

Uppgift 1

Ni har fått en spridningsmatris som beskriver sannolikheten att en mussellarv född på en plats kommer att spridas till och växa upp på en annan plats. Det aktuella området innefattar Gotlands syd- och östkust samt Gotska Sandön. Er uppgift är att identifiera isolerade subpopulationer i detta området. Det finns också en fil med longitud och latitud koordinater motsvarande indexen i spridningsmatrisen.

Använd minimeringsproblemet

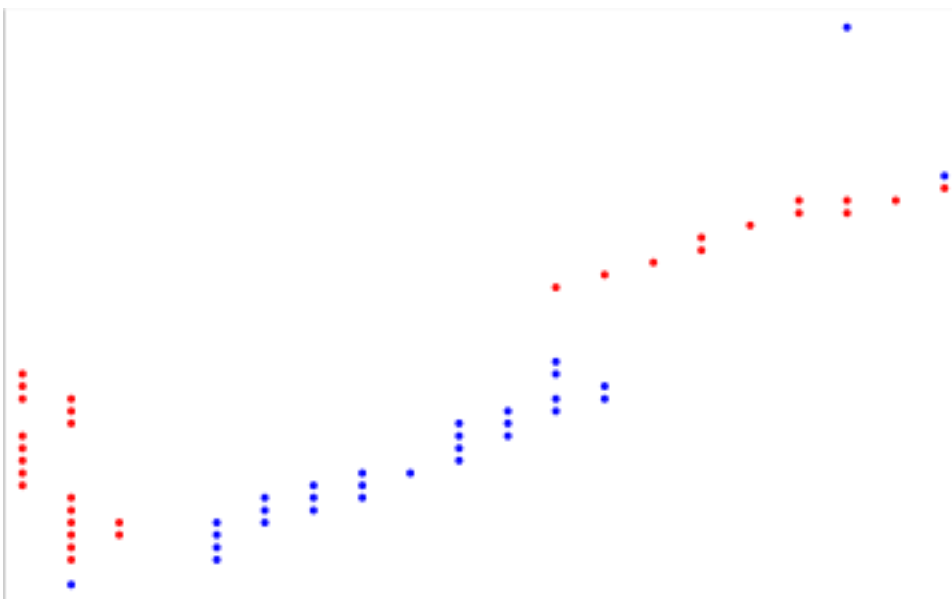
$$\min_{s_i = \pm 1} - \sum_{i,j} s_i (P_{ij} - \beta) s_j$$

för att splitta populationen i två isolerade delar (kom ihåg att ett vettigt val av parametern β är $1/(\text{antalet rader eller kolumner i } P)$). Det räcker med att göra en greedy search ett antal gånger och ta det bästa resultatet. Illustrera resultatet av din uppdelning med en matrisplot som visar att spridningen mellan subpopulationerna är liten jämfört med spridningen inom subpopulationerna. Tips: problemet har ingen klart bästa lösning.

Minimeringsproblemet löses med en Monte Carlo algoritm:

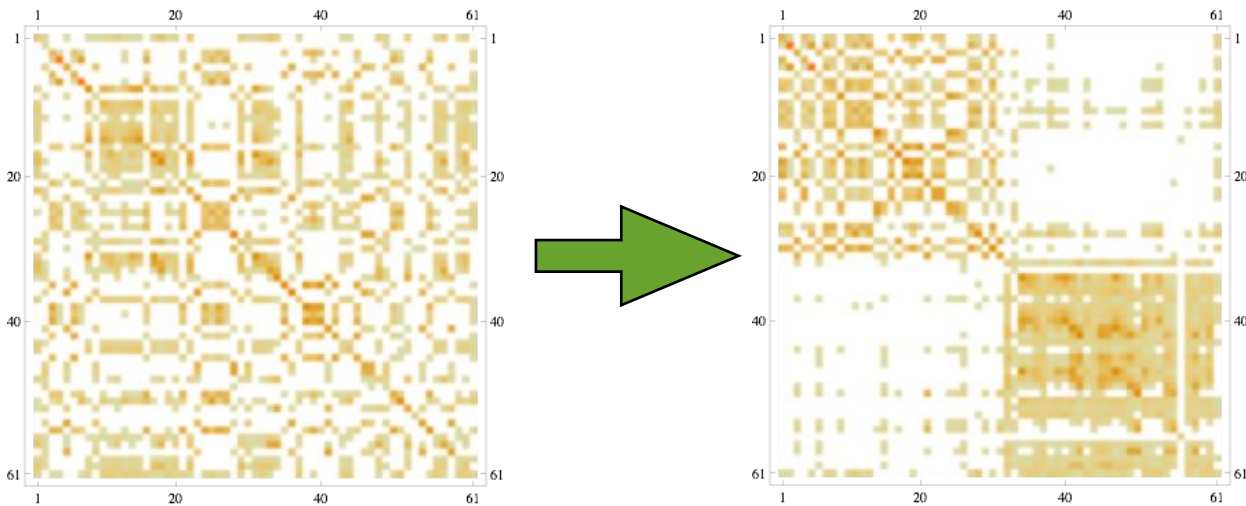
- 1) initiera elementen i vektorn s slumpmässigt med ± 1
- 2) räkna ut värdefunktionen (ekvationen given i uppgiften)
- 3) starta iteration
- 4) byt tecken på slumpmässigt elementet i s
- 5) räkna värdefunktionen för den nya s vektorn
- 6) om värdet är mindre än förut behåll s , annars byt tillbaka till det gamla s
- 7) gå till 3 (iterera typ 1000 steg, eller till ingen förbättring sker på ca 100 flips)

För $\beta=1/61$ får man, om man kör ovanstående algoritm ett antal (typ 10-100) ggr, torligen en bästa lösning som ser ut som:




Spridningsmatrisen före och efter sortering efter grupper:

Där de isolerade subpopulationerna syns tydligt eftersom matrisen är "nästan block diagonal".



Uppgift 2

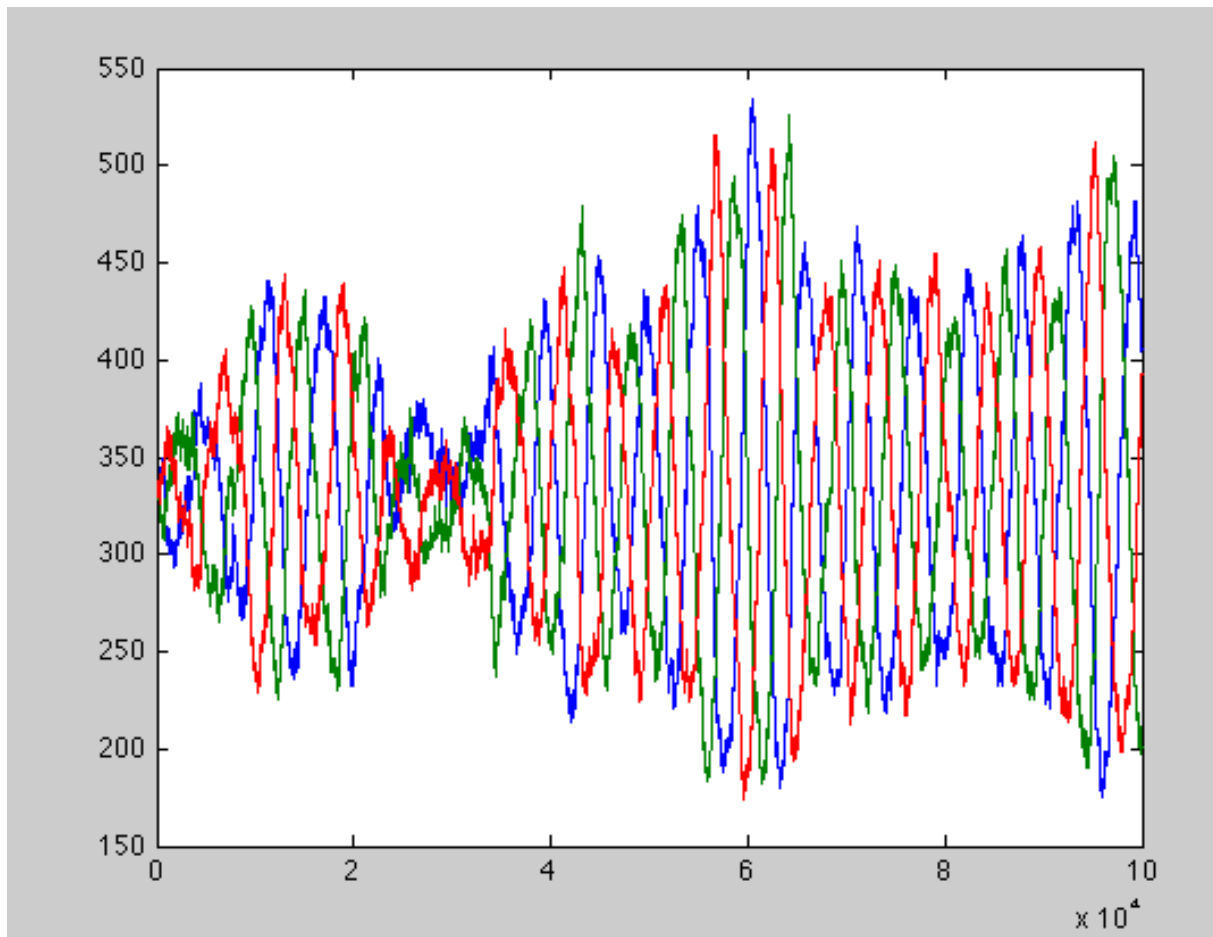
 This image cannot currently be displayed.

Förslag på lösning:

Representera fördelningen av spelare i populationen mha en vektor \mathbf{A} av längd 1000, där $A_i = 1$ om spelare i spelar sten, 2 om den spelar påse och 3 om den spelar sax. I varje tidssteg av algoritmen väljs två slumpvisa index m och n , alltså heltal mellan 1 och 1000, och operationerna som motsvarar ersättningsreglerna tillämpas.

Utan att simulera dynamiken kan man inse att $(1000,0,0)$, $(0,1000,0)$ och $(0,0,1000)$ är stabila fixpunkter. För att undersöka om det finns fler så simuleras dynamiken. Det visar sig att populationssammansättningen oscillerar kring ett (instabilt jämvikts-)tillstånd där alla strategier är lika vanliga. Amplituden på dessa svängingar varierar pga av att modellen är stokastisk (valet av agenter är slumpmässigt).

Plot av dynamiken:



Matlabkod:

```

1 - clear all
2 - N=1000;
3 - A=zeros(1,N);
4 - tmax=2e5;
5 - A(1:N/3)=1; %sten
6 - A(N/3+1:2*N/3)=2; %sax
7 - A(2*N/3+1:end)=3; %pase
8
9 - for i=1:tmax
10 - x=ceil(N*rand);
11 - y=ceil(N*rand);
12 - if A(x)==1
13 -     if A(y)==2
14 -         A(y)=1;
15 -     elseif A(y)==3
16 -         A(x)=3;
17 -     end
18 - elseif A(x)==2
19 -     if A(y)==1
20 -         A(x)=1;
21 -     elseif A(y)==3
22 -         A(y)=2;
23 -     end
24 - elseif A(x)==3
25 -     if A(y)==1
26 -         A(y)=3;
27 -     elseif A(y)==2
28 -         A(x)=2;
29 -     end
30 - end
31 - P(i,1)=length(find(A==1));
32 - P(i,2)=length(find(A==2));
33 - P(i,3)=length(find(A==3));
34 - end
35 - plot(P)
36

```

Uppgift 3

3. Antag att vi har ett miljöfarligt ämne i en porös miljö vars koncentration kommer i jämvikt mellan absorption och desorption. Genom en serie experiment har vi fått fram följande tabell av koncentrationen, k_s , av ämnet absorberat i det fasta mediet som en funktion av koncentrationen k .

k	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
k_s	0.2140	0.3777	0.4880	0.5825	0.6550	0.7091	0.7299	0.7784	0.7946	0.8230

Vi vill gärna modellera denna absorptions process. Man kan se direkt från datan att vi inte bör välja en linjär modell, så vi tvekar om vi ska använda Freundlich eller Langmuir. Motivera ditt val och ge approximativa parametrar, samt hur du kom fram till dem, för modellen.

Förslag på lösning:

Om du inte kommer ihåg Langmuir- och Freundlichmodellerna. Gå in på Comsol och skapa en "tom" fluid modell för poröst flöde som tex Solute Transport (Mobil Liquid, Immobile Solid 1).

Där kan du sedan se hur både Langmuir och Freundlich ser ut med sina par av parametrar:

Sorption

Species c:

Langmuir

$$c_{p,j} = \frac{k_{L,j} \bar{s}_j c_j}{1 + k_{L,j} c_j}, \quad k_{p,j} = \frac{\partial c_{p,j}}{\partial c_j} = \frac{k_{L,j} \bar{s}_j}{(1 + k_{L,j} c_j)}$$

Langmuir constant:

$k_{L,c}$ m

Sorption maximum:

\bar{s}_c 1

Untitled.mph (root)

- Global Definitions
- Model 1 (mod1)
 - Definitions
 - Geometry 1
 - Materials
 - Solute Transport (esst)
 - Mobile Liquid, Immobile Solid 1
 - No Flux 1
 - Initial Values 1
 - Free Flow 1
 - Mech 1

Freundlich

$$c_{p,j} = k_{F,j} c_j^N, \quad k_{p,j} = \frac{\partial c_{p,j}}{\partial c_j} = N_{F,j} k_{F,j} c_j^{N_{F,j} - 1}$$

Freundlich constant:

$k_{F,c}$

Freundlich exponent:

$N_{F,c}$ 1

Notera att Holzbecher uttrycker detta i en något annan form i sin bok, men det spelar ingen roll bara man anger vilket form man använder.

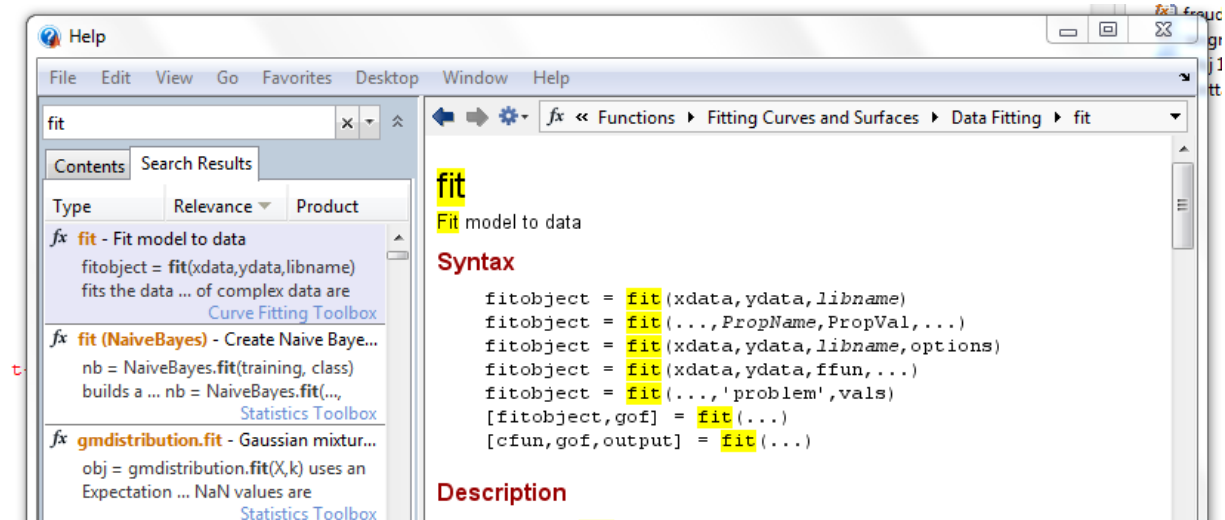
Nu kan man tex gå till matlab för att parameteranpassa och se vilken modell som passar bäst. Kap. 10 i Holzbecher handlar ju om detta.

Vi kan klippa in vektorerna från tentatesen:

```
>> k= [0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0];
```

```
>> ks=[ 0.2140 0.3777 0.4880 0.5825 0.6550 0.7091 0.7299 0.7784 0.7946 0.8230];
```

Vi kollar in help i Matlab och fastnar för fit (polyfit passar ju inte så bra här). Se figuren nedan.



```
>> f = fitype('kl*s*x/(1+kl*x)'); %för Langmuir
```

```
>> fit(k,ks,f)
```

ans =

General model:

$$\text{ans}(x) = kl*s*x/(1+kl*x)$$

Coefficients (with 95% confidence bounds):

kl = 2.406 (2.138, 2.674)

s = 1.175 (1.121, 1.229)

```
ff = fitype('kf*x^N'); % för Freudental ger
```

```
fit(k',ks',ff)
```

ans =

General model:

$$\text{ans}(x) = kf*x^N$$

Coefficients (with 95% confidence bounds):

N = 0.4898 (0.4031, 0.5765)

kf = 0.863 (0.8136, 0.9124)

Vi definerar funktioner med dessa paramtervärden

```
function y=freud(x)
kf=0.863;
N=0.4898;
y=kf.*x.^N;
```

```
function y=langmuir(x)
kl = 2.406;
s = 1.175;
y=kl.*s.*x./(1+kl.*x);
```

och jämför minstakvadratenormen för dessa båda funktioner:

```
>> norm(ks-freud(k))
```

```
ans =
```

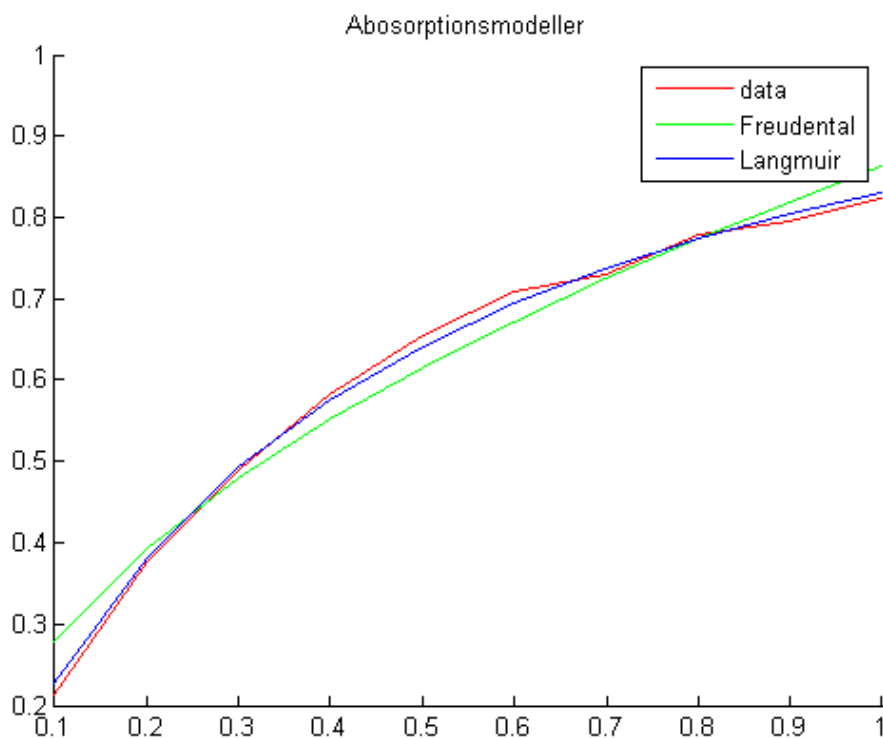
```
0.1042
```

```
>> norm(ks-langmuir(k))
```

```
ans =
```

```
0.0298
```

Vilket betyder att Langmuirmodellen passar bäst för denna uppmätta absorptionsprocessen, vilket också kan ses i följande graf:



Vi väljer alltså att modellera processen med följande Langmuirmodell:

$$k_s = \frac{2.406 * 1.175 * k}{1 + 2.406 * k}$$

Alternativt kan vi förstås använda Mathematica och funktionen FindFit:

```
In[10]=
```

```
data = {{0.1, 0.2140}, {0.2, 0.3777}, {0.3, 0.4880}, {0.4, 0.5825}, {0.5, 0.6550},  
        {0.6, 0.7091}, {0.7, 0.7299}, {0.8, 0.7784}, {0.9, 0.7946}, {1.0, 0.8230}};
```

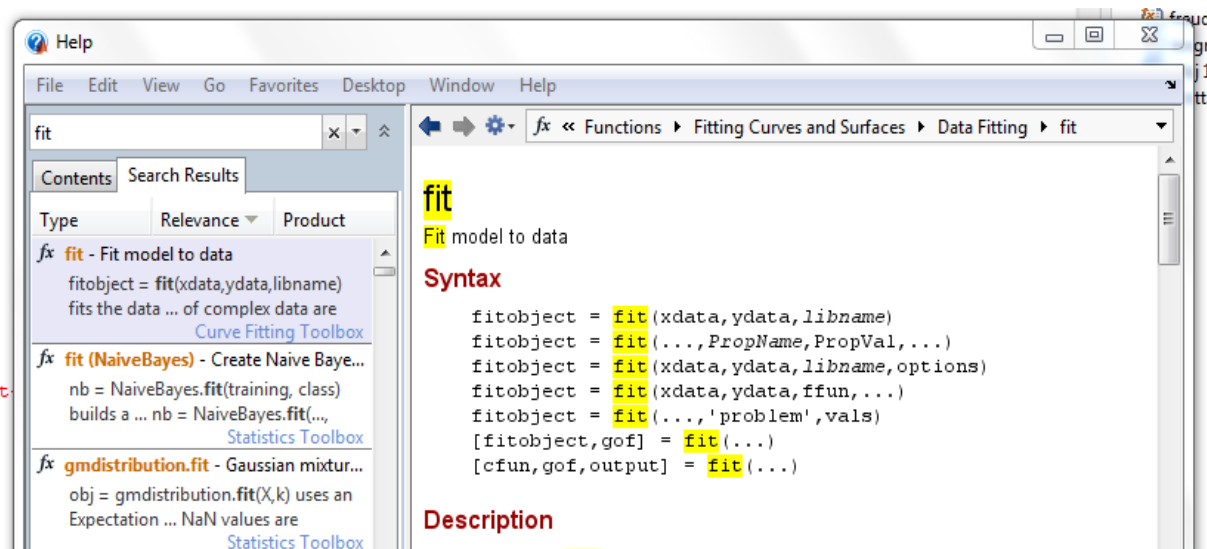
```
In[11]= FindFit[data,  $k f k^N$ , {kf, N}, k]
```

```
Out[11]= {kf → 0.86304, N → 0.489808}
```

```
In[12]= FindFit[data,  $k l s k / (1 + k l k)$ , {kl, s}, k]
```

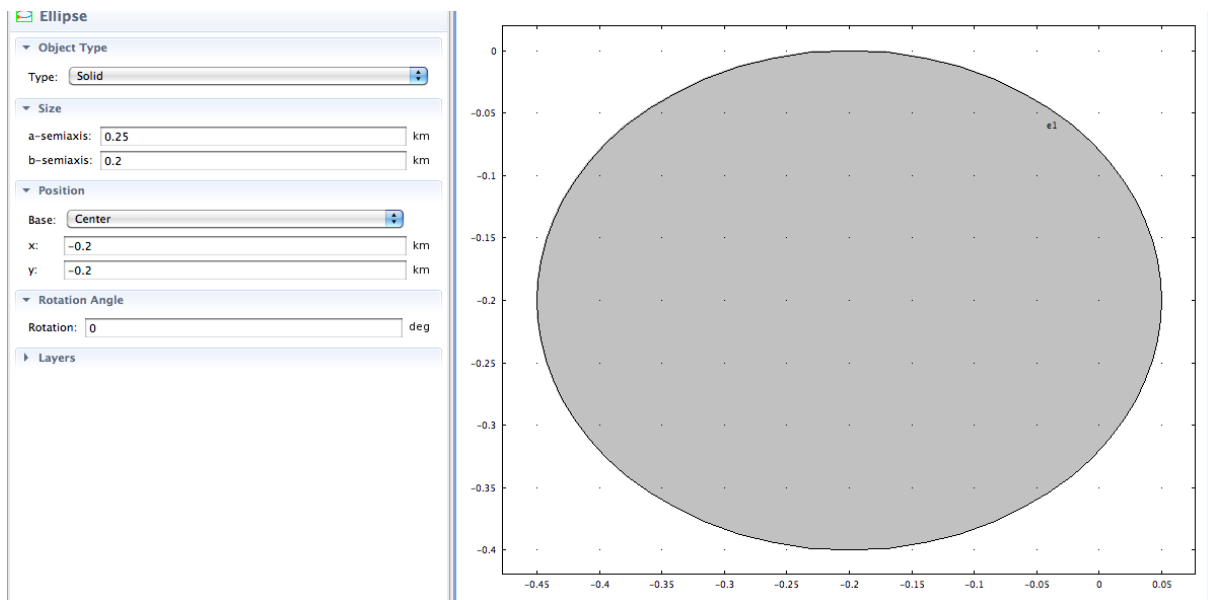
```
Out[12]= {kl → 2.40605, s → 1.17526}
```


Uppgift 4



Förslag på lösning:

Problemet löses enklast i Comsol med modulen 'Coefficient form PDE', där vi kommer behöva två variabler (eller två moduler), samt använda oss av en stationär lösning. Börja med att rita upp sjöns geometri, tex en ellips:



Ekvationen för u definieras som:

Equation

$$e_a \frac{\partial^2 u}{\partial t^2} + d_a \frac{\partial u}{\partial t} + \nabla \cdot (-c \nabla u - \alpha u + \gamma) + \beta \cdot \nabla u + \alpha u = f$$

Diffusion Coefficient

c

Absorption Coefficient

a

Source Term

f

Mass Coefficient

e_a

Damping or Mass Coefficient

d_a

Och ekvationen för v som:

Equation

$$e_a \frac{\partial^2 v}{\partial t^2} + d_a \frac{\partial v}{\partial t} + \nabla \cdot (-c \nabla v - \alpha v + \gamma) + \beta \cdot \nabla v + \alpha v = f$$

Diffusion Coefficient

c

Absorption Coefficient

a

Source Term

f

Mass Coefficient

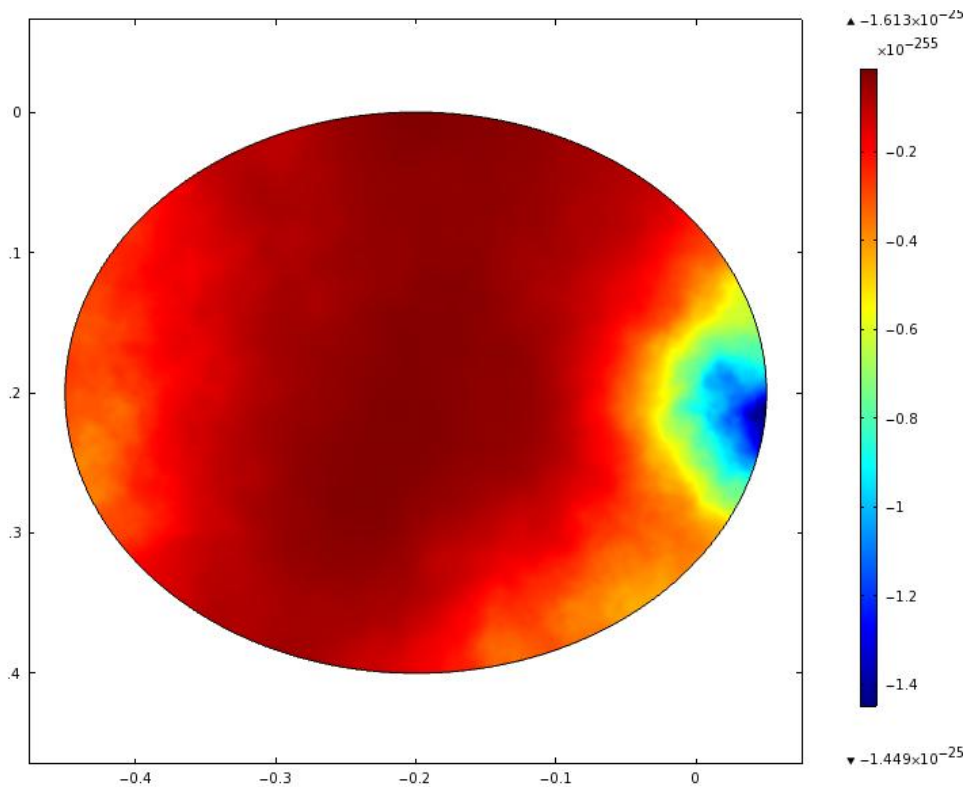
e_a

Damping or Mass Coefficient

d_a

Ange sedan no-flux villkor för v på hela randen och för u ett Dirichlet villkor med $u = c$ (som vi sedan ska variera). Initialvärdena sätts enligt uppgiften och vi använder en 'extra fine' trianguleringen.

Vi ska nu lösa systemet för olika värden på c och börjar med att sätta det litet till $c=0.1$. Även fast lösningen inte konvergerar ser vi att den algkoncentrationen är väldigt liten:



Vi testar med ett högre värde $c=1$ som då ger en stationär lösning. Med hjälp av 'Results/Derived values/Surface integration' och genom att dela med ytan för sjön (som kan beräknas genom att integrera 1 över domänen) kan vi beräkna medelalgkoncentrationen till 2.83, alltså för hög.

// Surface Integration

Data

Data set: Solution 1

Selection

Selection: Manual

1

Expression

Expression: v

Unit:

Description: Dependent variable v

Integration Settings

Genom att testa några flervärden på c kommer vi till slut fram till att c som störst kan vara ungefär 0.65. Observera att svaret kommer bero på geometrin på sjön samt dess exakta storlek.