    //variable
    Object obj = actionevent.getSource();

    if(obj==nullstill){
        nullstill();
    }//end nullstill
    else if(obj == rutenett){
        rutenett(rutenett.getLabel());
    }//end rutenett
    else if(obj == ferdige){
        ferdige();
    }//end ferdige funksjoner
    else if(obj == tall){
        tall(tall.getLabel());
    }//end tall
    else if(obj ==hjelp){
        hjelp(hjelp.getLabel());
    }//end help
    else if(obj==stillWindow){
        vindu(stillWindow.getLabel());
    }//end stillwindow
    else if(obj==nVerdiLik){
        stillNverdi(stillWindow.getLabel());
    }//end nverdiLik
    else if(obj==fjernFn){
        fjern();
    }

```

```

    }//end nverdiLik
    else{ //zoom
        zoom(actionevent);
    }//end zoom
} //end ActionPerformed

} //end

```

8.1.4 Klassen: FerdigeFunksjoner

```

import java.awt.*;
import java.io.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

```

```

class FerdigeFunksjoner extends Frame implements ActionListener{

```

```

    //variable deklarasjon
    private int width;
    private int height;
    HovedVindu hovedvindu;
    private int antallLinjerIFil=0;
    private int antallTegn;
    private String[] funksjonsTekst;
    private String[] funksjonsTekstTilButton;
    private String[] funksjonsTekstTilUttrykk;
    private Button[] funksjonButton;

    public FerdigeFunksjoner(String s,String[] FunksjonsTekst){
        //lager et vindu for å få frem automatiske funksjoner som ligger i filen ferdigeFunksjoner.txt
        //blir lest inn i funksjonsutvikling.java
        super(s);//setter tittel
        Toolkit toolkit = Toolkit.getDefaultToolkit();
        Dimension dimension = toolkit.getScreenSize();
        antallTegn=0;
        antallLinjerIFil=FunksjonsTekst.length;
        funksjonsTekst=FunksjonsTekst;
        if(funksjonsTekst.length==0){
            //fikk ikke lest inn noe fra ferdigeFunksjoner.txt
            width = 350;
            height = 70;
            setSize(width, height);
        } //end if
        else{
            width = 250;
            height = 40+30*antallLinjerIFil;
            setSize(width, height);
            funksjonButton = new Button[antallLinjerIFil];

```

```

        funksjonsTekstTilButton = new String[antallLinjerIFil];
        funksjonsTekstTilUttrykk = new String[antallLinjerIFil];
        skilleButtonUttrykk(); //rutine for å skille tekst til knapp og til funksjonsuttrykket
        panelinnstillinger();
    } //end else

    //legger til vinduslistener
    HovedVindu hovedvindu = new HovedVindu(this);
    addWindowListener(hovedvindu);
    setVisible(true);

} //end DelingsVindu

public void skilleButtonUttrykk(){
    //skille tekst til button og til funksjonsuttrykket
    for(int i=0;i<antallLinjerIFil;i++){ //sjekke hver linje
        for(int j=0;j<funksjonsTekst[i].length();j++){ //sjekke hver bokstav til =
            if(funksjonsTekst[i].charAt(j)=='='){
                funksjonsTekstTilButton[i]=funksjonsTekst[i].substring(0,j);

                funksjonsTekstTilUttrykk[i]=funksjonsTekst[i].substring(j+1,funksjonsTekst[i].length());
            } //end if == '='
        } //end j-forløkke
    } //end i-forløkke
} //end skilleButtonUttrykk

public void panelinnstillinger(){
    //stiller innstillinger for panelet
    Panel grunnPanel = new Panel();
    setBackground(new Color(255,255,255));
    Panel panel = new Panel();
    panel.setLayout(new GridLayout(antallLinjerIFil-1,2));
    setBackground(new Color(255,255,255));
    Color fargePaKnapp = new Color(200, 0, 100);

    for(int i=0;i<antallLinjerIFil;i++){ //legger til knappene i panelet
        funksjonButton[i]=new Button(""+funksjonsTekstTilButton[i]);
        funksjonButton[i].addActionListener(this);
        funksjonButton[i].setBackground(fargePaKnapp);
        panel.add(funksjonButton[i]);
    } //end i-forløkke
    grunnPanel.add(panel);
    add(grunnPanel);
} //end panelinnstillinger

    public void actionPerformed(ActionEvent actionevent){
//variable
        Object obj = actionevent.getActionCommand();

```

```

        for(int i=0;i<antallLinjerIFil;i++){
            if(funksjonsTekstTilButton[i].contains(""+obj)){
                DelingsVindu.utviklingPanel.funksjonsUttrykk.setText(funksjonsTekstTilUttrykk[i]);
                dispose();
            }
        }
    }
}

public void paint(Graphics g){
    if(funksjonsTekst.length==0) g.drawString("Ingen funksjoner er lagt til, kontakt
programansvarlig",10,50);
}

}

```

8.1.5 Klassen: HjelpVindu

```

import java.awt.*;
import java.io.*;

class HjelpVindu extends Frame {

    //variable deklarasjon
    private int width;
    private int height;
    HovedVindu hovedvindu;
    private String[] hjelpeTekst; //hjelpeteksten som er blitt lest inn fra filen hjelp.txt skal inn i denne
    tabellen
    private int antallLinjerIFil=0; //hvor mange filer i hjelp.txt
    private int antallTegn; //hvor mange tegn den lengste filen har

    public HjelpVindu(String s,String[] HjelpeTekst){
        //lager et vindu med hjelpeinformasjon til brukeren...
        super(s); //navn

        hjelpeTekst=HjelpeTekst; //overfører innlesningen
        Toolkit toolkit = Toolkit.getDefaultToolkit();
        Dimension dimension = toolkit.getScreenSize();
        antallTegn=0;

        for(int i=0;i<hjelpeTekst.length;i++){ //finner ut hvor mange bokstaver den lengste linjen har for å
        stille vindu
            if(antallTegn<hjelpeTekst[i].length())antallTegn=hjelpeTekst[i].length();
        }
        if(hjelpeTekst.length==0){ //ingen tegn fra lest fil.
            width = 350;
            height = 70;
        }
    }
}

```

```

        setSize(width, height);
    } //end length==0
    else{ //setter bredde og høyde ut fra antall linjer og antall bokstaver
        width = 10+antallTegn*6;
        height = 45+22*hjelpetekst.length;
        setSize(width, height);
    } //end else

    HovedVindu hovedvindu = new HovedVindu(this);
    addWindowListener(hovedvindu);
    setVisible(true);

} //end DelingsVindu

public void paint(Graphics g){
    g.setColor(Color.WHITE);
    g.fillRect(0,0,width,height);
    g.setColor(Color.BLACK);

    for (int i=0;i<hjelpetekst.length;i++){
        //tegner opp hjelpeteksten linje for linje
        g.drawString(""+hjelpetekst[i],10,45+20*i);
    } //end i-forløkke
    if(hjelpetekst.length==0) g.drawString("Ingen hjelp er lagt til, kontakt programansvarlig",10,50);

} //end paint

public void update(Graphics g){
    paint(g);
} //end update

} //end

```

8.1.6 Klassen: InnstillingsVindu

```
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

class InnstillingsVindu extends JFrame implements ActionListener{

    //variable deklarasjon
    private int width;
    private int height;
    //diverse tekstfelt
    private TextField xminTF;
    private TextField yminTF;
    private TextField xmaxTF;
    private TextField ymaxTF;
    private TextField xIntervallTF;
    private TextField yIntervallTF;
    private TextField xMinTekst;
    private TextField yMinTekst;
    private TextField xMaxTekst;
    private TextField yMaxTekst;
    private TextField xIntervallTekst;
    private TextField yIntervallTekst;
    private Button godkjennVerdier;
    private Button avbryt;
    private Color fargePaKnapp;
    HovedVindu hovedvindu;
    private String ulovlig; //string med ulovlige tegn
        private char ulovligTegn[]; //tabell som inneholder de ulovlige tegnene

    public InnstillingsVindu(String s,int w, int h){
        super(s); //setter navn på vinduet

        Toolkit toolkit = Toolkit.getDefaultToolkit();
        Dimension dimension = toolkit.getScreenSize();
        width = w;
        height = h;
        setSize(width, height);
        setLocation(dimension.width-2*w,0);

        ulovlig = new String
("AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZzÆæØøÅå<>:;?+*/^)&%# $"); //ulovlige
tegn...

        ulovligTegn = new char[ulovlig.length()];
        for(int i=0;i<ulovlig.length();i++){
            //legger inn de ulovlige tegnene fra stringen til tabellen
            ulovligTegn[i]=ulovlig.charAt(i);
        }
    }
}
```

```

    }

    //
    Panel grunnPanel = new Panel();
setBackground(new Color(255,255,255));
add(grunnPanel);

    Panel panel = new Panel();
    panel.setLayout(new GridLayout(7, 2));
setBackground(new Color(255,255,255));

//de forskjellige instillingene for tegnevinduet
panel.add(new Label("X-Min :"));
xMinTekst = new TextField("", 10);
xMinTekst.setText(""+DelingsVindu.tegneVindu.xMin);
xminTF = new TextField("", 10);
xminTF.setText(""+DelingsVindu.tegneVindu.xMin);
panel.add(xminTF);

panel.add(new Label("X-Max :"));
xMaxTekst = new TextField("", 10);
xMaxTekst.setText(""+DelingsVindu.tegneVindu.xMax);
xmaxTF = new TextField("",10);
xmaxTF.setText(""+DelingsVindu.tegneVindu.xMax);
panel.add(xmaxTF);

    panel.add(new Label("X-intervall:"));
    xIntervallTekst = new TextField("", 10);
    xIntervallTekst.setText(""+(double)((int)(DelingsVindu.tegneVindu.xskala*10000))/10000);
xIntervallTF = new TextField("",10);
xIntervallTF.setText(""+(double)((int)(DelingsVindu.tegneVindu.xskala*10000))/10000);
panel.add(xIntervallTF);

panel.add(new Label("Y-Min :"));
yMinTekst = new TextField("", 10);
yMinTekst.setText(""+DelingsVindu.tegneVindu.yMin);
yminTF = new TextField("", 10);
yminTF.setText(""+DelingsVindu.tegneVindu.yMin);
panel.add(yminTF);

panel.add(new Label("Y-Max :"));
yMaxTekst = new TextField("", 10);
yMaxTekst.setText(""+DelingsVindu.tegneVindu.yMax);
ymaxTF = new TextField("", 10);
ymaxTF.setText(""+DelingsVindu.tegneVindu.yMax);
panel.add(ymaxTF);

panel.add(new Label("Y-intervall:"));
yIntervallTekst = new TextField("", 10);

```

```

yIntervallTekst.setText(""+(double)((int)(DelingsVindu.tegneVindu.yskala*10000))/10000);
yIntervallTF = new TextField("", 10);
yIntervallTF.setText(""+(double)((int)(DelingsVindu.tegneVindu.yskala*10000))/10000);
panel.add(yIntervallTF);

//buttons
Button godkjennVerdier = new Button("Sett verdier");
Button avbryt = new Button("Avbryt");
fargePaKnapp = new Color(200, 0, 100);
godkjennVerdier.setBackground(fargePaKnapp);
avbryt.setBackground(fargePaKnapp);
panel.add(avbryt);
panel.add(godkjennVerdier);
grunnPanel.add(panel);
    godkjennVerdier.addActionListener(this);
    avbryt.addActionListener(this);

    //legger til window listener
    HovedVindu hovedvindu = new HovedVindu(this);
    addWindowListener(hovedvindu);
    setVisible(true);
} //end DelingsVindu

public void actionPerformed(ActionEvent actionevent){
    //variable
    Object obj = actionevent.getActionCommand();
    //henter x-verdiene
    String tempXmin= new String(ulovligVerdi(xminTF.getText()));
    String tempXmax= new String(ulovligVerdi(xmaxTF.getText()));
    String tempXintervall= new String(ulovligVerdi(xIntervallTF.getText()));
    //henter y-verdiene
    String tempYmin= new String(ulovligVerdi(yminTF.getText()));
    String tempYmax= new String(ulovligVerdi(ymaxTF.getText()));
    String tempYintervall= new String(ulovligVerdi(yIntervallTF.getText()));

    //valget som er gjort
    if(obj=="Sett verdier"){
        if(DelingsVindu.tegneVindu.gjorTilTall(tempXmax)<=DelingsVindu.tegneVindu.gjorTilTall(tempXmin)){
            Advarsel a = new Advarsel("X-max mindre enn X-min",false);
            a.show();
        } //end else if tempXmax<tempXmin
        else
            if(DelingsVindu.tegneVindu.gjorTilTall(tempYmax)<=DelingsVindu.tegneVindu.gjorTilTall(tempYmin)){
                Advarsel a = new Advarsel("Y-max mindre enn Y-min",false);
                a.show();
            } //end else if tempYmax<tempYmin
        else{
            DelingsVindu.utviklingPanel.infoFeltTF.setText("Tegner opp graf");
        }
    }
}

```

```

//sender verdiene til sett origo for å kalkulere hvor origo skal opprettes

DelingsVindu.tegneVindu.settOrigo(tempXmin,tempXmax,tempXintervall,tempYmin,tempYmax,temp
Yintervall);
    DelingsVindu.tegneVindu.repaint();
    dispose();
} //end else

} //end if sett verdier
else if(obj=="Avbryt"){
    DelingsVindu.tegneVindu.repaint();
    dispose();
} //end if Avbryt
else{
    //ingenting
} //end else
} //end actionPerformed

public String ulovligVerdi(String sjekk){

    for(int i=0;i<sjekk.length();i++){
        for(int j=0;j<ulovlig.length();j++){
            if(sjekk.charAt(i)==ulovligTegn[j]){
                if(sjekk.charAt(i)=='P' || sjekk.charAt(i)=='p'){

                    if(i<sjekk.length()-1){
                        if(sjekk.charAt(i+1)=='l' || sjekk.charAt(i+1)=='i'){

                            if(i==0){

                                sjekk=DelingsVindu.tegneVindu.gangeDele(Math.PI+sjekk.substring(i+2,sjekk.length()));
                                }
                                else{
                                    if(sjekk.charAt(i-
1)=='*')sjekk=DelingsVindu.tegneVindu.gangeDele(sjekk.subSequence(0,i)+(""+Math.PI)+sjekk.substring(i+2,sje
kk.length()));

                                else
                                sjekk=DelingsVindu.tegneVindu.gangeDele(sjekk.subSequence(0,i)+(""+Math.PI)+sjekk.substring(i+2,sjekk.len
gth()));

                                }

                                j=0;
                                i=0;
                            } //end sjekk l
                        } //end sjekk length -1
                    } //end sjekk ==p
                else{
                    sjekk=(sjekk.substring(0,i)+sjekk.substring(i+1,sjekk.length()));
                    j=0;

```

```

        i=0;
    } //end else

} //end if
if(sjekk.charAt(i)==' ') {
    if(i==sjekk.length()-1){

        sjekk=sjekk.substring(0,i);

        j=0;
        i=0;
    }
    else {

        sjekk=sjekk.substring(0,i)+"."+sjekk.substring(i+1,sjekk.length());
        j=0;
        i=0;
    }
}
} //enf j-forløkke
} //end i-forløkke
return sjekk;
} //end ulovig verdi

} //end

```

8.1.7 Klassen: DelingsVindu

```

import java.awt.*;

class DelingsVindu extends Frame {

    //variable deklarasjon
    protected int width;
    protected int height;
    static TegneVindu tegneVindu;
    static UtviklingPanel utviklingPanel;
    static KnappPanel KnappPanel;
    static NavnPanel navnPanel;

    public DelingsVindu(String s,String[] FunksjonsTekst,String[] HjelpeTekst){
        //deler inn vinduet i de 4 delene... tegne funksjon, laget av, knapper og utviklingen

        super(s);
        Toolkit toolkit = Toolkit.getDefaultToolkit();
        Dimension dimension = toolkit.getScreenSize();
        width = dimension.width;
        height = dimension.height - 30;
        setSize(width, height);
    }
}

```

```
tegneVindu = new TegneVindu(width, height);
KnappPanel = new KnappPanel(this,FunksjonsTekst,HjelpeTekst);
navnPanel = new NavnPanel();
utviklingPanel = new UtviklingPanel();
setLayout(new GridBagLayout());
```

```
GridBagConstraints tegneVinduInnstillinger = new GridBagConstraints();
tegneVinduInnstillinger.gridx = 0;
    tegneVinduInnstillinger.gridy = 0;
tegneVinduInnstillinger.gridwidth = 5;
tegneVinduInnstillinger.gridheight = 4;
tegneVinduInnstillinger.fill = 1;
tegneVinduInnstillinger.insets = new Insets(1, 1, 1, 1);
tegneVinduInnstillinger.weightx = 1;
    tegneVinduInnstillinger.weighty = 90;
add(tegneVindu, tegneVinduInnstillinger);
```

```
GridBagConstraints navnVinduInnstillinger = new GridBagConstraints();
navnVinduInnstillinger.gridx = 5;
navnVinduInnstillinger.gridy = 0;
navnVinduInnstillinger.gridwidth = 0;
navnVinduInnstillinger.gridheight = 1;
navnVinduInnstillinger.fill = 1;
navnVinduInnstillinger.insets = new Insets(1, 1, 1, 1);
navnVinduInnstillinger.weightx = 0;
navnVinduInnstillinger.weighty = 0;
add(navnPanel, navnVinduInnstillinger);
```

```
GridBagConstraints zoomVindu = new GridBagConstraints();
zoomVindu.gridx = 5;
zoomVindu.gridy = 1;
zoomVindu.gridwidth = 0;
zoomVindu.gridheight = 1;
zoomVindu.fill = 1;
zoomVindu.insets = new Insets(1, 1, 1, 1);
zoomVindu.weightx = 0;
zoomVindu.weighty = 0;
add(KnappPanel, zoomVindu);
```

```
GridBagConstraints utviklinVindu = new GridBagConstraints();
utviklinVindu.gridx = 5;
utviklinVindu.gridy = 3;
utviklinVindu.gridwidth = 0;
utviklinVindu.gridheight = 1;
utviklinVindu.fill = 1;
utviklinVindu.insets = new Insets(1, 1, 1, 1);
utviklinVindu.weightx = 0;
utviklinVindu.weighty = 0;
add(utviklingPanel, utviklinVindu);
```

```

//legger til vindu handler...
    HovedVindu hovedvindu = new HovedVindu(this);
    addWindowListener(hovedvindu);
    setVisible(true);
} //end DelingsVindu
} //end

```

8.1.8 Klassen: Hovedvindu

```

import java.awt.*;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;

class HovedVindu extends WindowAdapter {

    //variable      WindowAdapter
    private DelingsVindu vinduDelingsVindu;
    private InstillingsVindu vinduInstillingsVindu;
    private HjelpVindu vinduHjelpVindu;
    private FerdigeFunksjoner vinduFerdigeFunksjoner;
    private StillNVerdiVindu vinduStillNVerdi;
    private FjernFnVindu vinduFjernFnVindu;
    private GjettGrenseFunksjonsVindu vinduGjettGrenseVindu;
    private Advarsel vinduAlarm;

    public HovedVindu(DelingsVindu delingsVindu){
        vinduDelingsVindu = delingsVindu;
    } // end HovedVindu

    public HovedVindu(Advarsel vindualarm){
        vinduAlarm = vindualarm;
    } // end HovedVindu

    public HovedVindu(GjettGrenseFunksjonsVindu gjettGrenseVindu){
        vinduGjettGrenseVindu = gjettGrenseVindu;
    } // end HovedVindu

    public HovedVindu(InstillingsVindu instillingsVindu){
        vinduInstillingsVindu = instillingsVindu;
    } // end HovedVindu

    public HovedVindu(HjelpVindu hjelpVindu){
        vinduHjelpVindu = hjelpVindu;
    } // end HovedVindu

    public HovedVindu(FerdigeFunksjoner ferdigeFunksjoner){
        vinduFerdigeFunksjoner = ferdigeFunksjoner;
    }

```

```

} // end HovedVindu

public HovedVindu(FjernFnVindu fjernFnVindu){
    vinduFjernFnVindu = fjernFnVindu;
} // end HovedVindu

public HovedVindu(StillNVerdiVindu stillNVerdiVindu){
    vinduStillNVerdi = stillNVerdiVindu;
} // end HovedVindu

public void windowClosing(WindowEvent windowevent){
    //handling på lukke vindu
    if(windowevent.getWindow().getClass().getName()=="InstillingsVindu") {
        vinduInstillingsVindu.dispose();
    } //end if InstillingsVindu
    if(windowevent.getWindow().getClass().getName()=="HjelpVindu") {
        vinduHjelpVindu.dispose();
    } //end if HjelpVindu
    if(windowevent.getWindow().getClass().getName()=="FerdigeFunksjoner") {
        vinduFerdigeFunksjoner.dispose();
    } //end if FerdigeFunksjoner
    else if(windowevent.getWindow().getClass().getName()=="DelingsVindu") {
        vinduDelingsVindu.dispose();
    } //end if delingsvindu
    else if(windowevent.getWindow().getClass().getName()=="StillNVerdiVindu") {
        vinduStillNVerdi.dispose();
    } //end if delingsvindu
    else if(windowevent.getWindow().getClass().getName()=="FjernFnVindu") {
        vinduFjernFnVindu.dispose();
    } //end if delingsvindu
    else if(windowevent.getWindow().getClass().getName()=="GjettGrenseFunksjonsVindu") {
        vinduGjettGrenseVindu.dispose();
    } //end if delingsvindu
    else if(windowevent.getWindow().getClass().getName()=="Advarsel") {
        vinduAlarm.dispose();
    } //end if delingsvindu
    else {
        //ingenting...
    } //end else
} //end windowClosing

} //end

```

8.1.9 Klassen: GjettGrenseFunksjonsVindu

```
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

class GjettGrenseFunksjonsVindu extends Frame implements ActionListener{

    //variable deklarasjon
    private int width;
    private int height;
    private TextField grenseVerdiTF;
    private TextField epsilonTF;
    private Button godkjennVerdier;
    private Button avbryt;
    private Color fargePaKnapp;
    HovedVindu hovedvindu;
    private String ulovlig; //string med ulovlige tegn
        private char ulovligTegn[]; //tabllen med de ulovlige tegnene

    public GjettGrenseFunksjonsVindu(String s){
        super(s); //setter navn på vinduet

        Toolkit toolkit = Toolkit.getDefaultToolkit();
        Dimension dimension = toolkit.getScreenSize();

        setSize(350, 200);
        setLocation(dimension.width-300,0);

        //legger til panelene
        Panel grunnPanel = new Panel();
        setBackground(new Color(255,255,255));
        add(grunnPanel);
        Panel panel = new Panel();
        panel.setLayout(new GridLayout(7, 2));
        setBackground(new Color(255,255,255));

        //legger til gjett og epsilon verdi felt
        panel.add(new Label("Gjett grensefunksjon :"));
        grenseVerdiTF = new TextField(""+DelingsVindu.tegneVindu.grensefunksjon, 30);
        panel.add(grenseVerdiTF);

        panel.add(new Label("Sett epsilon verdi"));
        epsilonTF = new TextField(""+DelingsVindu.tegneVindu.epsilonVerdi, 10);
```

```

panel.add(epsilonTF);

//buttons
Button godkjennVerdier = new Button("OK");
Button avbryt = new Button("Avbryt");
fargePaKnapp = new Color(200, 0, 100);
godkjennVerdier.setBackground(fargePaKnapp);
avbryt.setBackground(fargePaKnapp);

panel.add(godkjennVerdier);
panel.add(avbryt);
grunnPanel.add(panel);
    godkjennVerdier.addActionListener(this);
avbryt.addActionListener(this);

ulovlig = new String ("BbDdFfHhJjKkMmQqRrUuVvWwYyZzÆæØøÅå<>:;?&%#|'$"); //ulovlige tegn...
    ulovligTegn = new char[ulovlig.length()];
    for(int i=0;i<ulovlig.length();i++){
        ulovligTegn[i]=ulovlig.charAt(i);//legger inn de ulovlige tegnene i tabellen
    }

    //legger til window listener
HovedVindu hovedvindu = new HovedVindu(this);
addWindowListener(hovedvindu);
setVisible(true);
} //end DelingsVindu

public String ulovligVerdi(String sjekk){
//sjekker etter ulovlige tegn og fjerner dem.
    for(int i=0;i<sjekk.length();i++){
        for(int j=0;j<ulovlig.length();j++){
            if(sjekk.charAt(i)=='P' | sjekk.charAt(i)=='p'){
                if(i<sjekk.length()-1){
                    if(sjekk.charAt(i+1)=='|' | sjekk.charAt(i+1)=='i'){
                        if(i==0){

                            sjekk="("+Math.PI+")"+sjekk.substring(i+2,sjekk.length());
                                //end if ==0
                            else{

sjekk=sjekk.subSequence(0,i)+"("+Math.PI+")"+sjekk.substring(i+2,sjekk.length());
                                //end else
                                j=0;
                                i=0;
                            } //end sjekk I
                        } //end sjekk length -1
                    } else if(i==sjekk.length()-1){
                        //nothing
                    }
                }
            }
        }
    }
}

```

```

        }//end else if length-1
    }//end sjekk ==p
    else if(sjekk.charAt(i)=='e'){//erstatte bokstaven e med 2.718281828.

    sjekk=sjekk.subSequence(0,i)+(" "+Math.E)+" "+sjekk.substring(i+1,sjekk.length());
        }//end else == 'e'
        else if(sjekk.charAt(i)=='ulovligTegn[j]){

    sjekk=(sjekk.substring(0,i)+sjekk.substring(i+1,sjekk.length()));
        j=0;
        i=0;

        }//end if
        if(sjekk.charAt(i)==',') {
            if(i==sjekk.length()-1){

                sjekk=sjekk.substring(0,i);

                j=0;
                i=0;
            }//end length -1
            else {

    sjekk=sjekk.substring(0,i)+"."+sjekk.substring(i+1,sjekk.length());
                j=0;
                i=0;
            }//end else
        }//end == ,
    }//enf j-forløkke
} //end i-forløkke

    return sjekk;
} //end ulovlig verdi

```

```

public void actionPerformed(ActionEvent actionevent){
    //variable
    Object obj = actionevent.getActionCommand();

    if(obj=="OK"){
        DelingsVindu.utviklingPanel.infoFeltTF.setText("Tegner opp graf");
        DelingsVindu.tegneVindu.grensefunksjon=ulovligVerdi(grenseVerdiTF.getText());
        if(DelingsVindu.tegneVindu.grensefunksjon.length()>0){
            DelingsVindu.tegneVindu.tegnGrensefunksjon=true;
            DelingsVindu.tegneVindu.tegnEpsilon=true;

            DelingsVindu.tegneVindu.epsilonVerdi=DelingsVindu.tegneVindu.gjørTilTall(epsilonTF.getText());
            DelingsVindu.tegneVindu.repaint();
        } //end if
    }
}

```

```

        dispose();
    } //end if OK
    else if(obj=="Avbryt"){
        DelingsVindu.tegneVindu.repaint();
        dispose();
    } //end if Avbryt
    else{
        //ingenting
    } //end else
} //end actionPerformed

} //end

```

8.2 Funksjonsfølger

Under følger de klassene til funksjonsfølger som er forskjellige fra rekkeutviklingen.

8.2.1 Klassen: FjernFnVindu

```

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

class FjernFnVindu extends Frame implements ActionListener{
    //variable...
    private int width;
    private int height;
    private Button[] funksjonButton;
    private Button nullstill;
    private Button OK;
    private Button avbryt;
    private Color fargePaKnapp;

    public FjernFnVindu(String s){
        super(s);
        Toolkit toolkit = Toolkit.getDefaultToolkit();
        Dimension dimension = toolkit.getScreenSize();
        if(DelingsVindu.utviklingPanel.hvilkenPosisjon<=0){
            width = 400;
            height = 70;
            setSize(width, height);
        } //end if
        else{
            int tempLength=1;
            for(int i=0;i<DelingsVindu.utviklingPanel.hvilkenPosisjon;i++){

                if(tempLength<DelingsVindu.utviklingPanel.uttrykk[i].getText().length())tempLength=DelingsVindu.utviklingPanel.uttrykk[i].getText().length();
            } //end i-for
        }
    }
}

```

```

        width = 230+tempLength;
        height = 40+DelingsVindu.utviklingPanel.antallrekker*30;
        setSize(width, height);
        funksjonButton = new Button[DelingsVindu.utviklingPanel.antallrekker];
        panelinnstillinger();
    }//end else

    //legger til vinduslister
    HovedVindu hovedvindu = new HovedVindu(this);
    addWindowListener(hovedvindu);
    setVisible(true);

} //end DelingsVindu

public void panelinnstillinger(){
    //stiller innstillinger for panelet
    Panel grunnPanel = new Panel();
    setBackground(new Color(255,255,255));
    Panel panel = new Panel();
    panel.setLayout(new GridLayout(DelingsVindu.utviklingPanel.antallrekker+3,3));
    setBackground(new Color(255,255,255));

    for(int i=0;i<DelingsVindu.utviklingPanel.antallrekker;i++){ //legger til knappene i panelet
        funksjonButton[i]=new Button(""+DelingsVindu.utviklingPanel.uttrykk[i].getText());
        funksjonButton[i].addActionListener(this);
        funksjonButton[i].setBackground(DelingsVindu.utviklingPanel.fargeKoder[i]);
        if(i>=DelingsVindu.utviklingPanel.hvilkenPosisjon)funksjonButton[i].setVisible(false);
        panel.add(funksjonButton[i]);
    } //end i-forløkke
    nullstill = new Button ("Vis alt");
    nullstill.addActionListener(this);
    OK = new Button("OK");
    avbryt = new Button("Avbryt");
    fargePaKnapp = new Color(200, 0, 100);
    OK.setBackground(fargePaKnapp);
    avbryt.setBackground(fargePaKnapp);

    grunnPanel.add(panel);
    OK.addActionListener(this);
    avbryt.addActionListener(this);
    panel.add(nullstill);
    panel.add(OK);
    panel.add(avbryt);
    grunnPanel.add(panel);
    add(grunnPanel);
} //end panelinnstillinger

public void stillTekst(){

```

```

for(int i=0;i<DelingsVindu.utviklingPanel.antallrekker;i++){ //legger til knappene i panelet
    funksjonButton[i].setLabel(""+DelingsVindu.utviklingPanel.uttrykk[i].getText());
    funksjonButton[i].setBackground(DelingsVindu.utviklingPanel.fargeKoder[i]);
    if(i>=DelingsVindu.utviklingPanel.hvilkenPosisjon){
        funksjonButton[i].setVisible(false);
    }//end if
    else{
        //viser nye knapper og skjuler de som er trykket på fra før...
        if(DelingsVindu.tegneVindu.tegneUttrykkTrue[i]) funksjonButton[i].setVisible(true);
        else funksjonButton[i].setVisible(false);
    }//end else
} //end i-forløkke
} //end stilltekst

```

```

public void actionPerformed(ActionEvent actionevent){
    Object obj = actionevent.getActionCommand();

    if(obj=="Vis alt"){
        for(int i=0;i<DelingsVindu.utviklingPanel.hvilkenPosisjon;i++){
            funksjonButton[i].setVisible(true);
            DelingsVindu.tegneVindu.tegneUttrykkTrue[i]=true;
        } //end i-forløkke
        DelingsVindu.tegneVindu.repaint();
    } //end if

    if(obj=="OK"){
        DelingsVindu.tegneVindu.repaint();
        dispose();

    } //end if ok

    else if(obj=="Avbryt"){
        DelingsVindu.tegneVindu.repaint();
        dispose();
    } //end if Avbryt

    else{
        for(int i=0;i<DelingsVindu.utviklingPanel.hvilkenPosisjon;i++){
            if(funksjonButton[i].isVisible()){
                if(funksjonButton[i].getLabel().equals(""+obj)){
                    funksjonButton[i].setVisible(false);
                    DelingsVindu.tegneVindu.tegneUttrykkTrue[i]=false;

                } //end if
            } else{
                //ingenting
            } //end else
        } //end if
    } //end i-for
    DelingsVindu.tegneVindu.repaint();
}

```

```

        }//end else
    }//end actionPerformed

    public void paint(Graphics g){
        if(DelingsVindu.utviklingPanel.hvilkenPosisjon<=0){
            g.drawString("Ingen uttrykk er tegnet opp derfor kan ingen fjernes.",20,50);
        }//end if
        else{
            stillTekst();
        }//end else
    }//end paint

    public void update(Graphics g){
        paint(g);
    }//end update

} //end

```

8.2.2 Klassen: TegneVindu

```

import java.awt.*;

class TegneVindu extends Panel {
    //variable
    private boolean tallPa;
    private boolean rutenettPa;
    protected int width;
    protected int height;
    protected double xskala;
    protected double yskala;
    private Image image;
    protected double Xintervall;//lengden mellom x-koordinatsystem
    protected double Yintervall;//lengden mellom y-koordinatsystem

    private int antallUttrykk;
    protected String[] tegneUttrykk;
    protected boolean[] tegneUttrykkTrue;
    protected int hvilketUttrykk=-3;
    private boolean Infinity=false;//skal punkt (x,y) tegnes?
    private boolean ForrigeInfinity=false; //variabelk for å huske om forrige x,y punkt skulle tegnes. ellers tegn
    opp fra samme verdi
    protected double xMin;
    protected double xMax;
    protected double yMin;
    protected double yMax;
    private int xNed=0;//zoom linjger når man trykker ned musen i koordinatsystemet
    private int xOpp=0;//zoom linjger når man trykker ned musen i koordinatsystemet

```

```

private int yNed=0;//zoom linjer når man trykker ned musen i koordinatsystemet
private int yOpp=0;//zoom linjer når man trykker ned musen i koordinatsystemet
protected int origoX;//origokoordinat
protected int origoY;//origokoordinat
private boolean start; // for å sette koordinatsystemet første gang...
private boolean fakultetParantes; //skal styre om det er negativ fakultet
private boolean negativParantes;
private int gammelHoyde;
private int gammelBredde;
private boolean negativFakultet;
boolean potensParantes;
protected String grensefunksjon;
protected boolean tegnGrenseFunksjon;
    protected boolean tegnEpsilon;
    protected double epsilonVerdi;
    private double yPlussEpsilon;
    private double yMinusEpsilon;
    protected int antallRekkerSomSkalVises;

    public TegneVindu(int i, int j){
    tallPa=true;//boolsk variabel som avgjør om tallene på aksene skal vises
    rutenettPa=true;//boolsk variabel som avgjør om rutenettet skal vises
    Xintervall=Yintervall=50; //standard avstand i pixler mellom 1..2..3..4..tallene på aksene
    xskala=yskala=1; //skala blir satt til 1 som standard. den teller da 1..2..3..4.. på aksene
    antallUttrykk=25; //antall uttrykk som er satt til å vises, max 25

    start=true; //boolsk variabel som setter noen verdier bare en gang.
    fakultetParantes=false;
    negativParantes=false;
    negativFakultet=false;
    potensParantes=false;
    grensefunksjon=""; //holder rede på grensefunksjonene
    origoY=0; //init verdier på origoY og origoX
    origoX=0;
    antallRekkerSomSkalVises=25;//hvor mange av de siste rekkeutviklingene som skal vises, max 25

    //init tabell for å huske de tidlige uttrykkene
    tegneUttrykk= new String[antallUttrykk];
    tegneUttrykkTrue = new boolean [antallUttrykk];
    for(int k=0;k<antallUttrykk;k++){
        tegneUttrykk[k]="";
        tegneUttrykkTrue[k]=true;
    }//end k-forløkke

} //end tegnevindu

public void paint(Graphics g){
    //noen variable

```

```

Dimension dimensionTegneVindu = getSize();
width = dimensionTegneVindu.width;
height = dimensionTegneVindu.height;
image = createImage(width,height);

if(start){
    //stiller inn vinduet første gang for å få origo i midten
    gammelBredde= width;
    gammelHoyde= height;
    origoY=width/2;
    origoX=height/2;
    xMax=(width/2)/Xintervall*xskala;
    xMin=-1.0*xMax;
    yMax=(height/2)/Yintervall*y skala;
    yMin=-1.0*yMax;
    start=false;

    }//end if (start)
if(height!=gammelHoyde || width!=gammelBredde){
    //stiller inn vinduet etter forandring for å få origo i midten
    gammelBredde=width;
    gammelHoyde=height;
    origoY=width/2;
    origoX=height/2;
    xMax=(width/2)/Xintervall*xskala;
    xMin=-1.0*xMax;
    yMax=(height/2)/Yintervall*y skala;
    yMin=-1.0*yMax;
} //end if height!=gammelHoyde || width!=gammelBredde

if(!start){

    Graphics g1 = image.getGraphics();
    tegnAkser(g1);//tegner opp akser
    tegnRutenett(g1); //tegner opp rutenett
    tallPaGraf(g1); //tegner opp tall
    DelingsVindu.utviklingPanel.infoFeltTF.setText("Tegner opp graf");
    tegnGraf(g1); //tegner opp grafene
    g.drawImage(image,0,0,null);

    //linjene som kommer opp når man zoomer
    g.drawLine(xNed,yNed,xOpp,yNed); //øvre
    g.drawLine(xNed,yNed,xNed,yOpp); //rett ned
    g.drawLine(xOpp,yNed,xOpp,yOpp); // rett ned siste kant
    g.drawLine(xOpp,yOpp,xNed,yOpp); // nederste

} //if !start
//oppdaterer infofeltet
String temp= new String(DelingsVindu.utviklingPanel.infoFeltTF.getText());

```



```

        //y-akse
        g.drawLine(origoY,0,origoY,height);
        Polygon px = new Polygon();
        px.addPoint(width,origoX);
        px.addPoint(width-10,origoX-5);
        px.addPoint(width-10,origoX+5);
        g.fillPolygon(px);

    }//end tegnAkser

    public void tegnRutenett(Graphics g){
        if(rutenettPa){
            g.setColor(Color.GRAY);

            ///////tegner x-rutenett
            for(int i=origoY+(int)Xintervall;i<width;i=i+(int)Xintervall){
                g.drawLine(i,0,i,height);
            }//for løkke
            for(int i=origoY-(int)Xintervall;i>0;i=i-(int)Xintervall){
                g.drawLine(i,0,i,height);
            }//for løkke

            ///////tegner y-rutenett
            for(int i=origoX+(int)Yintervall;i<height;i=i+(int)Yintervall){
                g.drawLine(0,i,width,i);
            }//for løkke
            for(int i=origoX-(int)Yintervall;i>0;i=i-(int)Yintervall){
                g.drawLine(0,i,width,i);
            }//for løkke
        }//if
    }//end tegnRutenett

    public void tallPaGraf(Graphics g){
        int tempX=0;
        int tempY=0;
        if(tallPa){
            g.setColor(Color.GRAY);

            // xverdier
            //tall til høyre for origo...
            boolean over=true;
            boolean akseforskjovet=false;
            tempX=origoX;
            tempY=origoY;
            if(origoX<=0){
                origoX=20;
                akseforskjovet=true;
            }//end if
        }
    }

```

```

else if(origoX>height){
    origoX=height-20;
    akseforskjovet=true;
} //end if

for(int i=origoY;i<=width;i=(int)(i+Xintervall)){
    if(i!=0){
        if(over){ //annethver tall over og under x-aksen
            g.setColor(Color.BLUE);
            // kutt ned kommatall
            if(((i-origoY)/Xintervall*xskala)<1) g.drawString(
""+(double)((int)(((i-origoY)/Xintervall*xskala)*100000))/100000,(int)(i+2),origoX-2); // kutt ned kommatall
            else g.drawString( ""+((i-origoY)/Xintervall*xskala),(int)(i+2),origoX-
2);

                over=false;
                if(akseforskjovet)over=true;
            } //end if over
            else{
                if(((i-origoY)/Xintervall*xskala)<1)g.drawString(
""+(double)((int)(((i-origoY)/Xintervall*xskala)*100000))/100000,(int)(i+2),origoX+15); // kutt ned kommatall

                    else g.drawString( ""+((i-
origoY)/Xintervall*xskala),(int)(i+2),origoX+15);
                over=true;
            } //end else
        } //end if
    } //end forløkke

    //tall til venstre for origo...
    for(int i=origoY-(int)Xintervall;i>=0;i=(int)(i-Xintervall)){
        if(over){ //annethver tall over og under x-aksen
            if(Math.abs((i-origoY)/Xintervall*xskala)<1)g.drawString(
""+(double)((int)(((i-origoY)/Xintervall*xskala)*100000))/100000,(int)(i+2),origoX-2); // kutt ned kommatall
            else g.drawString( ""+((i-origoY)/Xintervall*xskala),(int)(i+2),origoX-2);
            over=false;
            if(akseforskjovet)over=true;
        } //end if over
        else{
            if((Math.abs(i-origoY)/Xintervall*xskala)<1)g.drawString(
""+(double)((int)(((i-origoY)/Xintervall*xskala)*100000))/100000,(int)(i+2),origoX+15); // kutt ned kommatall

                else g.drawString( ""+((i-origoY)/Xintervall*xskala),(int)(i+2),origoX+15);

                    over=true;
            } //end else
        } //end forløkke

        origoX=tempX;

```

```

//// y verdier

tempY=origoY; //legger inn hvis origoX er mindre enn null, så skal teksten komme på venstre
side
if(origoY<=0){
    origoY=10;
} //end if
if(origoY>=width){
    origoY=width-50;
} //end if
g.setColor(Color.MAGENTA);
//under origo
for(int i=origoX+(int)Yintervall;i<height;i=(int)(i+Yintervall)){
    if(i!=0){
        if((Math.abs((origoX-i)/Yintervall*yskala)<1))g.drawString(
""+(double)((int)(((origoX-i)/Yintervall*yskala)*100000))/100000,origoY+5,(int)(i-2));
        else g.drawString( ""+(((origoX-i)/Yintervall*yskala)),origoY+5,(int)(i-2));

    } //end if
} //end forløkke

//over origo
for(int i=origoX-(int)Yintervall;i>0;i=(int)(i-Yintervall)){
    if(i!=0){
        if((Math.abs((origoX-i)/Yintervall*yskala)<1))g.drawString(
""+(double)((int)(((origoX-i)/Yintervall*yskala)*100000))/100000,origoY+5,(int)(i-2));
        else g.drawString( ""+((origoX-i)/Yintervall*yskala),origoY+5,(int)(i-2));

    } //end if
} //end forløkke

origoY=tempY; //
g.setColor(Color.BLACK);
} //end tall

} //end tallPaGraf

```

```

public String letEtterLNellerLOG(String uttrykk){
    //rutine for å regne ut ln eller log
    String temp="";
    if(uttrykk.contains("\n")){
        for(int i=0;i<uttrykk.length()-1;i++){
            if((uttrykk.charAt(i)=='L' || uttrykk.charAt(i)=='l')&&
(uttrykk.charAt(i+1)=='n' || uttrykk.charAt(i+1)=='N')){
                for(int j=i+2;j<uttrykk.length();j++){

```

```

if(sjekkEtterOperator(j,uttrykk)&&(j!=i+2)&&uttrykk.charAt(j-1)!='E'){
    if(gjorTilTall(uttrykk.substring(i+2,j))>0){
        temp="" + Math.log(gjorTilTall(uttrykk.substring(i+2,j)));
        if(!temp.contains("Infinity")){
            return
letEtterLNellerLOG(uttrykk.substring(0,i)+temp+uttrykk.substring(j,uttrykk.length()));
        } //end if !=infinity
        else{
            Infinity=true;
            return "0";
        } //end else
    } //temp >0
    else{
        DelingsVindu.utviklingPanel.infoFeltTF.setText("Negativ ln eller log");
        DelingsVindu.utviklingPanel.infoFeltTF.setBackground(Color.YELLOW);
        Infinity=true;
        return "0"; //her blir det litt
    }
} //end else
} //end operatorsjekk
else if(j==uttrykk.length()-1){
    if(gjorTilTall(uttrykk.substring(i+2,j+1))>0){
        temp="" + Math.log(gjorTilTall(uttrykk.substring(i+2,j+1)));
        if(!temp.contains("Infinity")){
            return
letEtterLNellerLOG(uttrykk.substring(0,i)+temp);
        } //end if !=infinity
        else{
            Infinity=true;
            return "0";
        } //end else
    } //end if >0
    else{
        DelingsVindu.utviklingPanel.infoFeltTF.setText("Negativ ln eller log");
        DelingsVindu.utviklingPanel.infoFeltTF.setBackground(Color.YELLOW);
        Infinity=true;
        return "0";
    }
}
}

```

```

        } //end else

        } //uttrykk.length()-1

        } //end j-forløkke
    } //end if

        } //end i-forløkke
    } // contain("LN")
    else{
        //ingen LN funnet...gjør ingenting
    }

    ///logaritme
    if(uttrykk.contains("log")){
        for(int i=0;i<uttrykk.length()-1;i++){
            if((uttrykk.charAt(i)=='L' || uttrykk.charAt(i)=='l')&&
(uttrykk.charAt(i+1)=='o' || uttrykk.charAt(i+1)=='O')&& (uttrykk.charAt(i+2)=='g' || uttrykk.charAt(i+2)=='G')){
                for(int j=i+3;j<uttrykk.length();j++){

                    if(sjekkEtterOperator(j,uttrykk)&&(j!=i+3)&&uttrykk.charAt(j-1)!='E'){

                        if(gjørTilTall(uttrykk.substring(i+3,j))>0){

                            temp="" + Math.log10(gjørTilTall(uttrykk.substring(i+3,j)));
                            if(!temp.contains("Infinity")){
                                return
                                letEtterLNeLLerLOG(uttrykk.substring(0,i)+temp+uttrykk.substring(j,uttrykk.length()));

                                } //end if != infinity
                            else{
                                Infinity=true;
                                return "0";
                            } //end else
                        } //temp >0
                    } else{

                        DelingsVindu.utviklingPanel.infoFeltTF.setText("Negativ ln eller log");

                        DelingsVindu.utviklingPanel.infoFeltTF.setBackground(Color.YELLOW);
                        Infinity=true;
                        return "0";
                    } //end else
                } //end operatorsjekk
            } else if(j==uttrykk.length()-1){

                if(gjørTilTall(uttrykk.substring(i+3,j+1))>0){

```

```

temp="" + Math.log10(gjørTilTall(uttrykk.substring(i+3,j+1)));
                                                                    if(!temp.contains("Infinity")){
                                                                    return
letEtterLNellerLOG(uttrykk.substring(0,i)+temp);
                                                                    }//end if !=infinity
                                                                    else{
                                                                    Infinity=true;
                                                                    return "0";
                                                                    }//end else

                                                                    }//end if >0
                                                                    else{

DelingsVindu.utviklingPanel.infoFeltTF.setText("Negativ ln eller log");

DelingsVindu.utviklingPanel.infoFeltTF.setBackground(Color.YELLOW);
                                                                    Infinity=true;
                                                                    return "0";
                                                                    }//end else

                                                                    } //uttrykk.length()-1
                                                                    }//end j-forløkke
                                                                    }//end if
                                                                    }//end i-forløkke
                                                                    }// contain("LOG")
                                                                    else{
                                                                    //      ingen Log funnet...gjør ingenting
                                                                    }//end else

                                                                    return uttrykk;
                                                                    }//end sjekkEtterLNellerLOG

public boolean sjekkEtterOperator(int sjekkPlass,String tempUttrykk){
    //leier etter operator i tempUttrykk på posisjon sjekkPlass
    if(tempUttrykk.charAt(sjekkPlass)=='+' || tempUttrykk.charAt(sjekkPlass)=='-'
    || tempUttrykk.charAt(sjekkPlass)=='*' || tempUttrykk.charAt(sjekkPlass)=='/' || tempUttrykk.charAt(sjekkPlass)
    =='^' || tempUttrykk.charAt(sjekkPlass)=='(' || tempUttrykk.charAt(sjekkPlass)==''){
        return true;
    }//end if
    else{
        return false;
    }//end else
}//end sjekkEtterOperator

public String SinCosTan(String uttrykk){
    //rutine for utregning av sin(h) cos(h) tan(h)
    //sinuser

```

```

if(uttrykk.contains("sinh")){
    for(int i=3;i<uttrykk.length();i++){
        if( (uttrykk.charAt(i-3)=='S' | uttrykk.charAt(i-3)=='s')&&(uttrykk.charAt(i-2)=='I' | uttrykk.charAt(i-2)=='i')&&(uttrykk.charAt(i-1)=='N' | uttrykk.charAt(i-1)=='n')&&(uttrykk.charAt(i)=='H' | uttrykk.charAt(i)=='h') ){
            for(int j=i+1;j<uttrykk.length();j++){
                if(sjekkEtterOperator(j,uttrykk)&&j!=i+1&&uttrykk.charAt(j-1)!='E'){
                    uttrykk=uttrykk.substring(0,i-3)+Math.sinh(gjørTilTall(uttrykk.substring(i+1,j)))+uttrykk.substring(j,uttrykk.length());
                    j=uttrykk.length()+50;
                    i=3; //leiter på nytt...
                }
            }
        }
        else if(j==uttrykk.length()-1){
            uttrykk=uttrykk.substring(0,i-3)+Math.sinh(gjørTilTall(uttrykk.substring(i+1,uttrykk.length())));
            j=uttrykk.length()+50;
            i=uttrykk.length()+50;
        }
    }
}
}

else if(uttrykk.contains("sin")){
    for(int i=2;i<uttrykk.length();i++){
        if( (uttrykk.charAt(i-2)=='S' | uttrykk.charAt(i-2)=='s')&&(uttrykk.charAt(i-1)=='I' | uttrykk.charAt(i-1)=='i')&&(uttrykk.charAt(i)=='N' | uttrykk.charAt(i)=='n') ){
            for(int j=i+1;j<uttrykk.length();j++){
                if(sjekkEtterOperator(j,uttrykk)&&j!=i+1&&uttrykk.charAt(j-1)!='E'){
                    uttrykk=uttrykk.substring(0,i-2)+Math.sin(gjørTilTall(uttrykk.substring(i+1,j)))+uttrykk.substring(j,uttrykk.length());
                    j=uttrykk.length()+50;
                    i=2; //leiter på nytt...
                }
            }
        }
        else if(j==uttrykk.length()-1){
            uttrykk=uttrykk.substring(0,i-2)+Math.sin(gjørTilTall(uttrykk.substring(i+1,uttrykk.length())));
            j=uttrykk.length()+50;
            i=uttrykk.length()+50;
        }
    }
}

//cosinuser
else if(uttrykk.contains("cosh")){

```

```

        for(int i=3;i<uttrykk.length();i++){
            if( (uttrykk.charAt(i-3)=='C' | | uttrykk.charAt(i-3)=='c')&&(uttrykk.charAt(i-
2)=='O' | | uttrykk.charAt(i-2)=='o')&&(uttrykk.charAt(i-1)=='S' | | uttrykk.charAt(i-
1)=='s')&&(uttrykk.charAt(i)=='H' | | uttrykk.charAt(i)=='h' )){
                for(int j=i+1;j<uttrykk.length();j++){
                    if(sjekkEtterOperator(j,uttrykk)&&j!=i+1&&uttrykk.charAt(j-
1)!='E'){
                        uttrykk=uttrykk.substring(0,i-
3)+Math.cosh(gjørTilTall(uttrykk.substring(i+1,j)))+uttrykk.substring(j,uttrykk.length());
                        j=uttrykk.length()+50;
                        i=3; //leiter på nytt...
                    }//end if
                    else if(j==uttrykk.length()-1){
                        uttrykk=uttrykk.substring(0,i-
3)+Math.cosh(gjørTilTall(uttrykk.substring(i+1,uttrykk.length())));
                        j=uttrykk.length()+50;
                        i=uttrykk.length()+50;
                    }//end length-1
                }//end j forløkke
            }//end ==C ==O ==S
        }//end forløkke
    }// contain "cosh"

    else if(uttrykk.contains("cos")){
        for(int i=2;i<uttrykk.length();i++){
            if( (uttrykk.charAt(i-2)=='C' | | uttrykk.charAt(i-2)=='c')&&(uttrykk.charAt(i-
1)=='O' | | uttrykk.charAt(i-1)=='o')&&(uttrykk.charAt(i)=='S' | | uttrykk.charAt(i)=='s' )){
                for(int j=i+1;j<uttrykk.length();j++){
                    if(sjekkEtterOperator(j,uttrykk)&&j!=i+1&&uttrykk.charAt(j-
1)!='E'){
                        uttrykk=uttrykk.substring(0,i-
2)+Math.cos(gjørTilTall(uttrykk.substring(i+1,j)))+uttrykk.substring(j,uttrykk.length());
                        j=uttrykk.length()+50;
                        i=2; //leiter på nytt...
                    }//end if
                    else if(j==uttrykk.length()-1){
                        uttrykk=uttrykk.substring(0,i-
2)+Math.cos(gjørTilTall(uttrykk.substring(i+1,uttrykk.length())));
                        j=uttrykk.length()+50;
                        i=uttrykk.length()+50;
                    }//end else length-1
                }//end j-forløkke
            }//end ==C ==O ==S
        }//end forløkke
    }// contain "cos"

    ///tangenser
    else if(uttrykk.contains("tanh")){
        for(int i=3;i<uttrykk.length();i++){

```

```

        if( (uttrykk.charAt(i-3)=='T' || uttrykk.charAt(i-3)=='t')&&(uttrykk.charAt(i-
2)=='A' || uttrykk.charAt(i-2)=='a')&&(uttrykk.charAt(i-1)=='N' || uttrykk.charAt(i-
1)=='n')&&(uttrykk.charAt(i)=='H' || uttrykk.charAt(i)=='h') ){
            for(int j=i+1;j<uttrykk.length();j++){
                if(sjekkEtterOperator(j,uttrykk)&&j!=i+1&&uttrykk.charAt(j-
1)!='E'){
                    uttrykk=uttrykk.substring(0,i-
3)+Math.tanh(gjorTilTall(uttrykk.substring(i+1,j)))+uttrykk.substring(j,uttrykk.length());
                    j=uttrykk.length()+50;
                    i=3; //leiter på nytt...
                }//end if
                else if(j==uttrykk.length()-1){
                    uttrykk=uttrykk.substring(0,i-
3)+Math.tanh(gjorTilTall(uttrykk.substring(i+1,uttrykk.length())));
                    j=uttrykk.length()+50;
                    i=uttrykk.length()+50;
                }//end length-1
            }//end j forløkke
        }//end ==C ==O ==S
    }//end forløkke
} // contain "cosh"

else if(uttrykk.contains("tan")){
    for(int i=2;i<uttrykk.length();i++){
        if( (uttrykk.charAt(i-2)=='T' || uttrykk.charAt(i-2)=='t')&&(uttrykk.charAt(i-
1)=='A' || uttrykk.charAt(i-1)=='a')&&(uttrykk.charAt(i)=='N' || uttrykk.charAt(i)=='n') ){
            for(int j=i+1;j<uttrykk.length();j++){
                if(sjekkEtterOperator(j,uttrykk)&&j!=i+1&&uttrykk.charAt(j-
1)!='E'){
                    uttrykk=uttrykk.substring(0,i-
2)+Math.tan(gjorTilTall(uttrykk.substring(i+1,j)))+uttrykk.substring(j,uttrykk.length());
                    j=uttrykk.length()+50;
                    i=2; //leiter på nytt...
                }//end if
                else if(j==uttrykk.length()-1){
                    uttrykk=uttrykk.substring(0,i-
2)+Math.tan(gjorTilTall(uttrykk.substring(i+1,uttrykk.length())));
                    j=uttrykk.length()+50;
                    i=uttrykk.length()+50;
                }//end else length-1
            }//end j-forløkke
        }//end ==C ==O ==S
    }//end forløkke
} // contain "cos"

return uttrykk;
} //end SinCosTan

```

```

public String parantes(String parantesUttrykk){
    //variable
    parantesUttrykk=potens(parantesUttrykk,false);
    parantesUttrykk=SinCosTan(parantesUttrykk);
    parantesUttrykk=letEtterLNellerLOG(parantesUttrykk);
    parantesUttrykk=fakultet(parantesUttrykk);
    parantesUttrykk=gangeDele(parantesUttrykk);
    parantesUttrykk=plussMinus(parantesUttrykk);
    if(parantesUttrykk.equalsIgnoreCase("-0.0")) return "0";
    //la inn en dummie her, siden -2*0.0 noen ganger gav svaret -0.0 og dette skapte krøll
    return (parantesUttrykk);
}
//end parantes

```

```

public void utregning(String regningInn, Graphics g,Color fargeValg,boolean tegne,boolean epsi){
    //variable
    double deltaX=xskala/Xintervall;
    double deltaY=yskala/Yintervall;
    double sendverdiTilbake=0;
    int spessialTegn=0;
    String regnUt="";
    int yForrige=origoX;
    int xForrige=origoY;
    int yForrigePlussEpsilon=0;
    int yForrigeMinusEpsilon=0;
    double yVerdi=0;
    int stillXverdi=0;

    if(tegne){//regner bare ut hvis funksjonen skal tegnes...

        ForrigeInfinity=true;

        for(int i=origoY;i<=width;i++){ //alle x-verdiene til høyre for origo

            regnUt=regningInn;
            Infinity=false;
            fakultetParantes=false;
            negativFakultet=false;
            for(int j=0;j<regnUt.length();j++){ //erstatte x-verdier
                if(regnUt.charAt(j)=='x' | regnUt.charAt(j)=='X'){
                    stillXverdi=(int)((i-origoY)*deltaX*1000);
                    if(j>1){
                        if(regnUt.charAt(j-1)==' '){

                            regnUt=regnUt.substring(0,j)+"*"+((double)(stillXverdi)/1000)+regnUt.substring(j+1,regnUt.length());

                        }
                    }
                }
            }
        }
    }
}
//end if == )
else{

```

```

regnUt=regnUt.substring(0,j)+((double)(stillXverdi)/1000)+regnUt.substring(j+1,regnUt.length());
        //end else
    } //end j>1
    else
regnUt=regnUt.substring(0,j)+((double)(stillXverdi)/1000)+regnUt.substring(j+1,regnUt.length());

        j=0; //sjekker etter flere x-verdier
    } // sjekk etter x-verdi
} //for j-løkke x er byttet ut...

spessialTegn=regnUt.length()-1;
int sluttParentes=regnUt.length()-1;
boolean potensParantes=false;
fakultetParantes=false;

for(int j=regnUt.length()-1;j>=0;j--){ //finne parentes uttrykk

    if(regnUt.charAt(j)==''){ //sluttparantes funnet
        sluttParentes=j;
        for(int k=j-1;k>=0;k--){ //leiter etter flere ) paranteser for en
eventuelt ny
                                if(regnUt.charAt(k)=='(')break; //fant start
parantes...hopper ut av løkka
                                else if(regnUt.charAt(k)==' '){
                                    sluttParentes=k;

                                } //end else if == )
                            } //end k-forløkke
                        if(j<=regnUt.length()-2){
                            if(regnUt.charAt(sluttParentes+1)=='^'){ //potens etter
parantes gjøres for å regnet ut negative potenser
                                potensParantes=true;
                            } //end if == ^
                            else if(regnUt.charAt(sluttParentes+1)=='!'){
                                fakultetParantes=true; //fakultet funnet etter
parantes...sjekk om negativ.
                            } //end else if == !
                        } // end if j<regnUt.length()-2
                    } //end of == )
                else if(regnUt.charAt(j)=='('){ //startparantes funnet

                    //legg til negativ parantes eller negativ fakultetsparantes her...
                    String tempParentes=new String("");
                    tempParentes = parantes(regnUt.substring(j+1,sluttParentes));

                    if(tempParentes.length(>0){
                        if(tempParentes.charAt(0)=='-'){ //negativ parentes... fikses opp
                            if(fakultetParantes){

```



```

regnUt=
regnUt.substring(0,j)+fakultet(tempParentes+"!")+regnUt.substring(sluttParentes+2,regnUt.length());
j=regnUt.length();
potensParantes=false;
fakultetParantes=false;

} //end fakultetparentes
else{
if(sluttParentes==regnUt.length()-1){
regnUt=regnUt.substring(0,j)+tempParentes;
j=regnUt.length();
potensParantes=false;
fakultetParantes=false;
} //end if length -1
else{

regnUt=regnUt.substring(0,j)+tempParentes+regnUt.substring(sluttParentes+1,regnUt.length());
j=regnUt.length();
potensParantes=false;
fakultetParantes=false;
} //end else

} //end else
} //end length >0
} //end else if ==(
else if(j==0){ //ingen flere parenteser
regnUt=potens(regnUt,false);
regnUt=letEtterLNellerLOG(regnUt);
regnUt=SinCosTan(regnUt);
regnUt=fakultet(regnUt);
regnUt=gangeDele(regnUt);
regnUt=plussMinus(regnUt);
} //end else if ==0
} //end parantes forløkke

if((i-origoY)==0){ //startpunkt på tegning
yVerdi=(gjorTilTall(regnUt)+0.00000001); //av en eller annen merkelig grunn ble
linjen F(x) = x hakkete uten denne lille feilopprettingen
//setter start verdiene
yPlussEpsilon=yVerdi+epsilonVerdi;
yMinusEpsilon=yVerdi-epsilonVerdi;
yVerdi=(yVerdi*Yintervall/yskala);
yPlussEpsilon=(yPlussEpsilon*Yintervall/yskala);
yMinusEpsilon=(yMinusEpsilon*Yintervall/yskala);
yForrige=(origoX)-(int)yVerdi;
yForrigePlussEpsilon=(origoX)-(int)yPlussEpsilon;

yForrigeMinusEpsilon=(origoX)-(int)yMinusEpsilon;
} //end if ==1

```

```

else{
    if(regnUt.contains("E-")){
        //nothing
    }//end if E-
    else if(regnUt.contains("E+") || regnUt.contains("E")){
        Infinity=true; //utelukker å tegne når det er alt for høye tall. får problemer
        DelingsVindu.utviklingPanel.infoFeltTF.setText("Ekstremt høye verdier");
        DelingsVindu.utviklingPanel.infoFeltTF.setBackground(Color.YELLOW);

    }//end if E+ eller E
    else{
        //nothing
    }//end else

    if(Infinity){
        xForrige++;
        ForrigeInfinity=true;
    }//end if
    else{
        yVerdi=(gjorTilTall(regnUt)+0.00000001);//av en eller annen merkelig grunn
        ble linjen  $F(x) = x$  hakkete uten denne lille feilopprettingen

        if(epsi){
            yPlussEpsilon=yVerdi+epsilonVerdi;
            yMinusEpsilon=yVerdi-epsilonVerdi;
            yPlussEpsilon=yPlussEpsilon*Yintervall/yskala;
            yMinusEpsilon=yMinusEpsilon*Yintervall/yskala;
        }//end epsi
        yVerdi=yVerdi*Yintervall/yskala;
        if(ForrigeInfinity){
            yForrige=(origoX-(int)yVerdi);
            yForrigePlussEpsilon=(origoX)-(int)yPlussEpsilon;

            yForrigeMinusEpsilon=(origoX)-(int)yMinusEpsilon;
            ForrigeInfinity=false;
        }//end if
        g.setColor(fargeValg);
        g.drawLine(xForrige,yForrige ,xForrige+1, origoX-(int)yVerdi);
        g.drawLine(xForrige,yForrige+1 ,xForrige+1,(origoX-(int)yVerdi)+1);
        g.drawLine(xForrige,yForrige-1 ,xForrige+1,(origoX-(int)yVerdi)-1);
        if(epsi){
            g.setColor(Color.red);
            //epsilon pluss
            g.drawLine(xForrige,yForrigePlussEpsilon ,xForrige+1, origoX-
(int)yPlussEpsilon);
            g.drawLine(xForrige,yForrigePlussEpsilon+1 ,xForrige+1,(origoX-
(int)yPlussEpsilon)+1);

```

```

        g.drawLine(xForrige,yForrigePlussEpsilon-1 ,xForrige+1,(origoX-
(int)yPlussEpsilon)-1);
        //epsilon minus
        g.drawLine(xForrige,yForrigeMinusEpsilon ,xForrige+1, origoX-
(int)yMinusEpsilon);
        g.drawLine(xForrige,yForrigeMinusEpsilon+1 ,xForrige+1,(origoX-
(int)yMinusEpsilon)+1);
        g.drawLine(xForrige,yForrigeMinusEpsilon-1 ,xForrige+1,(origoX-
(int)yMinusEpsilon)-1);
        g.setColor(fargeValg);
        yForrigePlussEpsilon=origoX-(int)yPlussEpsilon;
        yForrigeMinusEpsilon=origoX-(int)yMinusEpsilon;
    } //end epsi
    xForrige++;
    yForrige=origoX-(int)yVerdi;

        } //end else
    } //end else
} //end i-forløkke

/////*****til venstre for midtpunkt*****
xForrige=origoY;
yForrige=origoX;
yVerdi=0;
boolean nesteErpotens=false;
int potensslutt=regnUt.length();
ForrigeInfinity=true;

    for(int i=origoY;i>=0;i--){ //alle x-verdier til venstre for origo...

        regnUt=regningInn;
        Infinity=false;
        boolean potensParantes=false;
        fakultetParantes=false;
        negativFakultet=false;
        nesteErpotens=false ;
        for(int j=0;j<regnUt.length();j++){ //erstatte x-verdier
            if(regnUt.charAt(j)=='x' | |regnUt.charAt(j)=='X'){
                stillXverdi=(int)((i-origoY)*deltaX*1000);
                if(j>1){
                    if(regnUt.charAt(j-1)==' '){
                        regnUt=regnUt.substring(0,j)+"*"+((double)(stillXverdi)/1000)+" "+regnUt.substring(j+1,regnUt.length(
h()));
                    } //end if
                } else{
                    regnUt=regnUt.substring(0,j)+" "+((double)(stillXverdi)/1000)+" "+regnUt.substring(j+1,regnUt.length(
));
                }
            }
        }
    }
}

```

```

        } //end else
    } //end if j>1
    else
regnUt=regnUt.substring(0,j)+"("+(double)(stillXverdi)/1000)+"")+regnUt.substring(j+1,regnUt.length());

        j=0; //sjekker etter flere x-verdier
        } // sjekk etter x-verdi
    } //for j-løkke x er byttet ut...

int sluttParentes=regnUt.length()-1;
potensParantes=false;
fakultetParantes=false;
for(int j=regnUt.length()-1;j>=0;j--){ //finne parentes uttrykk
    if(regnUt.charAt(j)==''){
        sluttParentes=j;
        for(int k=j-1;k>=0;k--){ //leiter etter flere (
            if(regnUt.charAt(k)=='(') break;
            else if(regnUt.charAt(k)=='('){
                sluttParentes=k;
            } //end else if == )
        } //end k-forløkke
    } if(j<regnUt.length()-2){
        if(regnUt.charAt(sluttParentes+1)=='^'){ //potens etter parentes
gjøres for å regnet ut negative potenser
                potensParantes=true;
            } //end if
            else if(regnUt.charAt(sluttParentes+1)=='!'){
                fakultetParantes=true; //fakultet funnet etter
parantes...sjekk om negativ.
            } //end else if
        } // end if j<regnUt.length()-
    } //end of == )
    else if(regnUt.charAt(j)=='('){ //startparentes funnet
        String tempParentes=new String("");
        tempParentes = parentes(regnUt.substring(j+1,sluttParentes));

        if(tempParentes.length()>0){
            if(tempParentes.charAt(0)=='-'){ //negativ parentes... fikses opp
                if(fakultetParantes){
                    Infinity=true; //kan ikke tegne negative fakteteter
                    regnUt="0";
                } //end if fakultetparantes
            else if(potensParantes){ //finn slutten på potensen
                for(int k=sluttParentes+2;k<regnUt.length();k++){
                    if((regnUt.charAt(k)=='-'
'| | regnUt.charAt(k)=='+' | | regnUt.charAt(k)=='*' | | regnUt.charAt(k)=='/' | | regnUt.charAt(k)=='^' | | regnUt.char
At(k)=='')&&(k>sluttParentes+2)){
                                if(regnUt.charAt(k-1)=='E'){
                                    //nothing

```

```

        //end if
        else{

            regnUt=regnUt.substring(0,j)+potens(tempParentes+regnUt.substring(sluttParentes+1,k),true)+regnU
t.substring(k,regnUt.length());

            //leier på nytt etter paranteser

            j=regnUt.length();

            k=regnUt.length()+50;
            potensParantes=false;
            fakultetParantes=false;

        }//end else
    }//end if operator
    else if(k==regnUt.length()-1){

        regnUt=regnUt.substring(0,j)+potens(tempParentes+regnUt.substring(sluttParentes+1,regnUt.length(
)),true);

        //leier på
        nytt etter paranteser

        j=regnUt.length(); //leier på

        k=regnUt.length()+50;
        potensParantes=false;
        fakultetParantes=false;

    }//end else if length-1
    }//end k-forløkke
}//end potensparantes
else{ //bare minusparantes...

    regnUt=regnUt.substring(0,j)+tempParentes+regnUt.substring(sluttParentes+1,regnUt.length());
    j=regnUt.length(); //leier på nytt etter
    paranteser

    potensParantes=false;
    fakultetParantes=false;

    }//else bare minus

}//end charat(0)==-
else if(fakultetParantes){
    regnUt=
regnUt.substring(0,j)+fakultet(tempParentes+"!")+regnUt.substring(sluttParentes+2,regnUt.length());
    j=regnUt.length();
    potensParantes=false;
    fakultetParantes=false;

}//end else if
else{
    if(sluttParentes==regnUt.length()-1){
        regnUt=regnUt.substring(0,j)+tempParentes;
        j=regnUt.length();
    }
}

```

```

                potensParantes=false;
                fakultetParantes=false;
            } //end if
            else {
                regnUt=regnUt.substring(0,j)+tempParentes+regnUt.substring(sluttParentes+1,regnUt.length());
                j=regnUt.length();
                potensParantes=false;
                fakultetParantes=false;
            } //end else

        } //end else
    } //end length >0
} //end else if ==(
else if(j==0){
    //ingen flere parenteser
    regnUt=potens(regnUt,false);
    regnUt=letEtterLNellerLOG(regnUt);
    regnUt=SinCosTan(regnUt);
    regnUt=fakultet(regnUt);
    regnUt=gangeDele(regnUt);
    regnUt=plussMinus(regnUt);
} //end else if ==0
} //end parentes forløkke
if((origoY-i)==1){ //startpunkt på tegning
    yVerdi=(gjorTilTall(regnUt)+0.00000001); //av en eller annen merkelig grunn ble
linjen F(x) = x hakkete uten denne lille feilopprettingen
    yPlussEpsilon=yVerdi+epsilonVerdi;
    yMinusEpsilon=yVerdi-epsilonVerdi;
    yVerdi=(yVerdi*Yintervall/yskala);
    yPlussEpsilon=(yPlussEpsilon*Yintervall/yskala);
    yMinusEpsilon=(yMinusEpsilon*Yintervall/yskala);
    yForrige=(origoX)-(int)yVerdi;
    yForrigePlussEpsilon=(origoX)-(int)yPlussEpsilon;

    yForrigeMinusEpsilon=(origoX)-(int)yMinusEpsilon;
} //end if
else{
    if(regnUt.contains("E-")){
        //nothing
    } //end if E-
    else if(regnUt.contains("E+") || regnUt.contains("E")){
        Infinity=true; //utelukker å tegne når det er alt for høye tall. får problemer
med standardform

        DelingsVindu.utviklingPanel.infoFeltTF.setText("Ekstremt høye verdier");
        DelingsVindu.utviklingPanel.infoFeltTF.setBackground(Color.YELLOW);

    } //end if E+ eller E
    else{

```

```

        //nothing
    }//end else

    if(Infinity){
        xForrige--;
        ForrigeInfinity=true;
        yForrigePlussEpsilon=(origoX)-(int)yPlussEpsilon;

        yForrigeMinusEpsilon=(origoX)-(int)yMinusEpsilon;
    }//end if
    else{
        yVerdi=(gjorTilTall(regnUt)+0.0000001); //av en eller annen merkelig

```

grunn ble linjen $F(x) = x$ hakkete uten denne lille feilopprettingen

```

        if(epsi){
            yPlussEpsilon=yVerdi+epsilonVerdi;
            yMinusEpsilon=yVerdi-epsilonVerdi;
            yPlussEpsilon=yPlussEpsilon*Yintervall/yskala;
            yMinusEpsilon=yMinusEpsilon*Yintervall/yskala;
        }//end epsi
        yVerdi=yVerdi*Yintervall/yskala;
        if(ForrigeInfinity){
            yForrige=(origoX)-(int)yVerdi;
            yForrigePlussEpsilon=(origoX)-(int)yPlussEpsilon;

            yForrigeMinusEpsilon=(origoX)-(int)yMinusEpsilon;
            ForrigeInfinity=false;
        }//end if
        g.setColor(fargeValg);
        g.drawLine(xForrige,yForrige ,xForrige-1, origoX-((int)yVerdi));
        g.drawLine(xForrige,yForrige+1 ,xForrige-1,(origoX-((int)yVerdi))+1);
        g.drawLine(xForrige,yForrige-1 ,xForrige-1,(origoX-((int)yVerdi))-1);
        if(epsi){
            g.setColor(Color.red);
            //epsilon pluss
            g.drawLine(xForrige,yForrigePlussEpsilon ,xForrige-1, origoX-
(int)yPlussEpsilon);
            g.drawLine(xForrige,yForrigePlussEpsilon+1 ,xForrige-1,(origoX-
(int)yPlussEpsilon)+1);
            g.drawLine(xForrige,yForrigePlussEpsilon-1 ,xForrige-1,(origoX-
(int)yPlussEpsilon)-1);

            //epsilon minus
            g.drawLine(xForrige,yForrigeMinusEpsilon ,xForrige-1, origoX-
(int)yMinusEpsilon);
            g.drawLine(xForrige,yForrigeMinusEpsilon+1 ,xForrige-1,(origoX-
(int)yMinusEpsilon)+1);
            g.drawLine(xForrige,yForrigeMinusEpsilon-1 ,xForrige-1,(origoX-
(int)yMinusEpsilon)-1);

```

```

        g.setColor(fargeValg);

    }
    yForrigePlussEpsilon=origoX-(int)yPlussEpsilon;
    yForrigeMinusEpsilon=origoX-(int)yMinusEpsilon;
    xForrige--;
    yForrige=origoX-(int)yVerdi;
    }//end else
    }//end else
} //end i-forløkke */
} //end tegne

return;
} //end utregning

public String potens(String potensString, boolean negativPotens){
    //regner ut potenser
    int startOperator=0;

    if(potensString.contains("^")){ //inneholder potens
        for(int i=0;i<potensString.length();i++){
            if(negativPotens&& i==0)i++; //hopper over minustegnet på starten
            if(potensString.charAt(i)=='+' || potensString.charAt(i)=='-'
|| potensString.charAt(i)=='*' || potensString.charAt(i)=='/' || potensString.charAt(i)=='n' || potensString.charAt(i)
)=='s' || potensString.charAt(i)=='g'){ //opertor funnet eller sincostanlnlog funnet
                if(i>1){ //leiter etter standardform
                    if(potensString.charAt(i-1)=='E'){
                        //nothing standardform hopp over
                    } //end if
                } else{
                    startOperator=i+1;
                } //end else
            } //end i>1
        } else{
            startOperator=i+1;
        } //end else
    } //end if operator
    else if(potensString.charAt(i)=='^'){
        for(int j=i+1;j<potensString.length();j++){ //finne neste operator

            if(potensString.charAt(j)=='-'&&j==i+1)j++; //luker ut negativt
potens

            if(potensString.charAt(j)=='+' || potensString.charAt(j)=='-'
|| potensString.charAt(j)=='*' || potensString.charAt(j)=='/' || potensString.charAt(j)=='^'){

                if(potensString.charAt(j-1)=='E'){
                    //nothing standardform hopp over
                } //end if
            } else{

```

```

                                return
potensString=potens(potensString.substring(0,startOperator)+Math.pow(gjorTilTall(potensString.substring(sta
rtOperator,i)),gjorTilTall(potensString.substring(i+1,j)))+potensString.substring(j,potensString.length()),false);

                                //else

                                //end if
                                else if(j==potensString.length()-1){

                                return

potensString=potens(potensString.substring(0,startOperator)+Math.pow(gjorTilTall(potensString.substring(sta
rtOperator,i)),gjorTilTall(potensString.substring(i+1,potensString.length()))),false);

                                //end else if
                                } //end j-forløkke
                                if(negativPotens){
                                return
potensString="" +Math.pow(gjorTilTall(potensString.substring(0,i)),gjorTilTall(potensString.substring(i+1,potens
String.length())));
                                //end if
                                //end else if ==^
                                //end i-forløkke
                                return potensString;
                                //end if contains ^
                                else{
                                return potensString;
                                //end else

                                //end potens

public void tegnGraf(Graphics g){

                                if(hvilketUttrykk>=0){
                                String testJalla= new String();
                                for(int i=0;i<=hvilketUttrykk;i++){

                                utregning(tegneUttrykk[i].toLowerCase(),g,DelingsVindu.utviklingPanel.fargeKoder[i],tegneUttrykkTru
e[i],false);

                                //end i-for
                                //end if
                                if(tegnGrenseFunksjon){
                                if(epsilonVerdi==0.0)
utregning(grensefunksjon.toLowerCase(),g,Color.black,true,false);
                                else utregning(grensefunksjon.toLowerCase(),g,Color.black,true,true);

                                }

```

```

} //end tegnGraf

public String gangeDele(String GangeDeleMellomregning){
    //gange og dele rutine...
    if(GangeDeleMellomregning.contains("*") || GangeDeleMellomregning.contains("/")){
        int operatorFor=0;
        for(int i=1;i<GangeDeleMellomregning.length();i++){
            if(GangeDeleMellomregning.charAt(i)=='+' | GangeDeleMellomregning.charAt(i)=='-
        ){
            if(GangeDeleMellomregning.charAt(i-1)=='E'){
                //nothing...standarform tall
            } //end if
            else{
                operatorFor=i+1;
            } //end else
        } //end if operator
        else
        if(GangeDeleMellomregning.charAt(i)=='*' | GangeDeleMellomregning.charAt(i)=='/'){
            for(int j=i+1;j<GangeDeleMellomregning.length();j++){
                if(j==1+i&&GangeDeleMellomregning.charAt(j)=='-'){ //negativt tall
                    //nothing
                } //end if
                else if(sjekkEtterOperator(j,GangeDeleMellomregning)){
                    //operator funnet
                    if(GangeDeleMellomregning.charAt(j-1)=='E'){
                        //nothing...standardform hopp over
                    } //end if
                    else{
                        if(GangeDeleMellomregning.charAt(i)=='*'){
                            //gange

                            GangeDeleMellomregning=(GangeDeleMellomregning.substring(0,operatorFor)+(gjorTilTall(GangeDeleMellomregning.substring(operatorFor,i))*gjorTilTall(GangeDeleMellomregning.substring(i+1,j)))+GangeDeleMellomregning.substring(j,GangeDeleMellomregning.length()));

                            i=0;
                            j=GangeDeleMellomregning.length()+50;
                        } //end if *
                        else if(GangeDeleMellomregning.charAt(i)=='/'){
                            //dele

                            if(gjorTilTall(GangeDeleMellomregning.substring(i+1,j))==0.0){

                                DelingsVindu.utviklingPanel.infoFeltTF.setText("Divisjon med null!");

                                DelingsVindu.utviklingPanel.infoFeltTF.setBackground(Color.YELLOW);
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```
GangeDeleMellomregning=(GangeDeleMellomregning.substring(0,operatorFor)+(gjurTilTall(GangeDeleMellomregning.substring(operatorFor,i))/0.00000001)+GangeDeleMellomregning.substring(j,GangeDeleMellomregning.length()));
```

```
i=0;
```

```
j=GangeDeleMellomregning.length()+50;
```

```
}//end if ==0.0
```

```
else{
```

```
GangeDeleMellomregning=(GangeDeleMellomregning.substring(0,operatorFor)+(gjurTilTall(GangeDeleMellomregning.substring(operatorFor,i))/gjurTilTall(GangeDeleMellomregning.substring(i+1,j)))+GangeDeleMellomregning.substring(j,GangeDeleMellomregning.length()));
```

```
i=0;
```

```
j=GangeDeleMellomregning.length()+50;
```

```
}//end else
```

```
}//end else if == /
```

```
}//end else
```

```
}//end if sjekketteroperator
```

```
else if(j==GangeDeleMellomregning.length()-1){
```

```
//ingen flere operatører...på slutten
```

```
if(GangeDeleMellomregning.charAt(i)=='*'){ //gange
```

```
GangeDeleMellomregning=(GangeDeleMellomregning.substring(0,operatorFor)+(gjurTilTall(GangeDeleMellomregning.substring(operatorFor,i))*gjurTilTall(GangeDeleMellomregning.substring(i+1,GangeDeleMellomregning.length()))));
```

```
i=GangeDeleMellomregning.length()+50;
```

```
j=GangeDeleMellomregning.length()+50;
```

```
}//end if == *
```

```
else if(GangeDeleMellomregning.charAt(i)=='/'){ //dele
```

```
if(gjurTilTall(GangeDeleMellomregning.substring(i+1,GangeDeleMellomregning.length()))==0.0){
```

```
DelingsVindu.utviklingPanel.infoFeltTF.setText("Divisjon med null!");
```

```
DelingsVindu.utviklingPanel.infoFeltTF.setBackground(Color.YELLOW);
```

```
GangeDeleMellomregning=(GangeDeleMellomregning.substring(0,operatorFor)+(gjurTilTall(GangeDeleMellomregning.substring(operatorFor,i))/0.00000001));
```

```
j=GangeDeleMellomregning.length()+50;
```

```
i=GangeDeleMellomregning.length()+50;
```

```
}//end if == 0.0
```

```
else{
```

```
GangeDeleMellomregning=gangeDele(GangeDeleMellomregning.substring(0,operatorFor)+(gjorTilTall
(GangeDeleMellomregning.substring(operatorFor,i))/gjorTilTall(GangeDeleMellomregning.substring(i+1,Gange
DeleMellomregning.length()))));
```

```

i=GangeDeleMellomregning.length()+50;
j=GangeDeleMellomregning.length()+50;
    }//end else
    }//end else if == /
    }//end if length -1
    }//end j-forløkke
    }//end * eller / tegn
    }//end i-forløkke
} //end contains
else{
    return GangeDeleMellomregning;
} //end else
```

```

return GangeDeleMellomregning;
} //end gangedele
```

```

public boolean plussMinusGangeDeleTegn(int plass, String tekst){
    //rutine for sjekk etter + - * / tegn
    if(tekst.charAt(plass)=='+' | | tekst.charAt(plass)=='-'
| | tekst.charAt(plass)=='*' | | tekst.charAt(plass)=='/'){
        return true;
    }
    else
        return false;
} // end plussMinusGangeDeleTegn
```

```

public String plussMinus(String plussMinusMellomregning){
    //rutine for å pluss eller minus regning
    if(plussMinusMellomregning.contains("+") | | plussMinusMellomregning.contains("-")){

        int spesialtegn=0;
        for(int i=0;i<plussMinusMellomregning.length();i++){ //rutine for å fjerne +-, ++, --, --
            if(plussMinusMellomregning.charAt(i)=='+' | | plussMinusMellomregning.charAt(i)=='-')
        }

            if(i<plussMinusMellomregning.length()-1){

                if(plussMinusMellomregning.charAt(i)=='+' && plussMinusMellomregning.charAt(i+1)=='-'
| | plussMinusMellomregning.charAt(i)=='-' && plussMinusMellomregning.charAt(i+1)=='+'){

                    plussMinusMellomregning="" + plussMinusMellomregning.substring(0,i) + "-"
+ plussMinusMellomregning.substring(i+2,plussMinusMellomregning.length());
                    i=0;
                } //end erstattet +- eller -- med -
            }
        }
    }
}
```

```

else if(plussMinusMellomregning.charAt(i)=='-'
'&&plussMinusMellomregning.charAt(i+1)=='-'
'| | plussMinusMellomregning.charAt(i)=='+'&&plussMinusMellomregning.charAt(i+1)=='+'){

    if(i==0)plussMinusMellomregning=plussMinusMellomregning.substring(i+2,plussMinusMellomregning
.length());

    else
plussMinusMellomregning=""+plussMinusMellomregning.substring(0,i)+" "+plussMinusMellomregning.substri
ng(i+2,plussMinusMellomregning.length());
        i=0;
        }//end erstattet +- eller -+ med -
    }//end if < -1
    }//end if ==+ | | ==-
} //end i-forløkke

//nå er +- -+ -- ++ byttet ut riktig matematisk...
for(int i=1;i<plussMinusMellomregning.length();i++){
    if(plussMinusMellomregning.charAt(i)=='+' | | plussMinusMellomregning.charAt(i)=='-'
){ //finne operator

        if(plussMinusMellomregning.charAt(i-1)=='E'){//lukes ut standardform
            //nothing standardform hopp over
        } //end if
        else{
            for(int j=i+1;j<plussMinusMellomregning.length();j++){ //finne
neste operator

                if(plussMinusMellomregning.charAt(j)=='-'
| | plussMinusMellomregning.charAt(j)=='+'){

                    if(plussMinusMellomregning.charAt(j-1)=='E'){
                        //nothing standardform hopp over
                    } //end if
                    else{ //regn ut...

                        if(plussMinusMellomregning.charAt(i)=='+'){

                            plussMinusMellomregning=""+(gjorTilTall(plussMinusMellomregning.substring(0,i))+gjorTilTall(plussM
inusMellomregning.substring(i+1,j)))+plussMinusMellomregning.substring(j,plussMinusMellomregning.length()
);

                                i=1;

                            j=plussMinusMellomregning.length()+50;

                                } //end if ==+
                                else

                                if(plussMinusMellomregning.charAt(i)=='-'){

                                    plussMinusMellomregning=""+(gjorTilTall(plussMinusMellomregning.substring(0,i))-
gjorTilTall(plussMinusMellomregning.substring(i+1,j)))+plussMinusMellomregning.substring(j,plussMinusMello
mregning.length());

                                        i=1;

```

```

j=plussMinusMellomregning.length()+50;
                                                    //end if == -
                                                    //end else regn ut
// end if charat == - || == +
else if(j==plussMinusMellomregning.length()-1){
    if(plussMinusMellomregning.charAt(i)=='+'){

        plussMinusMellomregning=""+(gjorTilTall(plussMinusMellomregning.substring(0,i))+gjorTilTall(plussM
inusMellomregning.substring(i+1,plussMinusMellomregning.length())));

                                                    //end if == +
                                                    else if(plussMinusMellomregning.charAt(i)=='-'){

                plussMinusMellomregning=""+(gjorTilTall(plussMinusMellomregning.substring(0,i))-
gjorTilTall(plussMinusMellomregning.substring(i+1,plussMinusMellomregning.length())));
                                                    //end else if == -
                                                    //end length -1
                                                    //end j-forløkke, finne neste operator
                                                    //end else
                                                    //end i-forløkke operator
                                                    else if(i==plussMinusMellomregning.length()-1){
//
                                                    //end else if
                                                    //end i-forløkke
                                                    //end contain
                                                    else{
//fant ingen + eller - tegn
                                                    //end else

                return plussMinusMellomregning;
                                                    //end plussMinus

public double gjorTilTall(String tall){
//rutine for å parse...
double temp=0;
try{
    temp= Double.parseDouble(tall); //prøver å parse om til double...
//end try
catch(Exception e){ //feil oppstår , oppdater infofelt
    DelingsVindu.utviklingPanel.infoFeltTF.setText("En uventet feil oppstod");
    DelingsVindu.utviklingPanel.infoFeltTF.setBackground(Color.RED);
    DelingsVindu.utviklingPanel.neste.setLabel("Start på nytt");
//end catch

    return temp;
// end gjør til tall

```

```

public boolean sjekkEtterTall(int sjekkPlass,String tempUttrykk){
    //sjekker etter et tall på plassen sjekkPlass i uttrykket

    if(tempUttrykk.charAt(sjekkPlass)=='1' || tempUttrykk.charAt(sjekkPlass)=='2' || tempUttrykk.charAt(sj
ekkkPlass)=='3' || tempUttrykk.charAt(sjekkPlass)=='4' || tempUttrykk.charAt(sjekkPlass)=='5' || tempUttrykk.char
At(sjekkPlass)=='6' || tempUttrykk.charAt(sjekkPlass)=='7' || tempUttrykk.charAt(sjekkPlass)=='8' || tempUttrykk
.charAt(sjekkPlass)=='9' || tempUttrykk.charAt(sjekkPlass)=='0'){
        return true;
    }//end if
    else{
        return false;
    }//end else
} //end sjekkEtterTall

public String fakultet(String uttrykk){

    if(negativFakultet){
        Infinity=true;
        fakultetParantes=false;
        negativFakultet=false;
        return uttrykk;
    }//end negativ fakultet
    else{
        if(uttrykk.contains("!")){ //inneholder fakultet

            for(int i=uttrykk.length()-1;i>0;i--){
                if(uttrykk.charAt(i)=='!'){
                    for(int j=i-1;j>=0;j--){
                        if(sjekkEtterOperator(j,uttrykk)){
                            if(j>0){

                                if(uttrykk.charAt(j-1)=='E'){
                                    //nothing standardform hopp

                                }

                                }//end if ==E
                            else{

                                double tempFakultet=1;
                                int
                                n=(int)gjorTilTall(uttrykk.substring(j+1,i));

                                if(n==0)n=1; //pga 0!=1
                                if(n>=1){

                                    for(int k=1;k<=n;k++){

                                        tempFakultet=tempFakultet*k;
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

if(k==100){//legger begrensning på maks 100!

DelingsVindu.utviklingPanel.infoFeltTF.setText("Fakultet over 100");

DelingsVindu.utviklingPanel.infoFeltTF.setBackground(Color.YELLOW);

k=n+100;
}
//end k-forløkke

uttrykk=uttrykk.substring(0,j+1)+tempFakultet+uttrykk.substring(i+1,uttrykk.length());
j=-1;
i=-1;
//end if n>=1
else{
Infinity=true;
return "";
}
//end else
}
//end j>0

}
//end if operator
else if(j==0){
double tempFakultet=1;
int n=(int)gjorTilTall(uttrykk.substring(0,i));

if(n==0)n=1; //pga 0!=1

if(n>=1){
for(int k=1;k<=n;k++){

tempFakultet=tempFakultet*k;

if(k==100){//legger
begrensning på maks 100!

DelingsVindu.utviklingPanel.infoFeltTF.setText("Fakultet over 100");

DelingsVindu.utviklingPanel.infoFeltTF.setBackground(Color.YELLOW);

k=n+100;
}
//end k-forløkke

uttrykk=tempFakultet+uttrykk.substring(i+1,uttrykk.length());
j=-1;
i=-1;
}
//end if n>=1

```

```

else{
    Infinity=true;
    return "";
} //end else

} //end else iff j==0

} //end j-forløkke
} //end charAT=='!'
} //end i-forløkke
} //end contains "!"
} //else

return uttrykk;
} //end fakultet

public void nullstill(){
    //nullstiller alt
    hvilketUttrykk=-3;
    xskala=ykala=1;
    Xintervall=Yintervall=50;
    origoX=height/2;
    origoY=width/2;
    tallPa=true;
    rutenettPa=true;
    tegnEpsilon=false;
    tegnGrenseFunksjon=false;
    epsilonVerdi=0.0;
    grensefunksjon="";
    antallRekkerSomSkalVises=DelingsVindu.utviklingPanel.antallrekker;
    repaint();
} //end minus

public void tallVekk(boolean pa){
    //setter tall av
    if(pa)tallPa=false;
    else tallPa=true;
    repaint();
} //end tallvekk

public void rutenettVekk(boolean pa){
    //setter rutenett av
    if(pa)rutenettPa=false;
    else rutenettPa=true;
    repaint();
} //end rutenettvekk

public void settOrigo(String XMIN,String XMAX, String XSKALA,String YMIN,String YMAX, String YSKALA){
    //får inn tekst fra innstillingsvinduet...

```

```

//nullstiller
Xintervall=50;
Yintervall=50;
xskala=1;
yskala=1;
//gjør det om til tall og sender de videre...

settOrigo(gjørTilTall(XMIN),gjørTilTall(XMAX),gjørTilTall(XSKALA),gjørTilTall(YMIN),gjørTilTall(YMAX),gjørTilTall(
YSKALA));
repaint();

} //end settOrigo

public void settOrigo(double XMIN,double XMAX, double XSKALA,double YMIN,double YMAX, double
YSKALA){
xskala=XSKALA;
yskala=YSKALA;

//setter xretning
if(XMIN>=0){
Xintervall=(int)(width/((XMAX-XMIN)/xskala));
origoY=(int)(-XMIN*(width/(XMAX-XMIN)));
} //end if
else if(XMAX<=0){
Xintervall=(int)(width/((XMIN-XMAX)/xskala));
origoY=width+(int)(XMAX*(width/(XMIN-XMAX)));
} //end else if
else{
Xintervall=(int)(width/((XMAX-XMIN)/xskala));
origoY=(int)(Math.abs(XMIN)*(width/(XMAX+Math.abs(XMIN))));
} //end else

//setter y-retning
if(YMIN>0){
Yintervall=(int)(height/((YMIN-YMAX)/yskala));
origoX=height+(int)(YMIN*(height/(YMAX-YMIN)));
} //end if
else if(YMAX<=0){
Yintervall=(int)(height/((YMAX-YMIN)/yskala));
origoX=-(int)(YMAX*(height/(YMIN-YMAX)));
} //end else if
else{
Yintervall=(int)(height/((YMAX-YMIN)/yskala));
origoX=(int)(Math.abs(YMAX)*(height/(YMAX+Math.abs(YMIN))));
} //end else

Xintervall=Math.abs(Xintervall);
Yintervall=Math.abs(Yintervall);
xMin=XMIN;

```

```

xMax=XMAX;
yMin=YMIN;
yMax=YMAX;

} //end rutenettVekk

public boolean handleEvent(Event evt){
    //drar med musen i koordinatsystemet

    Graphics g = getGraphics();
    if(evt.id== Event.MOUSE_DOWN) {
        xNed=evt.x;
        yNed=evt.y;
        setCursor(Cursor.getPredefinedCursor(1)); //cursor lik kryss

    } //end if mousedown
    if(evt.id== Event.MOUSE_DRAG){
        xOpp=evt.x;
        yOpp=evt.y;
        g.setColor(Color.BLACK);
        g.drawLine(xNed,yNed,xOpp,yNed); //øvre
        g.drawLine(xNed,yNed,xNed,yOpp); //rett ned
        g.drawLine(xOpp,yNed,xOpp,yOpp); // rett ned siste kant
        g.drawLine(xOpp,yOpp,xNed,yOpp); // nederste
        repaint();
    } //end if mousedrag
    if(evt.id== Event.MOUSE_UP) {
        xOpp=evt.x;
        yOpp=evt.y;
        if(xOpp==xNed || yOpp==yNed){
            setCursor(Cursor.getDefaultCursor()); //sørger for at vinduet ikke forandres ved
            klikk..men man må dra vinduet
        } //end if
        else{
            setCursor(Cursor.getDefaultCursor());
            xMin=(double)((xNed-origoY)/Xintervall*xskala);
            xMax=(double)((xOpp-origoY)/Xintervall*xskala);
            yMin=-1*(double)((yNed-origoX)/Yintervall);
            yMax=-1*(double)((yOpp-origoX)/Yintervall);

            if(xMin>xMax){ //bytt rekkefølge, merket fra høyre til venstre
                double temp=xMax;
                xMax=xMin;
                xMin=temp;
            } //end if

            if(yMin>yMax){ //bytt rekkefølge, merket fra høyre til venstre
                double temp=yMax;
                yMax=yMin;
            } //end if
        }
    }
}

```

```

        yMin=temp;
    }//end if

    settOrigo(xMin,xMax,1,yMin,yMax,1);
    xNed=xOpp=yNed=yOpp=0;
    repaint();
} //end else
} //end if mouse up

return true;
} //end handleEvent

} //end

```

8.2.3 Klassen: StillNVerdiVindu

```

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

class StillNVerdiVindu extends Frame implements ActionListener{

    //variable deklarasjon
    private int width;
    private int height;
    HovedVindu hovedvindu;
    private TextField nVerdiTF;
    private TextField nStartVerdiTF;
    private Button godkjennVerdier;
    private Button avbryt;
    private Color fargePaKnapp;
    private String ulovlig;
    private char ulovligTegn[];

    private int StartOkN;

    public StillNVerdiVindu(String s){
        //vindu for å stille n-verdien
        super(s);
        Toolkit toolkit = Toolkit.getDefaultToolkit();
        Dimension dimension = toolkit.getScreenSize();
        width = 300;
        height = 150;
        setSize(width, height);
        setLocation((int)(dimension.getSize().getWidth()-2*width),0);

        //legger på grunnpanelet
        Panel grunnPanel = new Panel();
    }
}

```

```

setBackground(new Color(255,255,255));
add(grunnPanel);

    Panel panel = new Panel();
    panel.setLayout(new GridLayout(3, 2));
    setBackground(new Color(255,255,255));

//innstillinger for økning i nverdi
panel.add(new Label("Økning i n-verdi :"));
// StartOkN=DelingsVindu.utviklingPanel.nVerdi;
nVerdiTF = new TextField(""+(DelingsVindu.utviklingPanel.nVerdi), 10);
panel.add(nVerdiTF);

//innstilling for startverdi av n
panel.add(new Label("Start verdi på n :"));
nStartVerdiTF = new TextField("0", 10);

panel.add(nStartVerdiTF);

//buttons
Button godkjennVerdier = new Button("Sett verdier");
Button avbryt = new Button("Avbryt");
fargePaKnapp = new Color(200, 0, 100);
godkjennVerdier.setBackground(fargePaKnapp);
avbryt.setBackground(fargePaKnapp);
panel.add(avbryt);
panel.add(godkjennVerdier);
grunnPanel.add(panel);
    godkjennVerdier.addActionListener(this);
    avbryt.addActionListener(this);

    grunnPanel.add(panel);
HovedVindu hovedvindu = new HovedVindu(this);
addWindowListener(hovedvindu);
setVisible(true);

ulovlig = new String
("AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZzÆæØøÅå<>:;?+*/^(&%# $"); //ulovlige
tegn...
        ulovligTegn = new char[ulovlig.length()];
        for(int i=0;i<ulovlig.length();i++){
            ulovligTegn[i]=ulovlig.charAt(i);
        }

//end DelingsVindu

public void actionPerformed(ActionEvent actionevent){
//variable

```



```

        return sjekk;
    } //end ulovlig verdi

} //end

```

8.2.4 Klassen: UtviklingPanel

```

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

class UtviklingPanel extends Panel implements ActionListener{
    //variable
    protected Button neste;
    protected Button gjett;
    protected Label[] uttrykk;//legger inn uttrykkene i en liste...
    protected int antallrekker; //maksimalt antall rekker. regnes ut automatisk
    protected int hvilkenPosisjon; //hvilket uttrykk som skla tegnes
    protected Color[] fargeKoder; //fargekode på funksjonene og uttrykket, antallet varierer etter
    størrelsen på vinduet
    private Color[] FerdigefargeKoder; //en liste med fargekoder som legges inn til fargekoder.
    mak 25 stk her
    protected TextField funksjonsUttrykk; //brukeren skriver inn uttrykket i tekstfeltet
    protected TextField infoFeltTF; //infofeltet
    private String tempUttrykk; //midlertidig variabel for å sjekke uttrykket som skrives
    inn
    protected int nVerdi; //verdien på n som skal erstatte n
    private int gammelVerdi; //den gamle n verdien
    protected int nStartVerdi; //startverdi på n
    private boolean nStartBoolean; //en boolk som gjør at noen verdier settes bare en gang
    private String ulovlig; //alle de ulovlige tegnene
    private char ulovligTegn[]; //liste over de ulovlige tegnene blir generert ut fra stringen over
    protected GjettGrenseFunksjonsVindu gjettGrenseFunksjonsVindu;
    boolean nFunnet; //variabel for å si ifra om gyldig n verdi ble funnet

    public UtviklingPanel(){

        //init variable
        nFunnet=false;
        nVerdi=1;
        nStartVerdi=0;
        gammelVerdi=nStartVerdi-nVerdi;
        Toolkit toolkit = Toolkit.getDefaultToolkit();
        Dimension dimension = toolkit.getScreenSize();
        antallrekker=(int)((((dimension.getHeight()/1.41)/24.9)-6));
        //disse tallene er rare men er forholdet mellom størrelsen av delvinduer...
        //1.41 er forholdet mellom hele skjermvinduet...
        //24.9 er ca textboks størrelsen
    }
}

```

```

// 6 er siden det er 4 plasser over utviklingen og 2 plasser nederst som "forsvinner" til andre formål...
if(antallrekker>25)antallrekker=25;//max antall rekkeutviklinger blir satt til 25

    hvilkenPosisjon=0;
    nStartBoolean=true;
    Font skriftType = new Font("Verdana", 1, 14);
    Color fargeTekstboks = new Color(255,255,255);

//panel
Panel utvikling = new Panel();
utvikling.setLayout(new GridLayout(antallrekker+6,1)); //20 rader , 1 kollone
add(utvikling);

//legger til tekst og uttrykksfelt
    Label overskrift = new Label("Funksjonsfølger",0);
overskrift.setFont(skriftType);
utvikling.add(overskrift);

    Label overskrift2 = new Label("Skriv inn funksjonsfølgen",0);
overskrift2.setFont(skriftType);
utvikling.add(overskrift2);

    funksjonsUttrykk = new TextField("", 0);
funksjonsUttrykk.setFont(skriftType);
funksjonsUttrykk.setBackground(fargeTekstboks);
utvikling.add(funksjonsUttrykk);

//knappen tegn graf
neste = new Button("Tegn graf");
Color fargePaKnapp = new Color(200, 0, 100);
neste.setBackground(fargePaKnapp);
utvikling.add(neste);
neste.addActionListener(this);

uttrykk = new Label[antallrekker]; //lager plass til funksjonsuttrykkene
    for(int i=0;i<antallrekker;i++){
        uttrykk[i]=new Label("",0); //setter verdien lik ""

        utvikling.add(uttrykk[i]);
    }//end for-løkke

    //gjett grensefunksjon blir lagt til...
    gjett = new Button("Gjett grensefunksjon");
gjett.setBackground(fargePaKnapp);
utvikling.add(gjett);
gjett.addActionListener(this);

//legger fargekoder på...og finn ordentlige farger
ferdiggodkjentefarger();

```

```

        settFarger();

        //lager sjekklister for ulovlige tegn
        ulovlig = new String ("BbDdFfJjKkMmQqRrUuVvWwYyZzÆæØøÅå<>:;?&%#|$_~"); //ulovlige
tegn...
        ulovligTegn = new char[ulovlig.length()];

        //opprettet tabell med de ulovlige tegnene
        for(int i=0;i<ulovlig.length();i++){
            ulovligTegn[i]=ulovlig.charAt(i);
        }

        //gjett infofelt blir lagt til...
        infoFeltTF = new TextField("Informasjonsfelt", 0);
        infoFeltTF.setFont(skriftType);
        infoFeltTF.setBackground(fargeTekstboks);
        infoFeltTF.disable();
        utvikling.add(infoFeltTF);

    } //end UtviklingsPanel

    public void ferdiggodkjentefarger(){
        FerdigefargeKoder=new Color[25];
        FerdigefargeKoder[0]=new Color (128,0,64);
        FerdigefargeKoder[1]=new Color (128,255,128);
        FerdigefargeKoder[2]=new Color (0,128,255);
        FerdigefargeKoder[3]=new Color (255,128,64);
        FerdigefargeKoder[4]=new Color (0,128,128);
        FerdigefargeKoder[5]=new Color (128,0,128);
        FerdigefargeKoder[6]=new Color (128,255,255);
        FerdigefargeKoder[7]=new Color (128,128,192);
        FerdigefargeKoder[8]=new Color (255,0,255);
        FerdigefargeKoder[9]=new Color (128,128,0);
        FerdigefargeKoder[10]=new Color (64,128,128);
        FerdigefargeKoder[11]=new Color (64,0,64);
        FerdigefargeKoder[12]=new Color (0,255,255);
        FerdigefargeKoder[13]=new Color (0,128,192);
        FerdigefargeKoder[14]=new Color (128,64,64);
        FerdigefargeKoder[15]=new Color (0,64,128);
        FerdigefargeKoder[16]=new Color (255,0,128);
        FerdigefargeKoder[17]=new Color (255,128,0);
        FerdigefargeKoder[18]=new Color (255,128,255);
        FerdigefargeKoder[19]=new Color (64,128,128);
        FerdigefargeKoder[20]=new Color (0,255,128);
        FerdigefargeKoder[21]=new Color (255,128,128);
        FerdigefargeKoder[22]=new Color (255,0,128);
        FerdigefargeKoder[23]=new Color (192,192,192);
        FerdigefargeKoder[24]=new Color (128,64,0);
    } //end settFarger

```

```

public void settFarger(){
    fargeKoder = new Color[antallrekker];
    for(int i=0;i<antallrekker;i++) fargeKoder[i]=FerdigefargeKoder[i];
}

public boolean tallForran(int plass){
    if(sjekkEtterTall(plass)) return true;
    else return false;
}

public String StringTest(String test){
    //rutine for å sjekke det innskrevnet uttrykket...
    //fjerner + - * / i starten og på slutten
    //erstatte , med . osv...
    //variable
    int parantesTeller=0;

    //sjekker etter feil parentesbruk...
    for(int i=0;i<test.length();i++){
        if(test.charAt(i)=='(')parantesTeller++; //teller ( parenteser
        if(test.charAt(i)==')')parantesTeller--; //teller ) parenteser
        if(parantesTeller<0){
            infoFeltTF.setText("Feil med parenter");
            infoFeltTF.setBackground(Color.RED);
            i=test.length()+50;
        } //end if <0
    } //end i-forløkke
    if(parantesTeller!=0){
        infoFeltTF.setText("Feil med parenter");
        infoFeltTF.setBackground(Color.RED);
        neste.setLabel("Start på nytt");
        tempUttrykk="";
        return "";
    } //end if !=0

    //fjerner matematiske operatører på slutt og start, men ikke - i starten
    if(test.charAt(test.length()-1)=='+' || test.charAt(test.length()-1)=='-' || test.charAt(test.length()-1)=='*' || test.charAt(test.length()-1)=='/' || test.charAt(test.length()-1)=='^' || test.charAt(test.length()-1)=='('){
        test=StringTest(test.substring(0,test.length()-1));
    } //end if
    if(test.charAt(0)=='*' || test.charAt(0)=='/' || test.charAt(0)=='+' || test.charAt(0)=='^' || test.charAt(0)=='('){
        test=StringTest(test.substring(1,test.length()));
    } //end if

    for(int i=0;i<test.length();i++){
        if(test.charAt(i)=='&&i<(test.length()-1){ //bytter , med .
            test="" +test.substring(0,i)+ "." +test.substring(i+1,test.length());
        }
    }
}

```

```

        //end if test.charAt(i)=='&&i<(test.length()-1)
        if(test.charAt(i)==' '){// fjerner mellomrom
            test="" +test.substring(0,i)+test.substring(i+1,test.length());
        }//end if test.charAt(i)=='&&i<(test.length()-1)
    }//end i-forløkke

    return ulovligVerdi(test);
} //end StringTest

public void sjekkPotens(){
    //sette potenser riktig... fra f.eks. x^2n til x^(2n)
    if(tempUttrykk.contains("^")){
        for(int i=0;i<tempUttrykk.length();i++){
            if(tempUttrykk.charAt(i)=='^'){ //fant potensen
                if(i<tempUttrykk.length()-1){//sjekk at den ikke er på slutten for sikkerhet
                    for(int j=i+1;j<tempUttrykk.length();j++){//må finne slutten
                        if(tempUttrykk.charAt(i+1)=='('){//parantes etter ^ da er
det antatt ferdig

                            j=tempUttrykk.length()+50;
                        } //end if == (
                        else
if((tempUttrykk.charAt(j)=='+' || tempUttrykk.charAt(j)=='-'
' || tempUttrykk.charAt(j)=='*' || tempUttrykk.charAt(j)=='/' || tempUttrykk.charAt(j)=='^' || tempUttrykk.charAt(
j)=='(' || tempUttrykk.charAt(j)=='')&&j>i+1){

                            tempUttrykk=tempUttrykk.substring(0,i+1)+"("+tempUttrykk.substring(i+1,j)+")"+tempUttrykk.substri
ng(j,tempUttrykk.length());

                            i=0;//leit på nytt for å være sikker
                            j=tempUttrykk.length()+50;
                        } //end else if operator
                        else if(j==tempUttrykk.length()-1){

                            tempUttrykk=tempUttrykk.substring(0,i+1)+"("+tempUttrykk.substring(i+1,tempUttrykk.length()+")");
                            i=tempUttrykk.length()+50;
                            j=tempUttrykk.length()+50;//siden det blir lagt til
litt bokstaver blir lengden større enn før

                                } //end else if length-1
                            } //end j-forløkke
                        } //end if length-1
                    } //end if == ^
                } //end i-forløkke
            } //end if contains "^"
        } //end sjekk potens

    }

public boolean sjekkEtterTall(int sjekkPlass){
    //rutine for å sjekke om det er tall på plass "sjekkplass" i uttrykket tempUttrykk som er
funksjonssuttrykket men med ekstra paranteser for å få det rett
    //retunerer true hvis tall...false hvis ikke tall...

```

```

        if(tempUttrykk.charAt(sjekkPlass)=='1' || tempUttrykk.charAt(sjekkPlass)=='2' || tempUttrykk.charAt(sj
ekkkPlass)=='3' || tempUttrykk.charAt(sjekkPlass)=='4' || tempUttrykk.charAt(sjekkPlass)=='5' || tempUttrykk.char
At(sjekkPlass)=='6' || tempUttrykk.charAt(sjekkPlass)=='7' || tempUttrykk.charAt(sjekkPlass)=='8' || tempUttrykk
.charAt(sjekkPlass)=='9' || tempUttrykk.charAt(sjekkPlass)=='0'){
            return true;
        } //end if
        else{
            return false;
        } //end else
    } //end sjekkEtterTall

public boolean sjekkEtterOperator(int sjekkPlass){
    //rutine for å sjekke om det er operator på plass "sjekkplass" i uttrykket tempUttrykk
    //retunerer true hvis operator...false hvis ikke operator...
    if(tempUttrykk.charAt(sjekkPlass)=='+' || tempUttrykk.charAt(sjekkPlass)=='-'
' || tempUttrykk.charAt(sjekkPlass)=='*' || tempUttrykk.charAt(sjekkPlass)=='/' || tempUttrykk.charAt(sjekkPlass)
=='^' || tempUttrykk.charAt(sjekkPlass)=='(' || tempUttrykk.charAt(sjekkPlass)==''){
        return true;
    } //end if
    else{
        return false;
    } //end else
} //end sjekkEtterOperator

public boolean sjekkEtterX(int sjekkPlass){
    //rutine for å sjekke om det er 'x' på plass "sjekkplass" i uttrykket tempUttrykk
    //retunerer true hvis 'x'...false hvis ikke 'x'...

    if(tempUttrykk.charAt(sjekkPlass)=='x' || tempUttrykk.charAt(sjekkPlass)=='X'){
        return true;
    } //end if
    else{
        return false;
    } //end else
} //end sjekkEtterTall

public boolean sjekkEtterN(int sjekkPlass){
    //rutine for å sjekke om det er 'n' på plass "sjekkplass" i uttrykket tempUttrykk
    //retunerer true hvis 'n'...false hvis ikke 'n'...

    if(tempUttrykk.charAt(sjekkPlass)=='n' || tempUttrykk.charAt(sjekkPlass)=='N'){
        return true;
    } //end if
    else{
        return false;
    } //end else
} //end sjekkEtterTall

```

```

public boolean sjekkSinTanLn(int plass, String sjekk){
    //har funnet en 'n' i uttrykket...sjekker om det er n eller en del av Ln eller sin eller tan
    if(plass<2&&plass>=1){
        if(sjekk.charAt(plass-1)=='l' | sjekk.charAt(plass-1)=='L'){
            return true;
        }//end l | L
    }//end plass<2
    else if(plass>=2){
        if((sjekk.charAt(plass-2)=='s' | sjekk.charAt(plass-2)=='S'&&sjekk.charAt(plass-
1)=='i' | sjekk.charAt(plass-1)=='l')){
            if(plass>2){ //sjekker etter tall forran sin
                if(tallForran(plass-3)){
                    tempUttrykk=tempUttrykk.substring(0,plass-
2)+"*"+tempUttrykk.substring(plass-2,tempUttrykk.length());
                }//end if plass-3
            }//end if plass>2
            return true;
        }// end sin
        else if((sjekk.charAt(plass-2)=='t' | sjekk.charAt(plass-2)=='T'&&sjekk.charAt(plass-
1)=='a' | | sjekk.charAt(plass-1)=='A')){
            if(plass>2){ //sjekker etter tall forran tan
                if(tallForran(plass-3)){
                    tempUttrykk=tempUttrykk.substring(0,plass-
2)+"*"+tempUttrykk.substring(plass-2,tempUttrykk.length());
                }//end plass -3
            }//end plass >2
            return true;
        }//end tan
        else if(sjekk.charAt(plass-1)=='l' | sjekk.charAt(plass-1)=='L'){
            if(plass>1){ //sjekker etter tall forran sin
                if(tallForran(plass-2)){
                    tempUttrykk=tempUttrykk.substring(0,plass-
1)+"*"+tempUttrykk.substring(plass-1,tempUttrykk.length());
                }//end plass-2
            }//end plass>1
            return true;
        }//end Ln
        else
            return false;
    }//end else
return false;
} //end sjekk SinTanLn

```

```

public void sjekkFakultet(){
    /// sjekker etter fakultet
    if(tempUttrykk.contains("!")){
        for(int i=tempUttrykk.length()-1;i>=0;i--){
            if(tempUttrykk.charAt(i)=='!'){
                for(int j=i-1;j>=0;j--){ // finne slutten på fakulteten...

```

```

        if(tempUttrykk.charAt(j)=='&&j==(i-1)){
            break;
        }//end if
        else if(tempUttrykk.charAt(j)=='+' | tempUttrykk.charAt(j)=='-
'| tempUttrykk.charAt(j)=='*' | tempUttrykk.charAt(j)=='/' | tempUttrykk.charAt(j)=='^' | tempUttrykk.charAt(
j)=='(' | tempUttrykk.charAt(j)=='')){

            tempUttrykk=tempUttrykk.subSequence(0,j+1)+"("+tempUttrykk.substring(j+1,i)+")"+tempUttrykk.su
bstring(i,tempUttrykk.length());

            j=0;
        }//end else if
        else if (j==0){

tempUttrykk="("+tempUttrykk.substring(0,i)+")"+tempUttrykk.substring(i,tempUttrykk.length());
            }//end j=0
        }//end j-forløkke
    }//end if
} //end i-forløkke
} //end contains"!"
} //end sjekkfakultet

public void sjekkEtterNverdi(){
    //rutine for å sjekke etter n verdi og erstatte den med riktig verdi
    for(int i=tempUttrykk.length()-1;i>=0;i--){
        if(tempUttrykk.charAt(i)=='n' | tempUttrykk.charAt(i)=='N'){ //fant n sjekk om den
er "n" eller endel av si"n"...
            if(i>1){
                if(tempUttrykk.charAt(i-1)=='l' | tempUttrykk.charAt(i-1)=='L'){
                    //nothing
                }//end if
                else if(tempUttrykk.charAt(i-2)=='S' | tempUttrykk.charAt(i-
2)=='s' | tempUttrykk.charAt(i-2)=='l' | tempUttrykk.charAt(i-2)=='i'){
                    //nothing
                }//end else if
                else if(tempUttrykk.charAt(i-2)=='T' | tempUttrykk.charAt(i-
2)=='t' | tempUttrykk.charAt(i-2)=='A' | tempUttrykk.charAt(i-2)=='a'){
                    //nothing
                }//end else if
                else{
                    nFunnet=true;

                    if(i>0){ //ser om jeg må sette gangetegn forran n
                        if(tempUttrykk.charAt(i-1)=='') | sjekkEtterX(i-
1) | sjekkEtterTall(i-1) | tempUttrykk.charAt(i-1)=='N' | tempUttrykk.charAt(i-1)=='n'){

                            tempUttrykk=tempUttrykk.substring(0,i)+"*"+(gammelVerdi+nVerdi)+tempUttrykk.substring(i+1,tem
pUttrykk.length());

                                i=tempUttrykk.length();
                            }//end if

```

```

else{

tempUttrykk=tempUttrykk.substring(0,i)+(gammelVerdi+nVerdi)+tempUttrykk.substring(i+1,tempUttrykk.length());

i=tempUttrykk.length();

} //end else

} //end if i>0
if(i<tempUttrykk.length()-1){ //ser om jeg må sette
gangetegn bak n

if(tempUttrykk.charAt(i+1)=='|' | sjekkEtterX(i+1) | sjekkEtterTall(i+1) | tempUttrykk.charAt(i+1)=='N' | tempUttrykk.charAt(i+1)=='n'){

tempUttrykk=tempUttrykk.substring(0,i+1)+"*"+(gammelVerdi+nVerdi)+tempUttrykk.substring(i+2,tempUttrykk.length());

i=tempUttrykk.length();
} //end if
else{

tempUttrykk=tempUttrykk.substring(0,i+1)+(gammelVerdi+nVerdi)+tempUttrykk.substring(i+2,tempUttrykk.length());

i=tempUttrykk.length();
} //end else

} //end iof length -1

} //end else
} //end if >1
else if(i==1){
if(tempUttrykk.charAt(i-1)=='|' | tempUttrykk.charAt(i-1)=='L'){
//nothing
} //end if == L
else{ //ser om jeg må sette gangetegn forran n
nFunnet=true;
if(tempUttrykk.charAt(i-1)=='|' | sjekkEtterX(i-1) | sjekkEtterTall(i-1) | tempUttrykk.charAt(i-1)=='N' | tempUttrykk.charAt(i-1)=='n'){

tempUttrykk=tempUttrykk.substring(0,i)+"*"+(gammelVerdi+nVerdi)+tempUttrykk.substring(i+1,tempUttrykk.length());

i=tempUttrykk.length();
} //end if
else{

tempUttrykk=tempUttrykk.substring(0,i)+(gammelVerdi+nVerdi)+tempUttrykk.substring(i+1,tempUttrykk.length());

i=tempUttrykk.length();
}
}
}
}

```

```

        }//end else

        }//end else
    }//end else if <=1
    else if(i==0){
        nFunnet=true;

if(sjekkEtterX(i+1) | |sjekkEtterTall(i+1) | |tempUttrykk.charAt(i+1)=='N' | |tempUttrykk.charAt(i+1)=='n'){

    tempUttrykk=(gammelVerdi+nVerdi)+"*"+tempUttrykk.substring(i+1,tempUttrykk.length());

        }//end if
        else{

tempUttrykk=(gammelVerdi+nVerdi)+tempUttrykk.substring(i+1,tempUttrykk.length());
        }//end else
        }//end else if ==0
    }// end if == N
} //end i-forløkke
} //end sjekkforranBakNverdi

public void sjekkForanBakXverdi(){
    //sjekker forran og bak x verdier for å sette * tegn
    for(int i=tempUttrykk.length()-1;i>=0;i--){
        if(tempUttrykk.charAt(i)=='x' | |tempUttrykk.charAt(i)=='X'){ //fant x sjekk om den
trenger gangetegn forran...eller bak
            if(i>1){
                if(i>0){ //ser om jeg må sette gangetegn forran n
                    if(tempUttrykk.charAt(i-1)==' ' | |sjekkEtterX(i-
1) | |sjekkEtterTall(i-1)){

tempUttrykk=tempUttrykk.substring(0,i)+"*"+tempUttrykk.substring(i,tempUttrykk.length());
                    i=tempUttrykk.length();
                }//end if
            }//end if i>0
            if(i<tempUttrykk.length()-1){ //ser om jeg må sette gangetegn bak
n

if(tempUttrykk.charAt(i+1)=='(' | |sjekkEtterX(i+1) | |sjekkEtterTall(i+1)){

tempUttrykk=tempUttrykk.substring(0,i+1)+"*"+tempUttrykk.substring(i+1,tempUttrykk.length());

                i=tempUttrykk.length();
            }//end if
        }//end if lenght -1
    }//end if >1
    else if(i==1){
        if(tempUttrykk.charAt(i-1)==' ' | |sjekkEtterX(i-1) | |sjekkEtterTall(i-
1)){

```

```

tempUttrykk=tempUttrykk.substring(0,i)+"*" +tempUttrykk.substring(i,tempUttrykk.length());
                                i=tempUttrykk.length();
                                }//end if
                                }//end else if <=1
                                else{

if(tempUttrykk.charAt(i+1)=='(' | sjekkEtterX(i+1) | sjekkEtterTall(i+1)){

tempUttrykk=tempUttrykk.substring(0,i+1)+"*" +tempUttrykk.substring(i+1,tempUttrykk.length());

                                }//end if
                                }//end else
                                }//end if == X
                                }//end i-forløkke
                                }//end sjekkforranBakNverdi

public void sinCosTanLnLogParantesSjekk(){
    //rutine for å legger til parenteser etter f.eks. sin3x til sin(3x) ...
    //samt sjekker om det må settes inn gangetegn før sin f.eks. 3sinx = 3*sinx...
    //sinx+lnx+tanx+n

    // leiter etter sin
    if(tempUttrykk.contains("sin")){
        for(int i=2;i<tempUttrykk.length()-1;i++){
            if((tempUttrykk.charAt(i-2)=='S' | tempUttrykk.charAt(i-
2)=='s')&&(tempUttrykk.charAt(i-1)=='(' | tempUttrykk.charAt(i-
1)=='i')&&(tempUttrykk.charAt(i)=='N' | tempUttrykk.charAt(i)=='n')){
                //legger til parentes rundt sinusuttrykket

                if(i<tempUttrykk.length()-2){
                    if(tempUttrykk.charAt(i+1)=='h')i++;//hyperbolsk
sin

                }//end if
                if((tempUttrykk.charAt(i+1)=='(')){

                    //nothing, "riktig" parentesbruk

                }//end if i+1==(
                else{
                    for(int j=i+1;j<tempUttrykk.length();j++){

                        if(sjekkEtterOperator(j)&&j!=i+1){

                            tempUttrykk=tempUttrykk.substring(0,i+1)+"(" +tempUttrykk.substring(i+1,j)+"") +tempUttrykk.substri
ng(j,tempUttrykk.length());

                                i=1; //start på nytt i i løkka

```

```

        j=tempUttrykk.length()+50;

        //end sjekk operator
        else if(j==tempUttrykk.length()-1){

tempUttrykk=tempUttrykk.substring(0,i+1)+"("+tempUttrykk.substring(i+1,tempUttrykk.length()+1)";

        i=j=tempUttrykk.length()+50;
        //end length-1

        //end j-forløkke
        //end else
        //end sjekk etter sin

        //end i-forløkke

        //sjekker om det må settes gangetegn før sin
        for(int i=3;i<tempUttrykk.length()-1;i++){
            if(((tempUttrykk.charAt(i-2)=='S' | tempUttrykk.charAt(i-2)=='s')&&
(tempUttrykk.charAt(i-1)=='|' | tempUttrykk.charAt(i-1)=='i')&&
(tempUttrykk.charAt(i)=='N' | tempUttrykk.charAt(i)=='n'))&&(sjekkEtterN(i-3) | sjekkEtterX(i-
3) | sjekkEtterTall(i-3) | tempUttrykk.charAt(i-3)=='')){
                //legger til gange...
                tempUttrykk=tempUttrykk.substring(0,i-
2)+"*"+tempUttrykk.substring(i-2,tempUttrykk.length());
                i=2;

            //end if
        //end i forløkke
    //end sinus

    //leiter etter cos
    if(tempUttrykk.contains("cos")){
        for(int i=2;i<tempUttrykk.length()-1;i++){
            if((tempUttrykk.charAt(i-2)=='C' | tempUttrykk.charAt(i-
2)=='c')&&(tempUttrykk.charAt(i-1)=='O' | tempUttrykk.charAt(i-
1)=='o')&&(tempUttrykk.charAt(i)=='S' | tempUttrykk.charAt(i)=='s')){
                //legger til parantes rundt cosinusuttrykket
                if(i<tempUttrykk.length()-2){
                    if(tempUttrykk.charAt(i+1)=='h')i++;

//hyperbolsk cos

                //end if
                if((tempUttrykk.charAt(i+1)=='(')){
                    //nothing

                //end if i+1==(
                else{
                    for(int j=i+1;j<tempUttrykk.length();j++){

```

```

        if(sjekkEtterOperator(j)&&j!=i+1){

            tempUttrykk=tempUttrykk.substring(0,i+1)+"("+tempUttrykk.substring(i+1,j)+")"+tempUttrykk.substri
ng(j,tempUttrykk.length());

            i=1;
            j=tempUttrykk.length()+50;

            //end sjekk operator
            else if(j==tempUttrykk.length()-1){

                tempUttrykk=tempUttrykk.substring(0,i+1)+"("+tempUttrykk.substring(i+1,tempUttrykk.length()+1)+")";

                i=j=tempUttrykk.length()+50;
                //end length-1
            } //end j-forløkke
        } //end else
    } //end sjekk etter cos
} //end i-forløkke

//sjekker om det må settes gangetegn før cos
for(int i=3;i<tempUttrykk.length()-1;i++){
    if(((tempUttrykk.charAt(i-2)=='C' | tempUttrykk.charAt(i-2)=='c')&&
(tempUttrykk.charAt(i-1)=='O' | tempUttrykk.charAt(i-1)=='o')&&
(tempUttrykk.charAt(i)=='S' | tempUttrykk.charAt(i)=='s'))&&(sjekkEtterN(i-3) | sjekkEtterX(i-
3) | sjekkEtterTall(i-3) | tempUttrykk.charAt(i-3)=='')){
        //legger til gange...
        tempUttrykk=tempUttrykk.substring(0,i-
2)+"*"+tempUttrykk.substring(i-2,tempUttrykk.length());
        i=2;
    } //end if
} //end i forløkke
} //end cosinus

//leier etter tan
if(tempUttrykk.contains("tan")){
    for(int i=2;i<tempUttrykk.length()-1;i++){
        if((tempUttrykk.charAt(i-2)=='T' | tempUttrykk.charAt(i-
2)=='t')&&(tempUttrykk.charAt(i-1)=='A' | tempUttrykk.charAt(i-
1)=='a')&&(tempUttrykk.charAt(i)=='N' | tempUttrykk.charAt(i)=='n')){
            //legger til parentes rundt tangensuttrykket
            if(i<tempUttrykk.length()-2){
                if(tempUttrykk.charAt(i+1)=='h')i++; //hyperbolsk
            }
            tan
        } //end if
        if((tempUttrykk.charAt(i+1)=='(')){
            //nothing
        }
    }
}

```

```

        }//end if i+1==(
        else{
            for(int j=i+1;j<tempUttrykk.length();j++){
                if(sjekkEtterOperator(j)&&j!=i+1){

tempUttrykk=tempUttrykk.substring(0,i+1)+"("+tempUttrykk.substring(i+1,j)+")"+tempUttrykk.substri
ng(j,tempUttrykk.length());

                    i=1;
                    j=tempUttrykk.length()+50;

                }//end sjekk operator
                else if(j==tempUttrykk.length()-1){

tempUttrykk=tempUttrykk.substring(0,i+1)+"("+tempUttrykk.substring(i+1,tempUttrykk.length()+1)+")";

                    i=j=tempUttrykk.length()+50;
                }//end length-1
            }//end j-forløkke
        }//end else

    }//end sjekk etter tan

    }//end i-forløkke
    //sjekker om det må settes gangetegn før tan
    for(int i=3;i<tempUttrykk.length()-1;i++){
        if(((tempUttrykk.charAt(i-2)=='T' | tempUttrykk.charAt(i-2)=='t')&&
(tempUttrykk.charAt(i-1)=='A' | tempUttrykk.charAt(i-1)=='a')&&
(tempUttrykk.charAt(i)=='N' | tempUttrykk.charAt(i)=='n'))&&(sjekkEtterN(i-3) | sjekkEtterX(i-
3) | sjekkEtterTall(i-3) | tempUttrykk.charAt(i-3)=='')){
            //legger til gange...
            tempUttrykk=tempUttrykk.substring(0,i-
2)+"*"+tempUttrykk.substring(i-2,tempUttrykk.length());
            i=2;

        }//end if
    }//end i forløkke
}//end tan

//leier etter log
if(tempUttrykk.contains("log")){
    for(int i=2;i<tempUttrykk.length()-1;i++){
        if((tempUttrykk.charAt(i-2)=='L' | tempUttrykk.charAt(i-
2)=='l')&&(tempUttrykk.charAt(i-1)=='O' | tempUttrykk.charAt(i-
1)=='o')&&(tempUttrykk.charAt(i)=='G' | tempUttrykk.charAt(i)=='g')){
            //legger til parantes rundt loguttrykket
            if((tempUttrykk.charAt(i+1)=='(')){
                //nothing

            }//end if i+1==(

```

```

else{
    for(int j=i+1;j<tempUttrykk.length();j++){

        if(sjekkEtterOperator(j)&&j!=i+1){

            tempUttrykk=tempUttrykk.substring(0,i+1)+"("+tempUttrykk.substring(i+1,j)+")"+tempUttrykk.substri
ng(j,tempUttrykk.length());

            i=1;

            j=tempUttrykk.length()+50;

            //end sjekk operator
            else if(j==tempUttrykk.length()-
1){

                tempUttrykk=tempUttrykk.substring(0,i+1)+"("+tempUttrykk.substring(i+1,tempUttrykk.length()+")");

                i=j=tempUttrykk.length()+50;

                //end length-1
                //end j-forløkke
                //end else

                //end sjekk etter log
                //end i-forløkke

                //sjekker om det må settes gangetegn før log
                for(int i=3;i<tempUttrykk.length()-1;i++){
                    if(((tempUttrykk.charAt(i-2)=='L' | tempUttrykk.charAt(i-2)=='l')&&
(tempUttrykk.charAt(i-1)=='O' | tempUttrykk.charAt(i-1)=='o')&&
(tempUttrykk.charAt(i)=='G' | tempUttrykk.charAt(i)=='g'))&&(sjekkEtterN(i-3) | sjekkEtterX(i-
3) | sjekkEtterTall(i-3) | tempUttrykk.charAt(i-3)=='')){
                        //legger til gange...
                        tempUttrykk=tempUttrykk.substring(0,i-
2)+"*" +tempUttrykk.substring(i-2,tempUttrykk.length());
                        i=2;

                        //end if
                    } //end i forløkke
                } //end log

                //leiter etter ln
                if(tempUttrykk.contains("ln")){
                    for(int i=1;i<tempUttrykk.length()-1;i++){
                        if((tempUttrykk.charAt(i-1)=='L' | tempUttrykk.charAt(i-
1)=='l')&&(tempUttrykk.charAt(i)=='N' | tempUttrykk.charAt(i)=='n')){
                            //legger til parentes rundt lnuttrykket
                            if((tempUttrykk.charAt(i+1)=='(')){
                                //nothing

```

```

        } //end if i+1==(
        else{
            for(int j=i+1;j<tempUttrykk.length();j++){
                if(sjekkEtterOperator(j)&&j!=i+1){

tempUttrykk=tempUttrykk.substring(0,i+1)+"("+tempUttrykk.substring(i+1,j)+")"+tempUttrykk.substri
ng(j,tempUttrykk.length());

                    i=0;
                    j=tempUttrykk.length()+50;

                } //end sjekk operator
                else if(j==tempUttrykk.length()-1){

tempUttrykk=tempUttrykk.substring(0,i+1)+"("+tempUttrykk.substring(i+1,tempUttrykk.length()+1)+")";

                    i=j=tempUttrykk.length()+50;
                } //end length-1
            } //end j-forløkke
        } //else

    } //end sjekk etter ln
} //end i-forløkke
//sjekker om det må settes gangetegn før ln
for(int i=2;i<tempUttrykk.length()-1;i++){
    if(((tempUttrykk.charAt(i-1)=='L' | tempUttrykk.charAt(i-1)=='l')&&
(tempUttrykk.charAt(i)=='N' | tempUttrykk.charAt(i)=='n'))&&(sjekkEtterN(i-2) | sjekkEtterX(i-
2) | sjekkEtterTall(i-2) | tempUttrykk.charAt(i-2)==' ')){
        //legger til gange...
tempUttrykk=tempUttrykk.substring(0,i-
1)+"*"+tempUttrykk.substring(i-1,tempUttrykk.length());
        i=1;
    }
} //end if
} //end i forløkke
} //end ln

} //end sinCosTanLnLogParantesSjekk

public void parentessjekk(){
    //sette paranteser rett... f.eks. 3(x+2)(4+1)n til 3*(x+2)*(4+1)*n
    //men må sjekke om det er f.eks n(x+1) som skal bli n*(x+1) men ikke når det står f.eks
sin(x+1)

    if(tempUttrykk.contains("(")){ //trenger bare leite etter ( pga da må ) være med ellers får man
feil ved sjekkuttrykk rutinen.
        for(int i=0;i<tempUttrykk.length();i++){

```

```

        if(tempUttrykk.charAt(i)==''){ //sjekk etter tall ,( eller bokstav forran og
legg til *
            if(i>0){
                if(sjekkEtterN(i-1) | sjekkEtterX(i-1) | sjekkEtterTall(i-
1) | tempUttrykk.charAt(i-1)==''){
                    //sjekker om det er ln eller sin eller tan... da skal
det ikke gjøres noe..
                        if(i>2){
                            if((tempUttrykk.charAt(i-
2)=='L' | tempUttrykk.charAt(i-2)=='l') | ((tempUttrykk.charAt(i-3)=='S' | tempUttrykk.charAt(i-
3)=='s')&&(tempUttrykk.charAt(i-2)=='l' | tempUttrykk.charAt(i-2)=='i') | ((tempUttrykk.charAt(i-
3)=='T' | tempUttrykk.charAt(i-3)=='t')&&(tempUttrykk.charAt(i-2)=='A' | tempUttrykk.charAt(i-2)=='a'))){
                                //nothing
                            } //end if
                        } else{
tempUttrykk=tempUttrykk.substring(0,i)+"*" +tempUttrykk.substring(i,tempUttrykk.length());
                            i=0; //sjekk på nytt
                        } //end else
                    } //end if >2
                } else if(i>1){
                    if(tempUttrykk.charAt(i-
2)=='l' | tempUttrykk.charAt(i-2)=='L'){ //sjekker at det ikke står ln
                        //nothing
                    } //end if
                } else{
tempUttrykk=tempUttrykk.substring(0,i)+"*" +tempUttrykk.substring(i,tempUttrykk.length());
                            i=0; //sjekk på nytt
                        } //end else
                    } //end if i>=1
                } else{//bare et tall eller x eller n forran (
tempUttrykk=tempUttrykk.substring(0,i)+"*" +tempUttrykk.substring(i,tempUttrykk.length());
                            i=0; //sjekk på nytt
                        }
                    } //end if
                } //end if i>0
            } //end if == (
        if(tempUttrykk.charAt(i)==''){ //sjekk etter tall ,( eller bokstav forran og
legg til *
            if(i<tempUttrykk.length()-1){

```

```

if(sjekkEtterN(i+1) | sjekkEtterX(i+1) | sjekkEtterTall(i+1) | tempUttrykk.charAt(i+1)==''){

tempUttrykk=tempUttrykk.substring(0,i+1)+"*"+tempUttrykk.substring(i+1,tempUttrykk.length());

                                i=0; //sjekk på nytt
                                }//end if
                                }//end if i>0
                                }//end if == (
                                }//end i forløkke
                                }//end contains (
} //end parentessjekk

public void nullstill(){
    //nullstiller alt...
    hvilkenPosisjon=0;
    gammelVerdi=-1;
    nVerdi=1;
    nStartBoolean=true;
    nStartVerdi=0;
    funksjonsUttrykk.enable();
    neste.setLabel("Tegn graf");
    infoFeltTF.setText("Informasjonsfelt");
    infoFeltTF.setBackground(Color.WHITE);
    for(int i=0;i<DelingsVindu.utviklingPanel.antallrekker;i++){
        uttrykk[i].setText(null);
    } //end forløkke
    funksjonsUttrykk.setText("");
} //end nullstill

    public String ulovligVerdi(String sjekk){
//sjekker etter ulovlige verdier og fjerner dem.

        if(sjekk.length()<=1){

            for(int j=0;j<ulovlig.length();j++){
                if(sjekk.charAt(0)==ulovligTegn[j]){
                    nFunnet=false;
                    return"";

                } //end if
            } //end j-løkke
        } //bare et tegn
        else{
            //flere tegn enn 1
            for(int i=0;i<sjekk.length();i++){
                for(int j=0;j<ulovlig.length();j++){
                    if(sjekk.charAt(i)=='P' | sjekk.charAt(i)=='p'){ //sjekker etter PI og dermed
erstatte pi med (3.14...)

```

```

        if(i<sjekk.length()-1){
            if(sjekk.charAt(i+1)==' ' | sjekk.charAt(i+1)=='i'){
                if(i==0){

                    return

                }
            }
            ulovligVerdi("(" + Math.PI + ")" + sjekk.substring(i+2,sjekk.length()));
        }
        //end if
        else{
            return

        }
        ulovligVerdi(sjekk.subSequence(0,i)+("(" + Math.PI + ")" + sjekk.substring(i+2,sjekk.length()));
        //end else

        //end sjekk l
    }
    //end sjekk length -1
    else if(i==sjekk.length()-1){
        //nothing
    }
    //end else if
}
//end sjekk ==p
else if(sjekk.charAt(i)=='e'){//erstatte bokstaven e med 2.718281828.
    return
}
ulovligVerdi(sjekk.subSequence(0,i)+("(" + Math.E + ")" + sjekk.substring(i+1,sjekk.length()));
//end else if
else if(sjekk.charAt(i)==ulovligTegn[j]){
    return
}
ulovligVerdi((sjekk.substring(0,i)+sjekk.substring(i+1,sjekk.length())));

//end if
if(sjekk.charAt(i)==',') {
    if(i==sjekk.length()-1){

        return ulovligVerdi(sjekk.substring(0,i));

    }
}
//end if
else {
    return

}
ulovligVerdi(sjekk.substring(0,i)+"." + sjekk.substring(i+1,sjekk.length()));

//end else
}
//end if == ,
}
//enf j-forløkke
}
//end i-forløkke
}
//end else...flere tegn enn 1
return sjekk;
}
//end ulovlig verdi

public void sjekkUttrykk(){
    //sjekker uttrykket som er skrevet inn...

    if(funksjonsUttrykk.getText().length()==0){ //uttrykket inneholder ingen tekst

```

```

        nullstill();
        infoFeltTF.setText("Ingen gyldig n");
        infoFeltTF.setBackground(Color.RED);
        neste.setLabel("Start på nytt");
        return;
    }//end if
else{//uttrykket inneholder tekst
    nFunnet=false;
    tempUttrykk=funksjonsUttrykk.getText();
    tempUttrykk=StringTest(tempUttrykk);
    if(tempUttrykk.length()>0){
        sjekkPotens();
        sjekkFakultet();
        sinCosTanLnLogParantesSjekk();
        sjekkEtterNverdi();
        if(nFunnet){
            uttrykk[hvilkenPosisjon].setText(tempUttrykk);
            infoFeltTF.setBackground(Color.WHITE);
            infoFeltTF.setText("Tegner opp graf");
            //først for å bare bytte ut n - verdien med tall slik at uttrykket i
            utviklingsfeltet blir rett...
            nFunnet=false; //nullstiller for å være sikker.
            //sjekker på nytt for jeg lagde koden med all forandring og ønsker ikke at
            brukeren skal se alle skjulte tegn. kan forvirre...

            tempUttrykk=funksjonsUttrykk.getText();
            tempUttrykk=StringTest(tempUttrykk);
            sjekkPotens();
            parentessjekk();
            sinCosTanLnLogParantesSjekk();
            sjekkFakultet();
            sjekkEtterNverdi();
            sjekkForanBakXverdi();
            if(nFunnet){

                DelingsVindu.tegneVindu.tegneUttrykk[hvilkenPosisjon]=tempUttrykk;

                uttrykk[hvilkenPosisjon].setForeground(fargeKoder[hvilkenPosisjon]);
                DelingsVindu.tegneVindu.hvilketUttrykk=hvilkenPosisjon;
                DelingsVindu.tegneVindu.repaint();
                hvilkenPosisjon++;
            }
        }//end if nFunnet
    }
    else{
        nullstill();
        infoFeltTF.setText("Ingen gyldig n");
        infoFeltTF.setBackground(Color.RED);
        neste.setLabel("Start på nytt");

        return;
    }
}

```

```

        }//end else
    }//end if nFunnet
    else{
        nullstill();
        infoFeltTF.setText("Ingen gyldig n");
        infoFeltTF.setBackground(Color.RED);
        neste.setLabel("Start på nytt");
        return;
    }//end else
}//end length >0
else{
    nullstill();
    infoFeltTF.setText("Ingen gyldig n");
    infoFeltTF.setBackground(Color.RED);
    neste.setLabel("Start på nytt");
    return;
}
}
}//end else
}
}// ned sjekkUttrykk

public void actionPerformed(ActionEvent actionevent){
    //variable
    Object obj = actionevent.getSource();

    if(obj ==gjett){
        gjettGrenseFunksjonsVindu=new GjettGrenseFunksjonsVindu("Gjett
grensefunksjon");
    }//end if
    else if(neste.getLabel() == "Start på nytt"){
        nullstill();
        neste.setLabel("Tegn graf");
    }//end if
    else if(obj == neste && neste.getLabel() != "Start på nytt"){
        if(nStartBoolean){
            gammelVerdi=nStartVerdi-nVerdi;
            nStartBoolean=false;
        }//end if
        nStartBoolean=false;
        sjekkUttrykk();
        if(nFunnet){
            gammelVerdi=gammelVerdi+nVerdi;
            neste.setLabel("Øk verdien med "+nVerdi);
            funksjonsUttrykk.disable();
            String temp= new String(funksjonsUttrykk.getText());

            if (temp.equals("Feil med parenter")){
                nullstill();//nullstiller og setter tilbake infofeltet
                infoFeltTF.setText("Feil med parenter");
                infoFeltTF.setBackground(Color.RED);
            }
        }
    }
}

```

```

        neste.setLabel("Start på nytt");
    } //end if
} //end if length >0
} //end else if (obj == neste && neste.getLabel() != "Start på nytt")
else if(obj == neste&&neste.getLabel() != "Start på nytt"&&funksjonsUttrykk.getText().length()==0){
    nullstill();
    infoFeltTF.setText("Ingen gyldig n");
    infoFeltTF.setBackground(Color.RED);
    neste.setLabel("Start på nytt");
    funksjonsUttrykk.disable();
} //end else if (obj == neste&&neste.getLabel() != "Start på nytt"&&funksjonsUttrykk.getText().length()==0)

    if(hvilkenPosisjon==antallrekker){
        neste.setLabel("Start på nytt");
    } // end if hvilkenPosisjon==antallrekker

} //end actionperformed

} //end

```

8.2.5 Klassen: funksjonsfølger

```

import java.applet.Applet;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.*;
import java.net.*;

public class funksjonsfølger extends Applet implements ActionListener{

    //variable
    StringBuffer stringBuffer;
    private DelingsVindu DelingsVindu;
    private String[] funksjonsTekst;
    private String[] hjelpeTekst;
    private int antallLinjerIFil;

    public funksjonsfølger(){
    } //end Linear

    public void init(){

        //lese inn filer
        antallLinjerIFil=0;
        antallLinjer("ferdigeFunksjoner.txt");
        funksjonsTekst = new String[antallLinjerIFil];
        lesFil("ferdigeFunksjoner.txt",funksjonsTekst);
        antallLinjerIFil=0;
    }
}

```



```

String tekstFraFil = this.getParameter("fileToRead");
if (tekstFraFil != null) fileToRead = new String(tekstFraFil);
String linje;
URL url = null;
try{
    url = new URL(getCodeBase(), fileToRead);
} //end try
catch(MalformedURLException e){
    Advarsel a = new Advarsel("Feilmelding "+e.getMessage(),true);
    a.show(); //viser advarsel i hele vinduet...
} //end catch
try{
    InputStream in = url.openStream();
    BufferedReader bf = new BufferedReader(new InputStreamReader(in));
    stringBuffer = new StringBuffer();
    while((linje = bf.readLine()) != null){
        stringBuffer.append(linje + "\n");
        antallLinjerIFil++;
    } //end while
} //end try
catch(IOException e){
    System.out.println (e.getMessage());
    Advarsel a = new Advarsel("Feilmelding "+e.getMessage(),true);
    a.show(); //viser advarsel i hele vinduet...
} //end catch
} //end antallLinjer

} //end

```

8.3 Rekkeutviklingen

8.3.1 Klassen: FjernFnVindu

```
import java.awt.*;
import java.awt.event AdjustmentEvent;
import java.awt.event AdjustmentListener;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

class FjernFnVindu extends Frame implements AdjustmentListener,ActionListener{

    //variable...
    HovedVindu hovedvindu;
    Scrollbar viseAntallRekker; //scrollbaren ti lvalg av hvor mange tidligere man ønsker å fjerne
    Label viseLabel;
    int antallrekker; //det maksimale antall

    public FjernFnVindu(String s){
        super(s);//setter navn
        Toolkit toolkit = Toolkit.getDefaultToolkit();
        Dimension dimension = toolkit.getScreenSize();
        setSize(400, 200);
        setLocation(dimension.width-450,0);

        //legger til scrollbar
        antallrekker=DelingsVindu.tegneVindu.antallRekkerSomSkalVises;
        viseAntallRekker = new Scrollbar(0, antallrekker, 0, 0,DelingsVindu.utviklingPanel.antallrekker);

        //legger til panel
        Panel grunnPanel = new Panel();
        setBackground(new Color(255,255,255));
        add(grunnPanel);
        Panel panel = new Panel();
        panel.setLayout(new GridLayout(8, 2));
        setBackground(new Color(255,255,255));
        panel.add(new Label("Hvor mange av de tidligere delsummene ønsker du å se?"));
        panel.add(viseAntallRekker);

        if(antallrekker==0)ViseLabel = new Label("Bare siste delsum vil bli vist");
        else if(antallrekker>=DelingsVindu.utviklingPanel.antallrekker-1)ViseLabel = new Label("Du viser
        utviklingen til alle delsummene (som er mulig å vise)");
        else viseLabel = new Label("Det vil bli vist "+(antallrekker)+" av de foregående delsummene (hvis mulig)");
        panel.add(ViseLabel);
        viseAntallRekker.addAdjustmentListener(this);

        //buttons
        Button godkjennVerdier = new Button("OK");
        Button avbryt = new Button("Avbryt");
```

```

Color fargePaKnapp = new Color(200, 0, 100);
godkjennVerdier.setBackground(fargePaKnapp);
avbryt.setBackground(fargePaKnapp);

panel.add(godkjennVerdier);
panel.add(avbryt);

    godkjennVerdier.addActionListener(this);
avbryt.addActionListener(this);
    grunnPanel.add(panel);

    //legger til vinduslistener
HovedVindu hovedvindu = new HovedVindu(this);
addWindowListener(hovedvindu);
setVisible(true);

} //end DelingsVindu

    public void actionPerformed(ActionEvent actionevent){
//variable
Object obj = actionevent.getActionCommand();

if(obj=="OK"){
    DelingsVindu.utviklingPanel.infoFeltTF.setText("Tegner opp graf");

    DelingsVindu.tegneVindu.antallRekkerSomSkalVises=antallrekker;
    DelingsVindu.tegneVindu.repaint();
    dispose();
} //end if OK
else if(obj=="Avbryt"){
    DelingsVindu.tegneVindu.repaint();
    dispose();
} //end if Avbryt
else{
    //ingenting
} //end else
} //end actionPerformed

public void adjustmentValueChanged(AdjustmentEvent adjustmentevent)
{
    //oppdaterer
    antallrekker=viseAntallRekker.getValue();
    if(antallrekker==DelingsVindu.utviklingPanel.antallrekker-1)ViseLabel.setText("Du viser utviklingen til alle
delsummene (som er mulig å vise)");
    else if(antallrekker==0)ViseLabel.setText("Bare siste delsum vil bli vist");
    else ViseLabel.setText("Det vil bli vist "+(antallrekker)+" av de foregående delsummene (hvis mulig)");
}
} //end

```

8.3.2 Klassen: StillNVerdiVindu

```
import java.awt.*;
import java.awt.event AdjustmentEvent;
import java.awt.event AdjustmentListener;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

class StillNVerdiVindu extends Frame implements AdjustmentListener,ActionListener{

    //variable...
    HovedVindu hovedvindu;
    private Scrollbar startNVerdi;//skrollbar til start verdien til n
    private Scrollbar okNVerdi;//skrollbar til økningen i verdien til
    private Label StartNverdiLabel;
    private Label okNVerdiLabel;
    private int nStartVerdi; //startverien til n
    private int nOkVerdi; //økningen i n verdi
    private int scrollbarNVerdi; //lagre startverdien til n
    private int scrollbarOkNVerdi;//lagre økningsverdien til n

public StillNVerdiVindu(String s){
    super(s);
    Toolkit toolkit = Toolkit.getDefaultToolkit();
    Dimension dimension = toolkit.getScreenSize();
    setSize(200, 270);
    setLocation(dimension.width-450,0);
    scrollbarNVerdi=DelingsVindu.utviklingPanel.nStartVerdi;
    scrollbarOkNVerdi=DelingsVindu.utviklingPanel.nVerdi;
    //legger til scrollbar

    startNVerdi = new Scrollbar(0, scrollbarNVerdi, 0, -20,21);
    okNVerdi = new Scrollbar(0, scrollbarOkNVerdi, 0, 1,21);

    //legger til panel for å få fekst
    Panel grunnPanel = new Panel();
    setBackground(new Color(255,255,255));
    add(grunnPanel);
    Panel panel = new Panel();
    panel.setLayout(new GridLayout(10, 2));
    setBackground(new Color(255,255,255));
    panel.add(new Label("Startverdi til n"));
    panel.add(startNVerdi);
    startNVerdi.addAdjustmentListener(this);
    StartNverdiLabel = new Label("Startverden til n er: "+scrollbarNVerdi);
    panel.add(StartNverdiLabel);
    panel.add(new Label(""));
```

```

panel.add(new Label("Øk verdien til n med: "));
panel.add(okNVerdi);
okNVerdi.addAdjustmentListener(this);
okNVerdiLabel = new Label("n verdien økes med: "+scrollbarOkNVerdi);
panel.add(okNVerdiLabel);

//buttons
Button godkjennVerdier = new Button("Sett verdier");
Button avbryt = new Button("Avbryt");
Color fargePaKnapp = new Color(200, 0, 100);
godkjennVerdier.setBackground(fargePaKnapp);
avbryt.setBackground(fargePaKnapp);

panel.add(godkjennVerdier);
panel.add(avbryt);

        godkjennVerdier.addActionListener(this);
avbryt.addActionListener(this);
        grunnPanel.add(panel);

        //legger til vinduslistener
HovedVindu hovedvindu = new HovedVindu(this);
addWindowListener(hovedvindu);
setVisible(true);

} //end DelingsVindu

        public void actionPerformed(ActionEvent actionevent){
//variable
Object obj = actionevent.getActionCommand();

if(obj=="Sett verdier"){
        DelingsVindu.utviklingPanel.infoFeltTF.setText("Tegner opp graf");

        DelingsVindu.utviklingPanel.nVerdi=scrollbarOkNVerdi; //setter nverdi
        DelingsVindu.utviklingPanel.nStartVerdi=scrollbarNVerdi; //setter nverdi

        if(DelingsVindu.utviklingPanel.hvilkenPosisjon!=0)DelingsVindu.utviklingPanel.neste.setLabel("Øk
verdien med "+scrollbarOkNVerdi);
                dispose();
} //end if Sett verdier
else if(obj=="Avbryt"){
                dispose();
} //end if Avbryt
else{
                //ingenting
} //end else

```

```

} //end actionPerformed

public void adjustmentValueChanged(AdjustmentEvent adjustmentevent){

    scrollbarNVerdi=startNVerdi.getValue();
    StartNverdiLabel.setText("Startverden til n er: "+scrollbarNVerdi);
    scrollbarOkNVerdi=okNVerdi.getValue();
    okNVerdiLabel.setText("n verdien økes med: "+scrollbarOkNVerdi);
} //end adjustmentValueChanged

} //end

```

8.3.3 Klassen: TegneVindu

Hele klassen er lik i de to forskjellige bortsett fra rutinen "tegnGraf" derfor er bare rutinen "tegnGraf" lagt med her.

```

public void tegnGraf(Graphics g){
    //tegner opp grafer
    if(hvilketUttrykk>=0){
        for(int i=0;i<=hvilketUttrykk;i++){
            if(i>=(hvilketUttrykk-antallRekkerSomSkalVises)){
                tegneUttrykkTrue[i]=true;
            } //end if
            else{
                tegneUttrykkTrue[i]=false;
            } //end else
        } //end i-forløkke

        String temputtrykk= new String();
        for(int i=0;i<=hvilketUttrykk;i++){
            temputtrykk="";
            if(i!=0){
                for(int j=0;j<=i;j++){
                    if(j==0)temputtrykk=tegneUttrykk[j];
                    else temputtrykk=temputtrykk+" "+tegneUttrykk[j];
                } //end j-forløkke
            } //end i!=0
            else{
                temputtrykk=tegneUttrykk[0];
            } //end else
            System.out.println (temputtrykk.toLowerCase());

            utregning(temputtrykk.toLowerCase(),g,DelingsVindu.utviklingPanel.fargeKoder[i],tegneUttrykkTrue[i
],false);

        } //end i-forløkke
    } //end if

```

```

        if(tegnGrenseFunksjon){ //tegner opp grensefunksjon og eventuelt epsi
            if(epsilonVerdi==0.0)
                utregning(grensefunksjon.toLowerCase(),g,Color.black,true,false);
            else utregning(grensefunksjon.toLowerCase(),g,Color.black,true,true);
        }

    } //end tegnGraf

```

8.3.4 Klassen: UtviklingPanel

Det eneste som er forskjellig mellom de to programtypene er denne delen i UtviklingPanelet sin konstruktør og labelen får forskjellig tekst.

```

//legger til tekst og uttrykksfelt
    Label overskrift = new Label("Rekkeutvikling",0);
overskrift.setFont(skriftType);
utvikling.add(overskrift);

    Label overskrift2 = new Label("Skriv inn rekkeuttrykk",0);
overskrift2.setFont(skriftType);
utvikling.add(overskrift2);

} //end

```

8.3.5 Klassen: rekkeutvikling

Den eneste forskjellen mellom de to er navnet som skal stå på toppen i vinduet og navnet på knappen som starter opp programmet i et eget vindu.

```

import java.applet.Applet;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.*;
import java.net.*;

public class rekkeutvikling extends Applet implements ActionListener{

    //variable
    StringBuffer stringBuffer;
    private DelingsVindu DelingsVindu;
    private String[] funksjonsTekst;
    private String[] hjelpeTekst;
    private int antallLinjerIFil;

    public rekkeutvikling(){
    } //end Linear

    public void init(){

```

```

        //lese inn filer
        antallLinjerIFil=0;
        antallLinjer("ferdigeFunksjoner.txt");
        funksjonsTekst = new String[antallLinjerIFil];
        lesFil("ferdigeFunksjoner.txt",funksjonsTekst);
        antallLinjerIFil=0;
    antallLinjer("hjelpFil.txt");
        hjelpeTekst = new String[antallLinjerIFil];
        lesFil("hjelpFil.txt",hjelpeTekst);

    //layouten
    setLayout(new BorderLayout());
    Button button = new Button("Trykk for å starte rekkeutvikling");
    button.setBackground(new Color(155,247,191));
    button.setForeground(new Color(0, 0, 250));
    add(button, "Center");
    button.addActionListener(this);

} //end init

public void actionPerformed(ActionEvent actionevent){
    DelingsVindu = new DelingsVindu("Rekker",funksjonsTekst,hjelpeTekst);
} //end actionPerformed

public void lesFil(String fileToRead, String[] LesInnTabell){
    //leser inn informasjonen i den aktuelle filen...
    String tekstFraFil = this.getParameter("fileToRead");
    if (tekstFraFil != null) fileToRead = new String(tekstFraFil);
    String linje;
    URL url = null;
    try{
        url = new URL(getCodeBase(), fileToRead);
    } //end try
    catch(MalformedURLException e){
        Advarsel a = new Advarsel("Feilmelding "+e.getMessage(),true);
        a.show();//viser advarsel i hele vinduet...
    } //end catch
    try{
        InputStream in = url.openStream();
        BufferedReader bf = new BufferedReader(new InputStreamReader(in));
        stringBuffer = new StringBuffer();
        int i=0;
        while((linje = bf.readLine()) != null){
            LesInnTabell[i] = linje;
            i++;
        } //end while
    } //end try
    catch(IOException e){
        Advarsel a = new Advarsel("Feilmelding "+e.getMessage(),true);
    }
}

```

```

        a.show(); //viser advarsel i hele vinduet...
    } //end cathc
} //end lesFil

public void antallLinjer(String fileToRead){
    //rutine for å finne ut antall linjer i fila for å lage en automatisk størrelse på vinduet ut fra antall linjer
    String tekstFraFil = this.getParameter("fileToRead");
    if (tekstFraFil != null) fileToRead = new String(tekstFraFil);
    String linje;
    URL url = null;
    try{
        url = new URL(getCodeBase(), fileToRead);
    } //end try
    catch(MalformedURLException e){
        Advarsel a = new Advarsel("Feilmelding "+e.getMessage(),true);
        a.show(); //viser advarsel i hele vinduet...
    } //end catch
    try{
        InputStream in = url.openStream();
        BufferedReader bf = new BufferedReader(new InputStreamReader(in));
        stringBuffer = new StringBuffer();
        while((linje = bf.readLine()) != null){
            stringBuffer.append(linje + "\n");
            antallLinjerIFil++;
        } //end while
    } //end try
    catch(IOException e){
        System.out.println (e.getMessage());
        Advarsel a = new Advarsel("Feilmelding "+e.getMessage(),true);
        a.show(); //viser advarsel i hele vinduet...
    } //end cathc
} //end antallLinjer

} //end

```