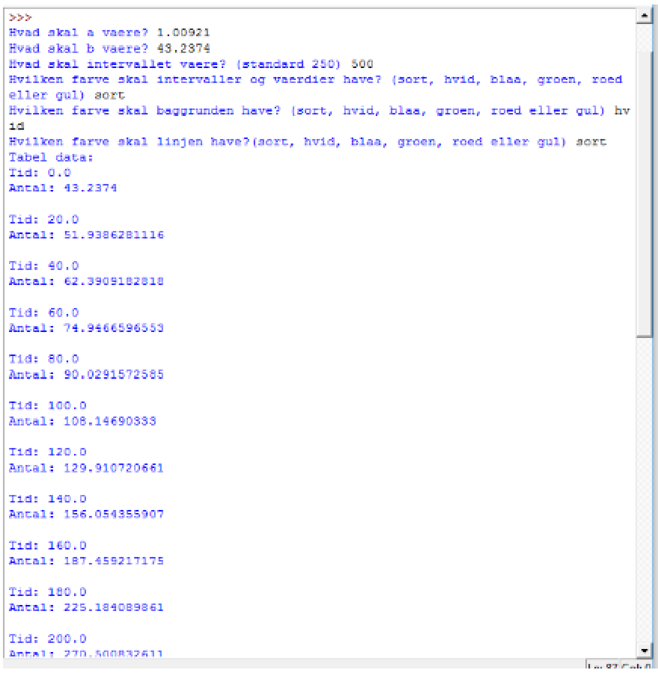
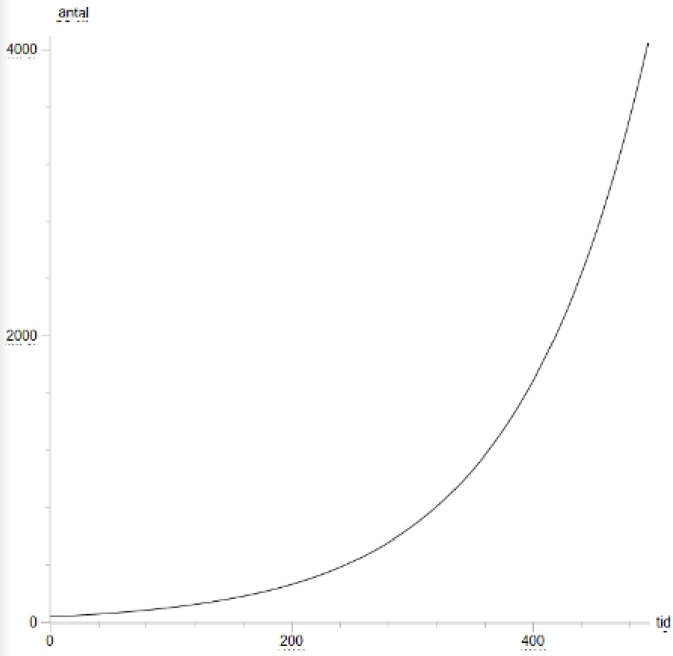
Eksponentielle modeller





Elever: Anders Skall Søby Nielsen og Magnus Langhoff Rasmusen

Tilsynsførende: Jørn Bendtsen og Karl Bjarnason

Fag: Matematik og IT (Mat/it)

Skole: HTX - Roskilde

Klasse: 2.4

Anders Skall Søby Nielsen
Magnus Langhoff Rasmusen

Indhold

Introduktion

Belbinsgrupperoller

	Magnus	Anders
Idémand	x	
Opstarter		x
Koordinator		x
Kontaktskaber	x	
Organisator		x
Specialist	x	
Formidler		
Afslutter		x
Analysator	x	

Tabel 1 Her ser vi hvem i gruppen, der laver hvad.

Konklusion: Vi har svært ved formidling.

Forventninger: Vi går efter en god start, og dermed god tid til at analysere og formidle som ikke er vores stærkeste opgaver. Som resulterer i et godt resultat.

Data fra forsøg

Der data vi har fået udleveret kommer fra en HTX-klasse, som har levet et forsøg med gærceller i opløste næringsstoffer. Klassen har valgt at fokuserer på at observere antallet af gærceller, ved hjælp af et tællekammer og mikroskop, næsten hvert 20 minut. (Af en eller anden oversat er der 40 minutter mellem 100 og 140). Der har i alt observeret gærceller i over 5,3 timer med ekstra 110 gærceller.

Teorien bag gærceller

Gærceller eller gærsvamp er eukaryote mikroorganismer som er klassificeret i riget Fungi med omkring 1.500 beskrevne arter. Gærsvamp har en størrelsesorden på mellem 5 og 10 µm og kan findes overalt i verden, meget i nektar, frugt og meget mere. Gærcellers anvendelse normalt i dagens Danmark, til at nedbryde sukker til kuldioxid, energi og alkohol. Ved disse nedbrydninger kan gærcellerne på en time omdanne en mængde glukose svarende til sin egen vægt.

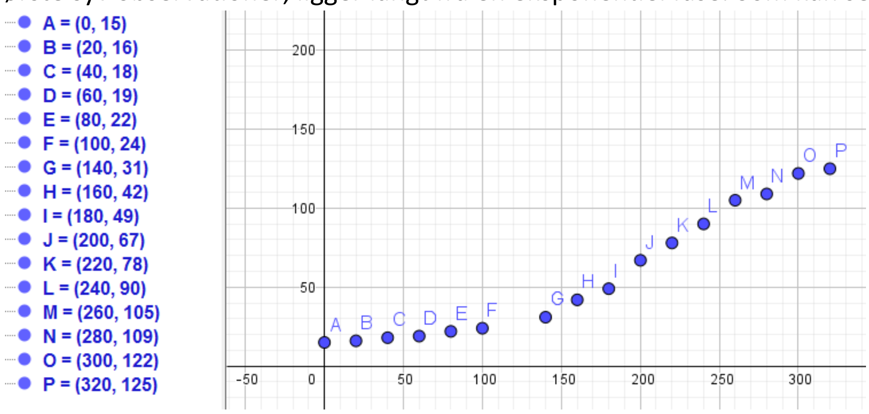
TID I MINUTTER	ANTAL GÆRCELLER
0	15
20	16
40	18
60	19
80	22
100	24
140	31
160	42
180	49
200	67
220	78
240	90
260	105
280	109
300	122
320	125

Hypotese om gærceller

Projektet har til formål at bekræfte eller afkræfte af en eksponentiel vækst i forsøget ved omkring midten. Vi for fortalt at gærcellerne vil klimatisere sig i den første periode og flade gen efter den eksponentielle vækst. Derfor skal der udvikles et program til at finde om hypotesen eller afkræfte om gærceller har en eksponentiel vækst.

Analyse

Elevernes observationer og vores GeoGebra, viser punkterne for de observationer som elleverne har lavet. Vi kan se at de første syv observationer, ligger langt fra en eksponentiel fase. Som kan ses på Figur 1.



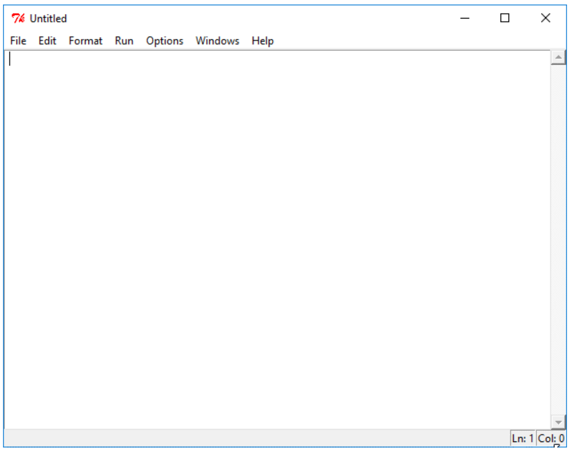
Figur 1 Hypotetisk koordinatsystem

Målgruppe

Målgruppen er ikke vild bred, vi har lavet vores program til de personer i gymnasiet eller folkeskolen, de skal lave dette forsøg med gærceller og gerne vil have nogle flere resultater at kunne relatere og visualisere med.

Valg af værktøj

VPython



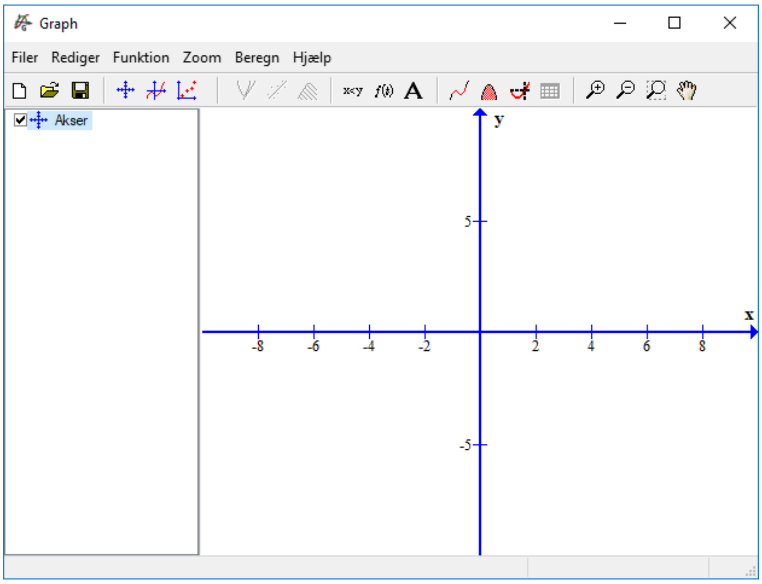
Vi startede med at prøve at lave vores produkt, i Netlogo da vi ikke kender sproget og vil ikke have for mange udfordringer. Valgte vi at Google og fandt ud af at VPython kunne lave lige præses det vi gerne ville. Efter at bruge VPythons egen dokumentation, kunne vi allerede få en graf. Vi vider eksperimenterede lidt med det og fik nogle gode resultater.

VPython var det vi manglede for at kunne lave produktet.

GeoGebra

Vi har valgt at bruge til alt de basale, da det er et let og ikke krævende program. Dog havde vi lidt problemer med programmet på en af vores computere, som resulterede i dårlige billeder.

Graph



Da GeoGebra ikke direkte kan lave et enkeltlogaritmisk koordinatsystem, bruger vi også Graph. Derudover er Graph også interageret i WordMat som er det matematisk udviklingsprogram vi bruger til Microsoft Word.

Bestemmelse af model

Indkredsning af eksponentielle faser

Vi vil i dette afsnit en eksponentielle model, som vi fik udleveret som et skema. Der vil først fremstå en konkret metode, hvorefter vi vil se at den allerede indeholder den eksponentielle fase.

Efter fuldførelsen af eksemplet og metoden er bevis, kan vi tage flere bider af forsøget of indkredse en eller flere eksponentielle faser. Hvis hypotesen ikke kan bekræftes i første omgang.

Metode

Der er flere måder at finde eksponentielle faser på, en af de metoder er at sætter datasættende i et enkeltlogaritmisk koordinatsystem. Dette vil give os et billede af, hvordan fasen kommer til at se ud. Det kan give os udfordringer når vi skal dokumentere den eller de eksponentielle faser.

Derfor kan vi anvende den eksponentielle udviklings udtryk: $f(x) = b * a^x$. Det vil sig at et vilkårligt punkt kan flyttes til $x = 0$ for at isolere b. B bruges til at udregne det næste punkts procentuelle difference. Så hvis $a_4 \approx a_5$er der en tilnærmelsesvis eksponentiel stigning.

Vi ved at $f(x) = b \; når \; x = 0 \; da \; a^0 = 1$

Det vil så sige at vi kan sætte punktet hvilket som helst sted på y-aksen som herfra kan testes eksponentialitet. Derefter kan formlen for a bruges på det punkt vi havde før på y-aksen med det næste punkt hen af x aksen osv.

Her anvender vi formlen:

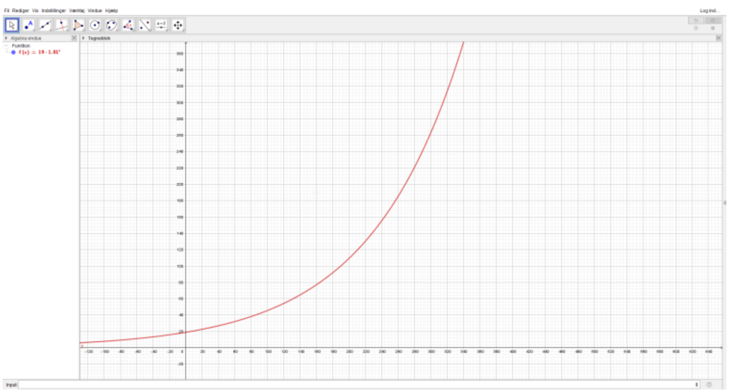
$$a = \sqrt[x_2 - x_1]{\frac{y_2}{y_1}}$$

Vi kan give a-værdien en procent afvigelse som beskriver, hvor mange vi kan ville lade den afvige fra den tidligere koefficient. For at finde afvigelsen fra a1 til a2 bruges denne formel.

$$afvigelse \; i \; procent = \frac{a_2}{a_1} * 100$$

Indkredsning af forsøgsdata

De givne datasæt har vi fået dette eksempel med gærceller. Som skrevet før bliver der taget et datasæt hvert 20 minut (Af en eller anden oversat er der 40 minutter mellem 100 og 140).



Figur 2 ligningen for grafen af alle intervallerne.

Ved denne formel finder vi vækstfaktoren.

$$a = \sqrt[x_2 - x_1]{\frac{y_2}{y_1}}$$

Her er eksempel på vækstfaktoren, af a_1d

$$a_1 = \sqrt[20-0]{\frac{16}{15}} \approx 1{,}003232$$

Fremskrivningsfaktor

NAVN	VÆRDI
a1	1,003232
a2	1,005907
a3	1,002707
a4	1,007357
a5	1,00436
a6	1,006419
a7	1,0153
a8	1,007737
a9	1,015767
a10	1,00763
a11	1,007181
a12	1,007737
a13	1,001871
a14	1,00565
a15	1,001215

Tabel 2 Fremskrivningsfaktor

Her kan vi se at der er meget forskel på, hvor langt punkterne ligger fra hinanden. Dette er relevant for hvilket intervaller vi skal anvende i modellerne.

Model - Interval (80-260)

Tid i minutter (x)	Antal gærceller (y)	Den enkle fremskrivningsfaktor
80	22	
100	24	1,0044
140	31	1,0064
160	42	1,0153
180	49	1,0077
200	67	1,0158
220	78	1,0076
240	90	1,0072
260	105	1,0077

For at vi kan teste intervallet skal vi finde funktionsforskriften for intervallet og beregne afvigelsen af den enkelte fase i procent.

Funktionsforskriften

Vi ved at det ligner meget en eksponentiel funktion hvilket betyder at vores udtryk skal så på denne formel:

$$f(x) = b * a^x$$

B = Er skæring med y-aksen

a = Er fremskrivningsfaktoren

x = Er tiden

Beregning af a

Vi regner gennemsnittet af alle fremskrivningsfaktoren ud med dette udtryk:

$$\sqrt[n]{a_1 * a_2 * a_3 * a_4 \dots}$$

Anders Skall Søby Nielsen
Magnus Langhoff Rasmusen

Vi har udregnet dette i Excel, da vi bruger Excel til grafer og allerede havde tallene inde. Resultater for den gennemsnitlige fremskrivningsfaktor gav:

$$a = 1.009016$$

Beregning af b

For at vi kan beregne b skal vi isolere den i udtrukket for en eksponentiel funktion, eftersom vi kender alle andre værdier.

Hvis vi isolerer b kommer der til at stå:

$$b = \frac{y_1}{a^{x1}}$$

Nu kan vi beregne b for hvert minut, da den første er 0 som er skæringen med y-aksen vil det sige at vi ikke behøver at tage gennemsnittet af det ved at tage summen for alle b-værdierne:

$$\frac{22}{1{,}009016^0} = 20{,}8778$$

$$b = 20{,}8778$$

Opstilling af udtrykket:

Nu har vi alle værdierne, derfra kan vi opstille vores eksponentialfunktion, som vil gøre med den formel vi startede med:

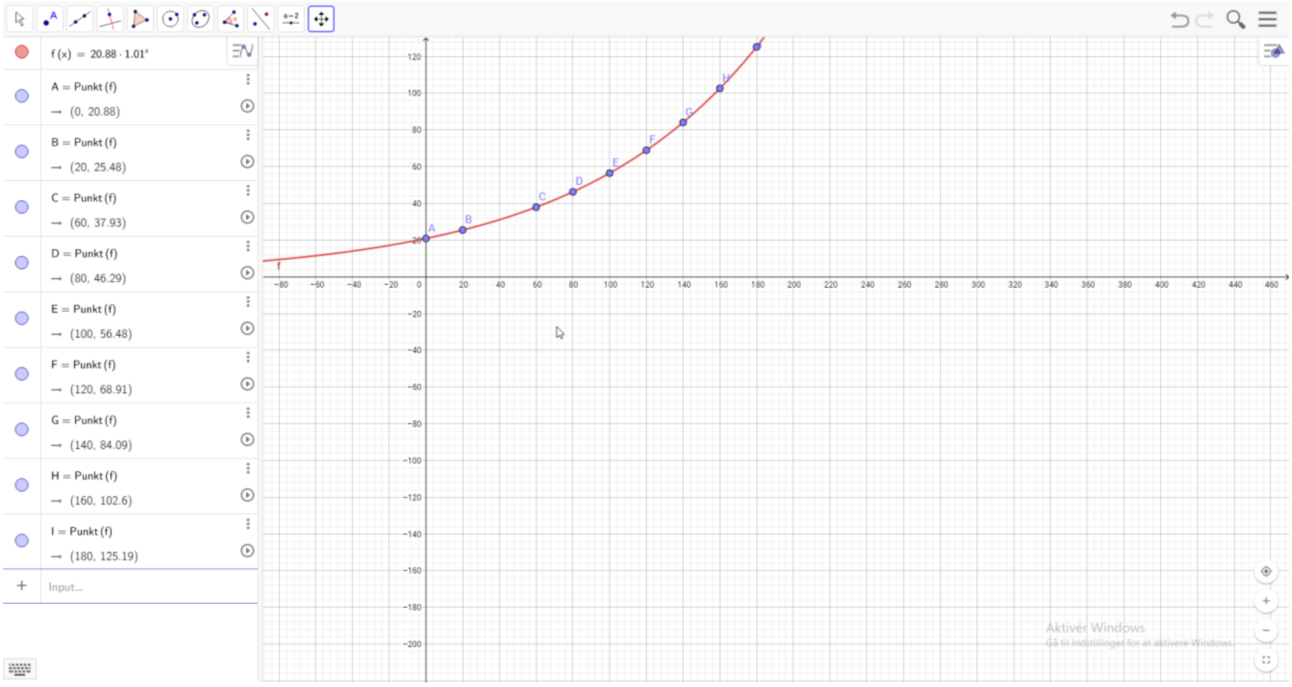
$$f(x) = b \cdot a^x$$

Hernæst kan vi udfyld datoerne.

$$f(x) = 20{,}8778 \cdot 1{,}009016^{\mathrm{x}}$$

Definitionsmængden: Funktionen er defineret i (0;180)

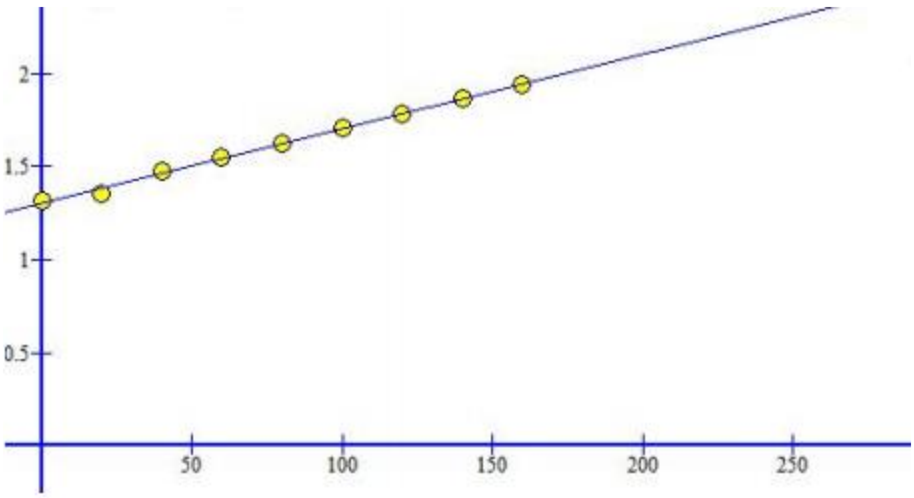
Her kan vi se funktionsudtrykket i en graf.



Figur 3 Model - Beregning og vurdering af 1. Interval (80-260)

I et enkeltlogaritmisk koordinatsystem:

Vi kan nu indtaste vores punkter i et enkeltlogaritmisk koordinatsystem, for at kunne teste den eksponentielle fase. Da vi ved at, hvis den er eksponentiel så er den lineær i et enkeltlogaritmisk koordinatsystem. Vi havde det svært ved at lave det enkeltlogaritmisk i GeoGebra, så vi skiftede over til Graph.



Figur 4 Her ser vi det enkeltlogaritmisk koordinatsystem for model. Med X-aksen som minutter og Y-aksen som Gærceller i log.

Vi kan konstatere ved at kigge linjen og punkterne at der er en afvigelse. Da vi har anvendt Graph har vi fået angivet en R^2 værdi. R^2 er determinationskoefficient, som fortæller hvor tæt punkterne ligger på linjen.

Anders Skall Søby Nielsen
Magnus Langhoff Rasmusen

Figur 5 R^2 værdien

Her kan vi se at vi får en R^2 værdi på 0,9969.

Beregning af afvigelse

Vi kan nu fra vores data finde ud af, hvor meget afvigelse der er fra hypotesen.

Det kan vi gøre ved at tage:

$$\frac{Vores\ 1.model\ data}{givne\ data} - 100\% = afvigelsen\ i\ procent$$

Vi er nød til at beregne den vores models data, det kan vi gøre ved at erstatter vores værdier i udtrykket for en eksponentiel funktion:

$$f(x) = b \cdot a^x$$

$$y = \ 20{,}8778 \cdot 1{,}009016^{\mathrm{x}}$$

Vi vil nu tage vores første værdi og sætte ind i udtrykket:

$$y = 20{,}8778 \cdot 1{,}009016^{80} = 20{,}8778$$

Da der ikke er nogen grund til at vise flere eksempler, da det er samme procedure, har vi lavet tabellen under.

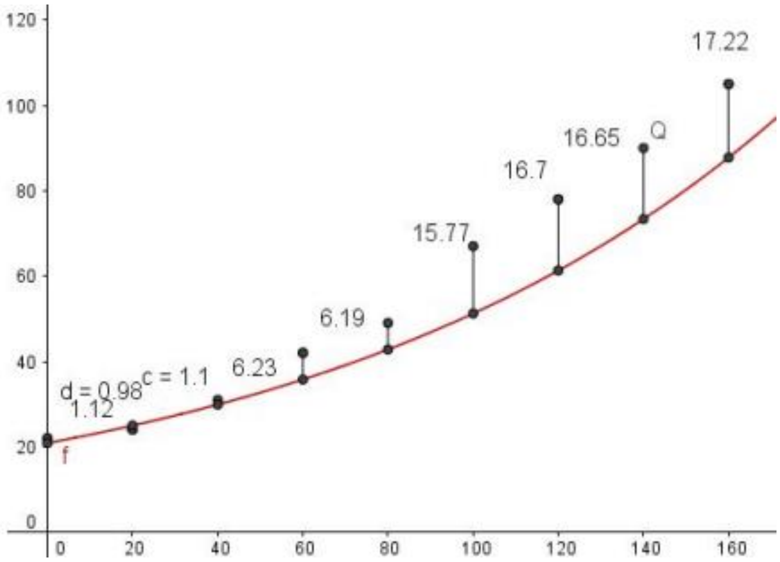
$$\frac{Vores\ 1.model\ data}{givne\ data} - 100\% = afvigelsen\ i\ procent$$

$$\frac{20{,}8770}{22} - 100\% = -5{,}75\%$$

Tid i minutter	Antal gærceller	1. Model data	Afvigelse i %
80	22	20,7340	-5,75%
100	24	24,8110	3,38%
140	31	29,6897	-4,23%
160	42	35,5276	-15,41%
180	49	42,5135	-13,24%
200	67	50,8731	-24,07%
220	78	60,8765	-21,95%
240	90	72,8468	-19,06%
260	105	87,1709	-16,98%

Regression

Regressionen kan vise, hvor langt punkterne ligger fra funktionsudtrykket. Vi kan nu lave regression på de nye intervaller vi har fået.



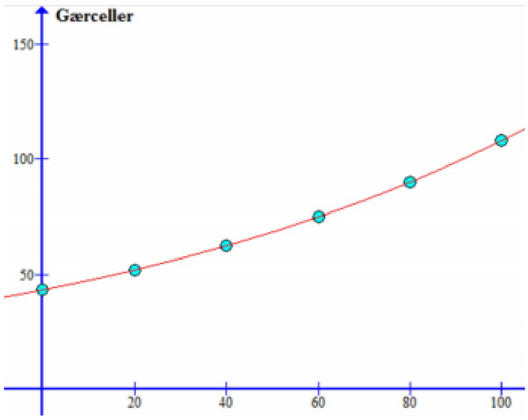
Figur 6 Vi kan se at jo større vores x værdi bliver jo længer væk er vores y værdi fra udtrykket.

Da vi ikke synes at vores resultat blev tæt nok vil vi prøve at indsnævre det yderligere.

2. Model - 2. Interval (160 - 260)

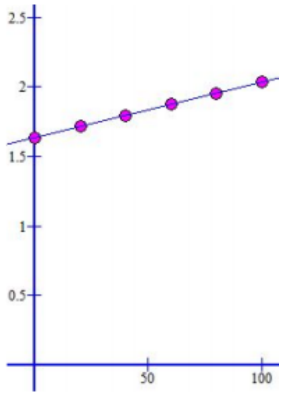
Vi har indsnævret vores interval yderligere, som sagt før, for at få en lavere afvigelse. Vi anvender sammen metode for at komme frem til den nye afvigelse. Vi er kommet frem til at vi skal anvende disse intervaller, ved at analysere de seneste intervaller og kigge på afvigelse.

- Finde b
- Udregne funktionsudtrykket, som er $F(x) = 43{,}2374 \cdot 1{,}00921^x$
- Dernæst finder vi definitionsmængden: Funktionen er defineret i (0:100)
- Der næste beregner vi afvigelsen
- Og laver regression af produktet



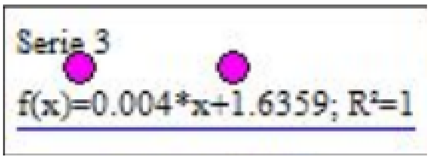
Figur 7 Nye punkter i 2. interval.

Vi plotter nu de nye punkter igen ind i et enkeltlogaritmisk koordinatsystem.



Figur 8 Her kan vi se at de ligger meget præcist på linjen, Men kender ikke R^2 endnu. Vi har igen minutter på x-aksen og gærceller i log på y-aksen.

Hvis vi får Graph til at finde R^2 kan vi se at R^2er = 1.

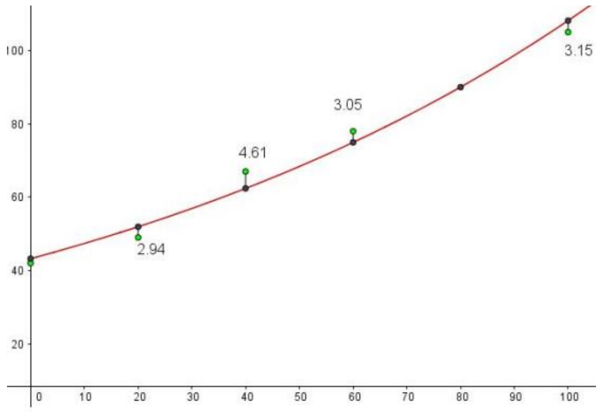


Figur 9 Her kan vi se at determinationskoefficient er lig med 1.

Nu kan vi udregne hvor tæt på den nye eksponentielle funktion lægger på vores nye interval. Dette gør vi ved at geregne afvigelsen i procent, lige som sidste gang.

Tid i minutter	Antal gærceller	1. Model data	Afvigelse i %
160	42	43,2374	-2,95%
180	49	51,9386	6,00%
200	67	62,3909	-6,88%
220	78	74,9467	-3,91%
240	90	90,0292	-0,03%
260	105	108,1469	3,00%

Vi kan allerede se at vi ligger meget tættere på, hvad vi var sidste gang. Hvis vi kigger på regressionen igen, kan vi også se at de ligger meget tæt på.



Figur 10 Regressionen af model 2.

Vi kan derved konstatere at vi ligger meget tætter på og nogle lige og af (-0,03%) fra. Derved betyder det at den er nedre eksponentiel end model 1 (den første).

Fordoblingskonstanten

For at lave fordoblingskonstanten, kan vi beregne den med formlen.

$$T_2 = \frac{\log(2)}{\log(a)}$$

Så er det bare at plotte vores a værdi ind og så har vi fordoblingskonstanten fra model 2, da den er tættes på.

$$T_2 = \frac{\log(2)}{\log(1{,}0092)} = \log(2) \cdot \frac{109{,}1949}{\log(e)} \approx 75{,}68813$$

Dette vil sige at fordoblingskonstanten er 75,68813

Delkonklusion

Vi har fundet et udstryg for den tilnærmelsesvis eksponentiel fase for vores data, som vi kunne se på model 2 ikke har en stor afvigelse, i forhold til den angivet data. Derved kan vi konkludere at de data elleveren har opsamlet har været meget nøjagtige, og da alt ikke er perfekt i verden og der er forskel på gærceller, er resultatet godt. Vi kan nu anvende vores data til at lave vores it-produkt, og hjælpe de næste elever der skal udarbejdet forsøget med gærceller, så de kan få noget mere data.

Kravspecifikation

Programmet skal kunnen levet op til de krav som vi fik udleveret. Det er dog meget frit, så vi selv lavet nogle krav vi skal følge.

- Ud fra en ligning skal programmet lave en graf.
- Mulighed for at kunne ændre på noget af det visuelle.

Originale userstories

Userstory 1 Som en gymnasieelev har vi brug for at kunne indsætte punkter og fremstille grafer så jeg kan illustrere min matematik grafisk og samtidig gøre den overskuelig.

Userstory 2 Som en gymnasieelev, der har matematik på et højere niveau, har jeg brug for et program med mulighed for at kunne udregne og indkredse eksponentielle faser, så jeg kan vise mine klassekammerater, hvordan eksponentielle findes.

Uddybelse af userstories

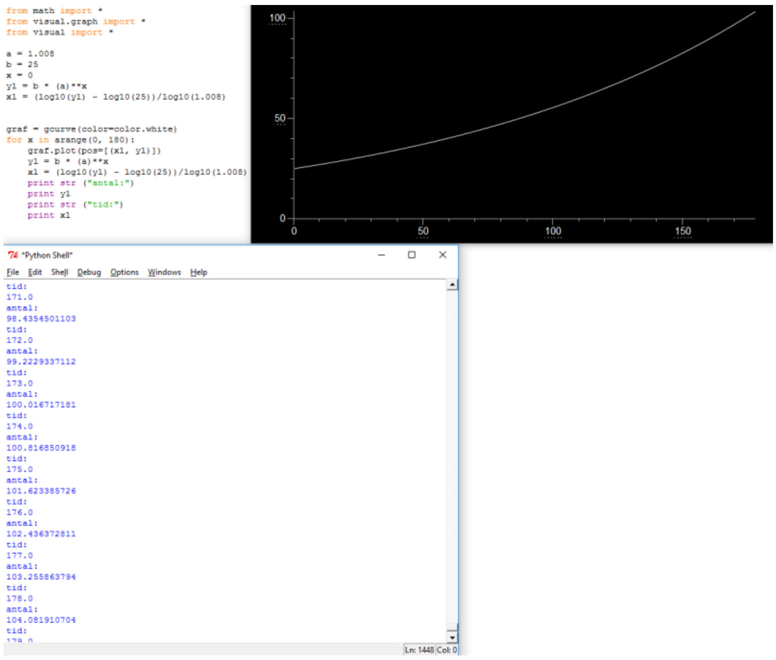
Userstory 1 At have en let tilgang til programmet uden for meget kompleksched i forhold til indsættelse af punkter samt at fremstille relevante grafer.

Overordnet design

Vores design er meget simpelt, det eneste designrelateret i programmet, er at der kan skiftes farve og at der i ud printningen har nogle mellemrum, for at gøre det overskueligt og let at se som en tabel. Det er til gengæld meget simpelt og let at bruge, da der står nogle farver at vælge mellem samt et standardinterval.

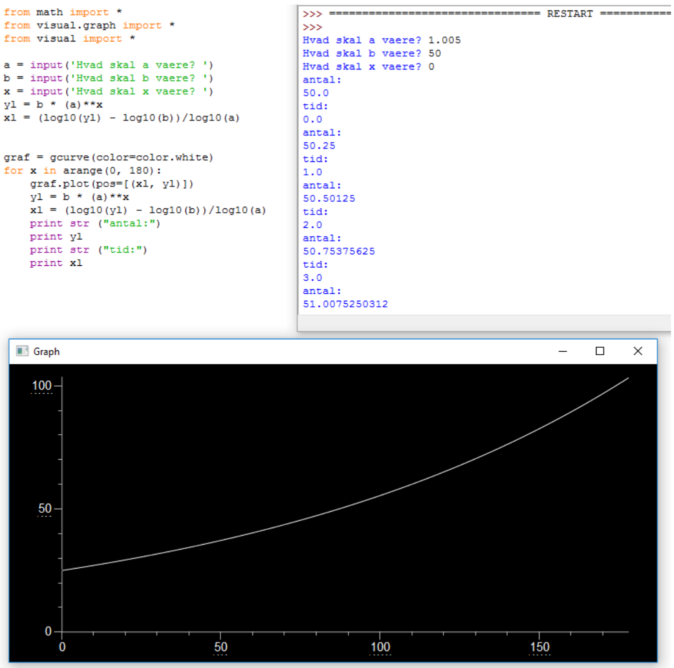
Systemudvikling

Vi startede med at lege lidt rundt med visual.graph modulet og prøve at få det til at virker. Efter lidt research på Vpython graphs dokumentation fik vi tegnet en tilfældig graf og printet nogle punkter, som kan ses på billedet herunder (Skærmbillede 1).



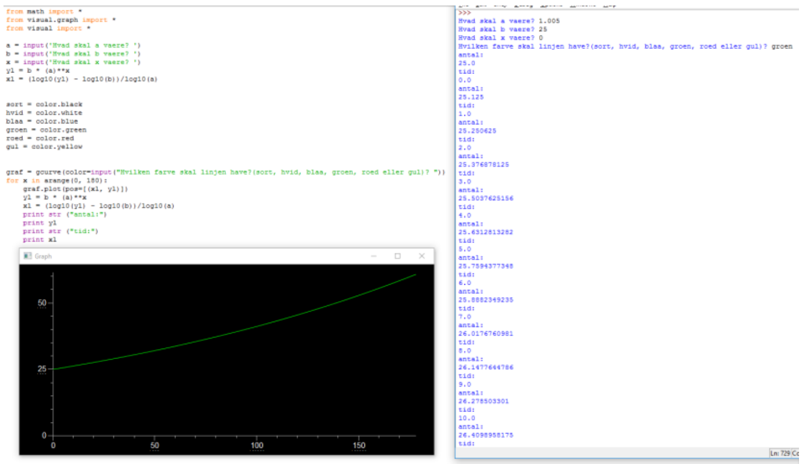
Skærmbillede 1: Første sprint af produktudviklingen.

Nu var næste mål at kunne input a, b og x værdier, først definerer vi a, b og x som input og spørger brugeren hvad den givne variable skal være.
Derefter indsætter jeg a, b og x i formlen inde i vores kode der tegner grafen. (se Skærmbillede 2)



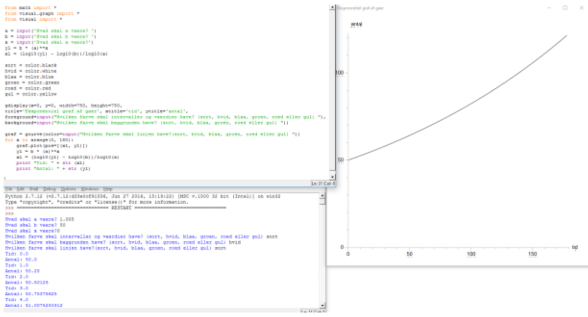
Skærmbillede 2 Step 2

Nu kan grafen sådan set tegne en eksponentiel funktion, derfor besluttede vi nu at man skulle kunne ændre farven på linjen i stedet for at den bare er angivet til at være hvid. Først prøvede vi at bede om input når grafen skulle tegnes og så skrive color.”farven” i inputtet. Da det ikke virkede kiggede vi lidt rundt på noget forskelligt dokumentation inde på Vpython, og endte med at prøve at definere farverne så ”farvenavn = color.farvenavn” og skrev så bare en af de farver vi havde defineret i inputtet, det betød så at brugeren nu kan vælge farve. (se Skærmbillede 3)



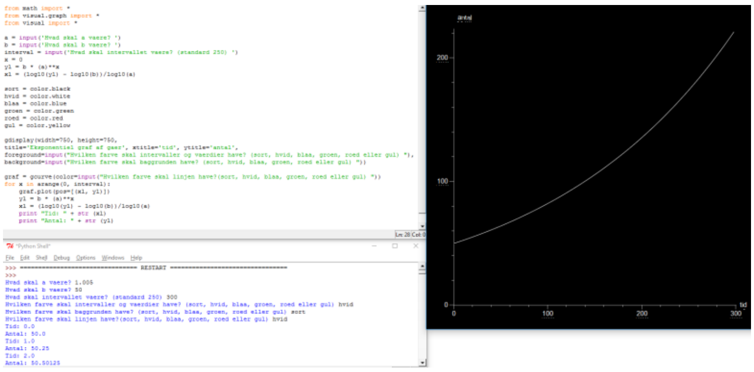
Skærmbillede 3 Step 3

Vi tænkte at når man kan bestemme farven på grafen skulle man også kun bestemme farven på de andre ting, så vi ledte efter en må at gøre dette inde og fandt frem til at man kunne definere vinduet i stedet for at det bare laver et defaultvindue. Vi definerede så vinduets størrelse samt y og x akserne til konstanter og forgrund og baggrund som et input til farve, hvor vi igen kan bruge de farver vi definerede tidligere eftersom farven defineres på samme måde som grafen, pga. begge ting er lavet med visual.graph.
(se Skærmbillede 4)



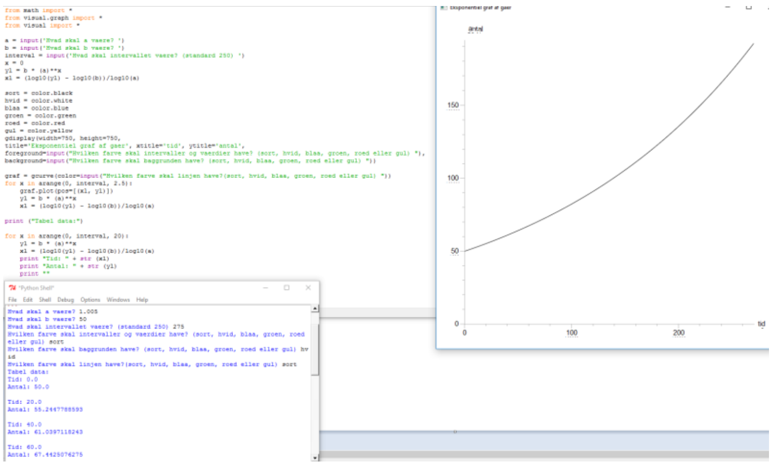
Skærmbillede 4 Step 4

Her indså vi at x altid skulle være lig med 0 i starten og skiftede det derfor ud med en måde at bestemme hvor stort et interval man vil have sin funktion til køre i ved at sætte x = 0 i koden og spørge om input til interval i starten. Vi indsætter også ”interval” frem for en konstant i vores ”for” loop der tegner graf og printer data. (se Skærmbillede 5)



Skærmbillede 5 Step 5

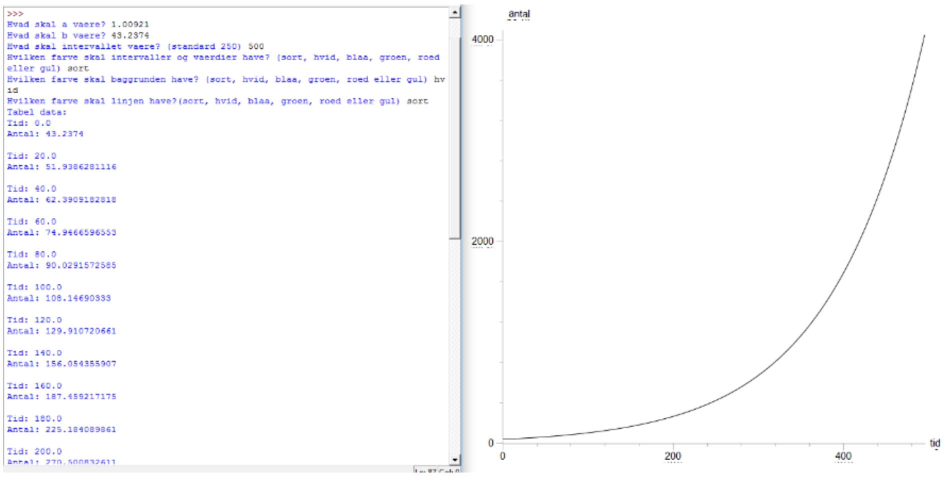
For at kunne vælge hvor ofte den udfører de forskellige commands, tilføjede vi en tredje faktor i vores loop. Dette gav en smule problemer da vi ønskede mindre print men det skulle ikke gå udover vores graf. Så vi splittede print og tegne funktionen op i to forskellige loops. Det betyder at der nu kan angives hvor ofte hver af dem skal udføres uafhængigt at hinanden.



Anders Skall Søby Nielsen
Magnus Langhoff Rasmusen

Resultatopgørelse

Vores produkt er blevet til et program, som er lavet i Vpython, som kan ses herunder.



Skærmbillede 6 Her er vores færdige resultat af produktet

Programmet kan hjælpe de studerende med at, få noget visuelt hvis de har en eksponentiel funktion. Der kan også aflæses punkter op grafen ved at holde venstre musseknap inde og fører musen hen på det punkt der ønskes at aflæse.

Konklusion

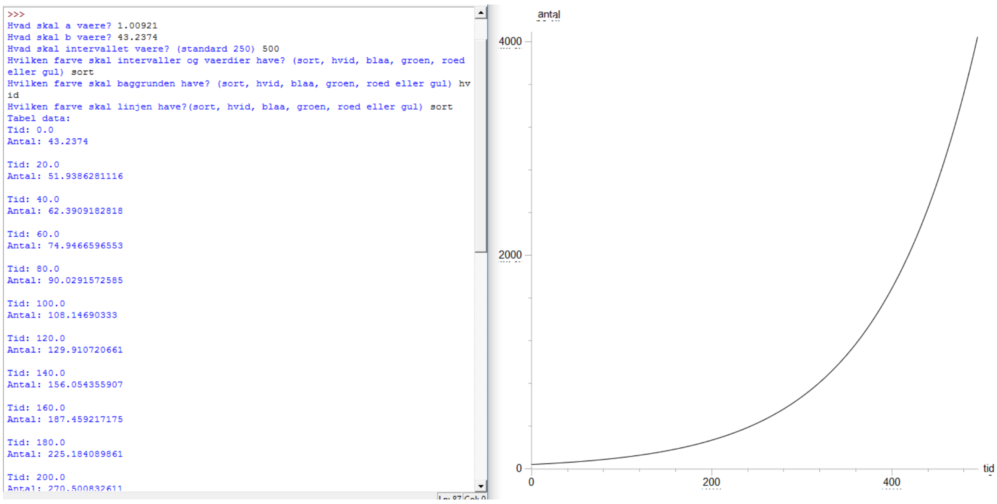
På baggrund af vores resultatopgørelse kan vi bekræfte at der er en eksponentiel som er meget kort, midtvejs i forsøget. Da programmer og omregning er lidt usikker, kom vi til nogle forskellige resultater. Dette gør at der nok er en lille usikkerhed et eller andet sted. Dog er der usikkerhed om det ellers påståede lineære faser i modellen på baggrund af programmet, som registrerede dem selv med en lav afvigelsesprocent.

Programmet udfylder de 2 kravspecifikationer grafen tegnes ved input af værdier til ligningen og der kan vælges farve på næsten alt.

Anders Skall Søby Nielsen
Magnus Langhoff Rasmusen

Bilag

VPython interface



Skærmbillede 7 Her kan man se vores interface

VPython kildekode

```
from math import *

from visual.graph import *

from visual import *

a = input('Hvad skal a vaere? ')

b = input('Hvad skal b vaere? ')

interval = input('Hvad skal intervallet vaere? (standard 250) ')

x = 0

yl = b * (a)**x

xl = (log10(yl) - log10(b))/log10(a)


sort = color.black

hvid = color.white

blaa = color.blue

groen = color.green

roed = color.red

gul = color.yellow

gdisplay(width=750, height=750,
```

```
#gdisplay(x=0, y=0, width=750, height=750,
title='Eksponentiel graf af gaer', xtitle='tid', ytitle='antal',
foreground=input("Hvilken farve skal intervaller og vaerdier have? (sort, hvid, blaa, groen, roed eller gul) "),
background=input("Hvilken farve skal baggrunden have? (sort, hvid, blaa, groen, roed eller gul) "))

graf = gcurve(color=input("Hvilken farve skal linjen have?(sort, hvid, blaa, groen, roed eller gul) "))
for x in arange(0, interval, 2.5):
    graf.plot(pos=[(xl, yl)])
    yl = b * (a)**x
    xl = (log10(yl) - log10(b))/log10(a)

print ("Tabel data:")

for x in arange(0, interval, 20):
    yl = b * (a)**x
    xl = (log10(yl) - log10(b))/log10(a)
    print "Tid: " + str (xl)
    print "Antal: " + str (yl)
    print ""
```