



HÁSKÓLINN Í REYKJAVÍK

T-406-TOLU

TÖLULEG GREINING

## Skilaverkefni 3

### HÖFUNDAR

Ásdís Erla Erlingsdóttir  
[asdise21@ru.is](mailto:asdise21@ru.is)

Halldór Jökull Ólafsson  
[halldoro22@ru.is](mailto:halldoro22@ru.is)

Helgi Snær Elíasson  
[helgie22@ru.is](mailto:helgie22@ru.is)

Hlynur Ingi Árnason  
[hlynura22@ru.is](mailto:hlynura22@ru.is)

Torfi Þorgrímsson  
[torfi22@ru.is](mailto:torfi22@ru.is)

### UMSJÓNARMENN

Ingunn Gunnarsdóttir  
[ingunngunnars@ru.is](mailto:ingunngunnars@ru.is)

Olivier Moschetta  
[olivierm@ru.is](mailto:olivierm@ru.is)

## Efnisyfirlit

Hluti 1: Lotka-Volterra líkanið .....	1
Dæmi 1 .....	1
Dæmi 2 .....	3
Dæmi 3 .....	6
Dæmi 4 .....	6
Dæmi 5 .....	8
Dæmi 6 .....	9
Hluti 2: Holling-Tanner líkanið .....	11
Dæmi 1 .....	11
Dæmi 2 .....	12
Dæmi 3 .....	13
Dæmi 4 .....	15
Hluti 3: Holling-Tanner líkanið með 3 tegundum .....	16
Dæmi 1 .....	16
Dæmi 2 .....	19
Dæmi 3 .....	20
Sjálfstæð vinna .....	22
Dæmi 4 .....	22

## Töfluskrá

Tafla 1: Hlutfallsleg skekkja milli ítrana $\Delta_{y_1}$ & $\Delta_{y_2}$ fyrir mismunandi gildi á $n$ . Þar sem $T = 50$ . Aðferð Euler .....	7
Tafla 2: Hlutfallsleg skekkja milli ítrana $\Delta_{y_1}$ & $\Delta_{y_2}$ fyrir mismunandi gildi á $n$ . Þar sem $T = 10$ . Aðferð Euler .....	7
Tafla 3: Hlutfallsleg skekkja milli ítrana $\Delta_{y_1}$ & $\Delta_{y_2}$ fyrir mismunandi gildi á $n$ . Þar sem $T = 50$ . Aðferð Runge-Kutta .....	10

## Myndaskrá

Mynd 1: Vigursvið fyrir $y' = F(y)$ samkvæmt Lotka-Volterra líkaninu .....	2
Mynd 2: Graf lausnanna yfir tímabil $y_1$ er bráð og $y_2$ er rándýr .....	5
Mynd 3: Fasaferlar fyrir mismunandi upphafsgildi .....	5
Mynd 4: Fasaferla mynd með $y_0 = \begin{pmatrix} 40 \\ 9 \end{pmatrix}$ ...	6

Mynd 5: Gröf þriggja lausna yfir tímabil fyrir mismunandi upphafsgildi og mismunandi $n$ sett inn í aðferð Eulers. ....	8
Mynd 6: Hallatalan er nánast $-4$ sem sýnir að aðferð Runge-Kutta er fjórða stigs aðferð. ....	9
Mynd 7: Holling-Tanner módel .....	13
Mynd 8: Holling-Tanner módel með $w = 1$ . 14	
Mynd 9: Lotka-Volterra módel fyrir mismunandi upphafsgildi .....	15
Mynd 10: Lausnir á diffurjöfnuhneppinu fyrir $t \in [0, 400]$ .....	18
Mynd 11: Fasarit af lausnum af jöfnu 7 með mismunandi upphafsgildi .....	19
Mynd 12: Fasarit af lausnum af jöfnu 7 með mismunandi gildi á $b_1$ .....	20
Mynd 13: Lausnir á diffurjöfnuhneppinu fyrir $t \in [0, 900]$ .....	21
Mynd 14: Lausnir á diffurjöfnuhneppinu fyrir $t \in [0, 1000]$ .....	21
Mynd 15: Lausnir á diffurjöfnuhneppinu fyrir $t \in [0, 1000]$ .....	22
Mynd 16: Lausn á 8 með föstum gefnum í 9 og upphafs gildum $x_0 = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.2 \end{pmatrix}$ .....	23

## Lýsing

Í þessu verkefni ætlum við að skoða líkön sem lýsa hvernig stofnar dýrategunda þróast í samspili tveggja og þriggja tegunda. Hér verður farið yfir fyrsta hluta verkefnisins.

## Hluti 1: Lotka-Volterra líkanið

Í þessum hluta skoðum við Lotka-Volterra líkanið þar sem samspili tveggja tegunda er lýst, annar vegar rándýrs (e. predator) í breytu  $y_2$  og hins vegar bráðar (e. prey) í breytu  $y_1$ . Jöfnurnar sem lýsa kerfinu eru:

$$y_1' = \alpha y_1 - \beta y_1 y_2 \quad (1.1)$$

$$y_2' = \gamma y_1 y_2 - \delta y_2 \quad (1.2)$$

þar sem  $\alpha$ ,  $\beta$ ,  $\gamma$  og  $\delta$  eru jákvæðir fastar. Fastarnir tákna


- $\alpha$ : Vaxtarhraði báðarinnar ef enginn rándýr eru til staðar.
- $\beta$ : Hlutfall af bráð sem hverfur af völdum rándýra.
- $\gamma$ : Vaxtarhraði rándýrsins ef bráð er til staðar.
- $\delta$ : Dánartíðni rándýra

Veljum til að byrja með fastana


$$\alpha = 0.5, \quad \beta = 0.01, \quad \gamma = 0.005, \quad \delta = 0.2$$

og skoðum lausnir á jöfnuhneppinu.

## Dæmi 1

Teiknið vigursvið fyrir jöfnurnar þar sem á x-ás er breyting í  $y_1$  og á y-ás er breyting í  $y_2$ . Látið  $y_1$  og  $y_2$  vera á bilinu  $[0, 100]$ . Hér er hægt að notast við quiver skipunina í  MATLAB.

## Svar

 MATLAB kóðinn framkvæmir útreikninga og sjónræna framsetningu á vigursviði Lotka-Volterra líkansins. Kóðann fyrir mynd 1 má finna í skjali **vigursvidplot1.m** sem fylgir með í skilum. Grafið sýnir sveiflur í fjölda bráða og rándýra í einföldu vistkerfi sem má greina út frá stefnu og lit vektoranna, eftirfarandi skref voru framkvæmd:

1. Fastar fyrir líkanið skilgreindir
2. Net (meshgrid) er skapað með gildum fyrir fjölda bráða ( $y_1$ ) og rándýra ( $y_2$ ) á bilinu 0 til 100 með skrefi upp á 5.
3. Hraðabreytingar á líkaninu reiknaðar fyrir bráð og rándýr ( $\frac{dy_1}{dt}$  og  $\frac{dy_2}{dt}$ ) byggt á líkani Lotka-Volterra. Breytingarhraðinn er reiknaður út frá náttúrulegum vexti, fækkun vegna rándýra, vexti rándýra vegna bráða og náttúrulegum dauða rándýra.
4. Stærð (magnitude) vektora er reiknuð til að ákvarða hraða breytinganna með formúlunni:

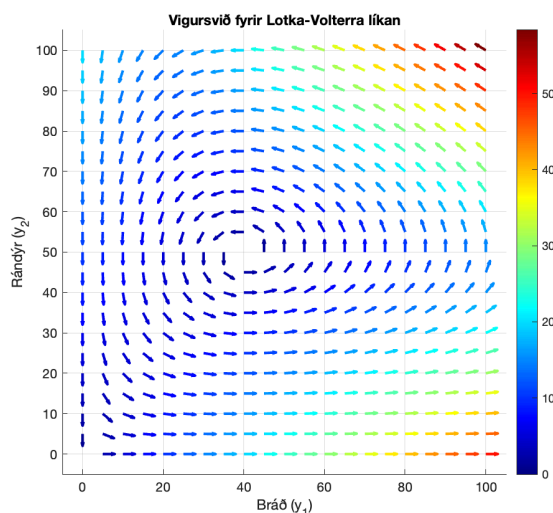
$$\|v\| = \sqrt{\left(\frac{dy_1}{dt}\right)^2 + \left(\frac{dy_2}{dt}\right)^2} \quad (2)$$

Vektorarnir eru normalíseraðir þannig að þeir sýni aðeins stefnu á grafinu með formúlunni:

$$\hat{u} = \frac{u}{\|u\|} \quad (3)$$

- Vigursviðið er teiknað með quiver, þar sem örvar tákna stefnu og hraða breytinga á fjölda bráða og rándýra. Litir örvanna tákna stærð breytinganna sem er byggt á stærðum vektoranna með colormap.
- Bætt er við litaskala (colorbar) til að sýna breytingarhraða.

Jafnvægispunkturinn í þessu líkani táknar stöðu þar sem báðir stofnar eru í jafnvægi og hann má finna með því að leysa jöfnurnar [1.1](#) og [1.2](#) jafnt og 0 sem gefa niðurstöðurnar  $y_1 = \frac{\delta}{\gamma} = 40$  og  $y_2 = \frac{\alpha}{\beta} = 50$ . Á mynd 1 kemur jafnvægispunkturinn fram sem svæði þar sem stærð vigrarnir verða nær engir og örvar mynda hringrás í kringum punktinn. Litir örvanna gefa til kynna að hraðabreytingar eru mestar þegar fjöldi bráða og rándýra eru í ójafnvægi (dökk rautt) en minnka þegar jafnvægispunktinum er náð (dökk blátt).



Mynd 1: Vigursvið fyrir  $y' = F(y)$  samkvæmt Lotka-Volterra líkaninu

Hluti af kóðanum til þess að teikna vigursviðið er eftirfarandi:

```
% Fastar
alpha = 0.5; beta = 0.01; gamma = 0.005; delta = 0.2;
axis_max = 100;

% Buum til grid fyrir y1 og y2
skref = 5; % Breytilegt skref fyrir mynd
[y1, y2] = meshgrid(0:skref:axis_max, 0:skref:axis_max);

% Finnum gildi a y1' og y2' fyrir hvert y1 og y2
dy1 = alpha.*y1 - beta.*y1.*y2;
dy2 = gamma.*y1.*y2 - delta.*y2;

% Normaliserum vigrana fyrir styttri vigra
magnitude = sqrt(dy1.^2 + dy2.^2);
dy1_norm = dy1 ./ magnitude;
dy2_norm = dy2 ./ magnitude;

% plotta vigursvið með lituðum örvum
figure;
hold on;

% Skala örvar fyrir skýrari mynd
arrow_scale = 3; % stærð örva
```

```

line_width = 2; % þykkt örva
num_colors = 256; % fjöldi lita
cmap = jet(num_colors); % Nota jet colormap (gott fyrir litaskiptingu)
colormap(cmap); % setja colormap fyrir current figure
magnitude_normalized = (magnitude - min(magnitude(:))) / (max(magnitude(:)) -
min(magnitude(:)));
color_indices = round(magnitude_normalized * (num_colors - 1)) + 1;

% Plotta örvar með tilsvareandi lit
for i = 1:size(y1, 1)
    for j = 1:size(y2, 2)
        % Teikna ör með lituðu skafti
        quiver(y1(i, j), y2(i, j), dy1_norm(i, j)*arrow_scale, dy2_norm(i,
j)*arrow_scale, 0, ...
            'Color', cmap(color_indices(i, j), :), 'LineWidth', line_width,
'MaxHeadSize', 1.5);
    end
end


```

## Dæmi 2

Skrifið forrit sem leysir diffurjöfnuhneppið með því að nota aðferð Eulers. Aðferð Eulers á að vera skilgreind í falli `eulersolver(f, y0, T, n)` þar sem  $f$  er fall sem gefur diffurjöfnurnar,  $y(0) = y_0$ , lausnin er fundin á bilinu  $[y_0, T]$  og fjöldi tímaskrefa er  $n$ .

Notið fyrst upphafsskilyrðin  $y_1(0) = 40$ ,  $y_2(0) = 9$ . Prófið ykkur svo áfram með mismunandi upphafsskilyrði og teiknið graf lausnanna sem fall af tíma fyrir a.m.k. tvö þeirra. Skilgreinið sér fall fyrir diffurjöfnuna í `lv.m`.

## Svar

Notast er við `eulersolve.m` skjalið frá  **canvas** nema bætt er við úttaksbreytum fyrir tímann og gildin á  $y$ . Það skjal inniheldur einnig `eulerstep` fallið.

`eulersolve` er aðalfallið sem leysir diffurjöfnuhneppið með því að nota aðferð Euler yfir ákveðið tímabil. Það skilar bæði tímaröðinni `tvec` og lausninni `yvec` fyrir hvert tímaskref.  $y_0$  er upphafsgildi lausnarinnar ( $y_1$  og  $y_2$ ),  $n$  er fjöldi tímaskrefa og  $T$  er lokatími tímabilsins. Tímaskrefin  $h$  eru reiknuð sem  $h = \frac{T}{n}$ , upphafsgildi lausnar er  $y = y_0$ , upphafstími er  $t = 0$ . `yvec` fylkið geymir lausnina fyrir  $y_1$  og  $y_2$  fyrir hvert tímaskref í hverjum dálk og er upphafsstillt á  $y_0$ . `tvec` geymir tímagildin fyrir hvert tímaskref, upphafsstillt á  $t = 0$ . Lykkjan yfir tímaskrefin fær nýja lausn  $y$ , reiknuð með því að bæta við breytingunni sem fæst úr fallinu `eulerstep`. Þá uppfærir `tvec` tímann  $t$  með því að bæta við  $h$  og ný lausn og nýr tími eru geymd í `yvec` og `tvec`. Niðurstaðan skilar síðan bæði `tvec` og `yvec`.

Fallið `eulerstep` er hjálparfall sem framkvæmir eitt tímaskref í aðferð Euler, reiknar breytinguna  $w$  á lausninni  $y$  fyrir hvert tímaskref. `eulerstep` fallið tekur inn  $h$  stærð tímaskrefsins,  $t$  núverandi tímagildi og  $y$  núverandi lausn. Það kallar á fallið `lv` sem reiknar breytingarhraða diffurjöfnunnar við tíma  $t$  og lausn  $y$ , margfaldar svo þessar breytingar með  $h$  til að reikna breytingu fyrir eitt skref og skilar  $w$  sem er breytingin á lausninni  $y$  yfir eitt tímaskref.

```
function [tvec, yvec] = eulersolve(y0,n,T)
    h=T/n;
    y=y0;t=0;
    yvec=[y0];tvec=[0];
    for i=1:n
        y=eulerstep(h,t,y);
        t=t+h;
        tvec(i+1)=t;
        yvec(i+1)=y;
    end
end
```

```
function w=eulerstep(h,t,y)
    w = h * lv(t,y);
end
```

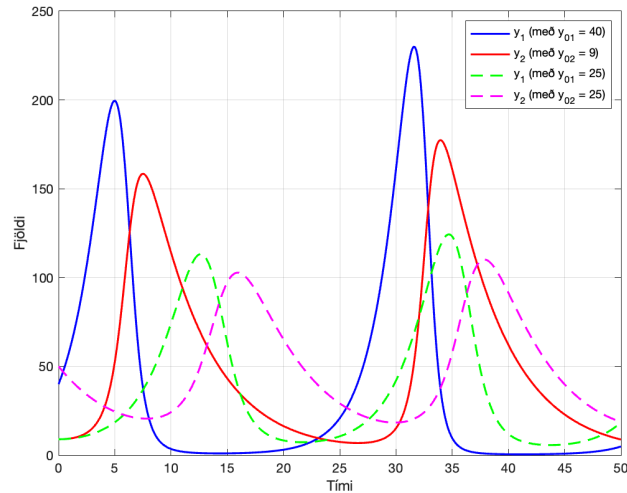
`lv.m` skráin inniheldur fallið `lv` fyrir diffurjöfnurnar og fastana. Fallið `eulerstep` kallar á fallið, `lv` tekur inn  $t$  sem er núverandi tímagildi en í þessu tilviki hefur  $t$  ekki áhrif á útreikning en það er tekið inn til að gera fallið samhæft við aðferðir sem byggja á tíma. Fallið tekur líka inn  $y$  sem er vigur með tvö gildi,  $y(1)$  er fjöldi bráða ( $y_1$ ) á tímagildinu  $t$  og  $y(2)$  er fjöldi rándýra ( $y_2$ ) á tímagildinu  $t$ .

Fallið reiknar breytingarhraða fyrir báða stofnana út frá Lotka-Volterra líkaninu. Fallið skilar vigrinum  $\left[\frac{dy_1}{dt}, \frac{dy_2}{dt}\right]$  sem inniheldur breytingarhraða bráða fyrir tímagildi  $t$  og breytingarhraða rándýra við sama tímagildi.

```
function z = lv(t,y)
    % fallið í diffurjöfnu
    alpha = 0.5; beta = 0.01;
    gamma = 0.005; delta = 0.2;

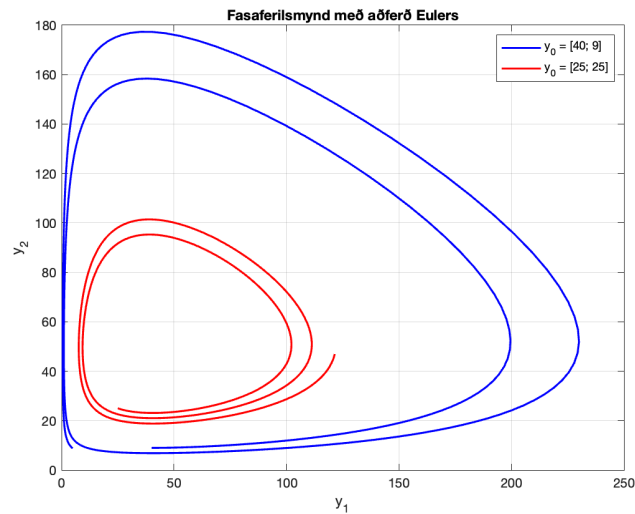
    z = [
        alpha * y(1) - beta * y(1) * y(2); % dy1/dt
        gamma * y(1) * y(2) - delta * y(2) % dy2/dt
    ];
end
```

Við prófuðum okkur áfram með nokkur mismunandi upphafsgildi ásamt  $y_0 = \begin{pmatrix} 40 \\ 9 \end{pmatrix}$ . Eftirfarandi graf (Mynd 2) sýnir þróunina á  $y_1$  og  $y_2$  yfir valið tímabil  $T = 50$ . Fyrir bæði upphafsskilyrðin sýnir  $y_1$  hærri toppa en  $y_2$  frá upphafi, og með tímanum eykst þessi munur. Þetta er í samræmi við Lotka-Volterra líkanið og bendir til þess að bráðir séu almennt fleiri en rándýr, eins og í raunverulegum vistkerfum þar sem rándýr þurfa stóran stofn bráða til að lifa af. Toppgildi  $y_1$  aukast smám saman með tímanum, sem bendir til þess að bráðastofninn nái sífellt hærri fjölda í hverri sveiflu. Toppgildi  $y_2$  hækka einnig en minna en  $y_1$ , sem sýnir að bráðastofninn hefur yfirburði. Þetta stafar af því að bráðastofninn fjölgar sér náttúrulega þegar rándýrastofninn er lítill, en rándýrastofninn er háður bráðunum fyrir vöxt og fjölgar sér hægar. Sveiflurnar í  $y_1$  og  $y_2$  tengjast fyrri þróun: þegar  $y_1$  nær hærri toppi fær  $y_2$  meira framboð af bráð og hækkar tímabundið.



Mynd 2: Graf lausnanna yfir tímabil  $y_1$  er bráð og  $y_2$  er rándýr

Til viðbótar teiknuðum við fasaferla fyrir bæði upphafsgildin. Fasaferilsmyndin (Mynd 3) sýnir sambandið milli bráða og rándýra sem fall af hvort öðru, í stað þess að sýna stofnstærðir sem fall af tíma. Báðir ferlarnir sýna að stofnstærðirnar sveiflast í lokuðum hringrásum, sem er í samræmi við Lotka-Volterra líkanið. Þessi lokaða sveifla þýðir að þegar  $y_1$  eykst, þá eykst  $y_2$  einnig tímabundið. Þegar rándýrastofninn nær hámarki, minnkar bráðastofninn vegna aukins áts, sem leiðir til minnkunar í rándýrum í kjölfarið.

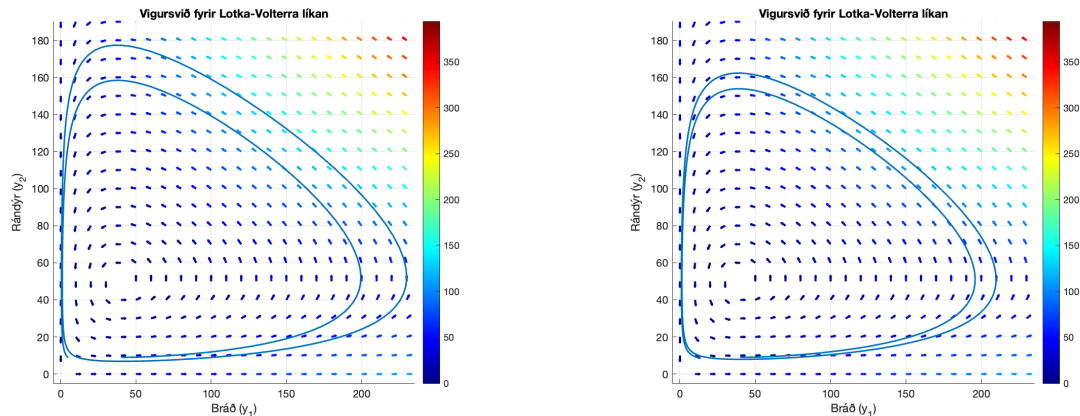


Mynd 3: Fasaferlar fyrir mismunandi upphafsgildi

### Dæmi 3

Teiknið fasaferla lausnanna (e. phase-space plot) sem þið fáið og vigursvið á eitt plott. Útskýrið hvað myndin segir okkur.

#### Svar



(a)  $n = 500$

(b)  $n = 1000$

Mynd 4: Fasaferla mynd með  $y_0 = \begin{pmatrix} 40 \\ 9 \end{pmatrix}$

Kóðann fyrir mynd 4 má finna í skjali **vigursvidplot2.m**. Eins og sést á myndum 3 og 4, þá fara ferlarnir sirka í hring. Og ferlarnir ferðast í þéttari hring þegar  $n$  stækkar. Þetta gefur til kynna að það sé einhver stöðugleikapunktur sem ferlarnir eru að ferðast í kringum, og hann sést á vigursviðs myndinni,  $y' \begin{pmatrix} 40 \\ 50 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ .

### Dæmi 4

Sýnið að aðferð Eulers er fyrsta stigs aðferð. Hér má t.d. láta  $T = 50$  og  $n = 100, 200, 400, 800$ .

Notið ode45 í MATLAB til að fá viðmiðunarlausn.

#### Svar

Notast var við aðferð Runge-Kutta með  $n = 204800$  til þess að fá viðmiðunarlausn þar sem ode45 fallið skilaði ekki nægilega nákvæmari lausn. Í MATLAB var skjalið **RKsolver.m** eftirfarandi:

```
function [tvec, yvec] = RKsolver(y0,n,T)
    y = zeros(length(y0),n);
    y(:,1)=y0;
    h=T/n;
    t=0;
    yvec=[y0]; tvec=[0];
    for i=1:n
        t=t+h;
        k1=lv(t,y(:,i));
        k2=lv(t+h/2,y(:,i)+h/2.*k1);
        k3=lv(t+h/2,y(:,i)+h/2.*k2);
        k4=lv(t+h,y(:,i)+h.*k3);
        y(:,i+1)=y(:,i)+h/6*(k1+2.*k2+2.*k3+k4);
        tvec(i+1)=t;
        yvec(:,i+1)=y(:,i+1);
    end
end
```



Að auki þá virðist aðferð Eulers vera mjög slembin þegar  $n < 1600$  og  $T$  er tiltölulega stórt. Eftirfarandi gildi á  $n$  voru valin og skekkjan metin. Kóðinn er eftirfarandi:

```
y0 = [40;9]; T = 50;
n_vector = [100, 200, 400, 800, 1600, 3200, 6400, 12800, 25600, 51200, 102400];

[~, yvec_rk] = RKsolver(y0,n_vector(end)*2,T);
error1 = zeros(2,length(n_vector));

for i = 1:length(n_vector)
    [~, yvec_euler] = eulersolve(y0,n_vector(i),T);
    error1(:,i) = abs(yvec_euler(:,end) - yvec_rk(:,end));
end

fractional_error_euler = error1(:,1:end-1)./error1(:,2:end);
```

Tafla 1: Hlutfallsleg skekkja milli ítrana  $\Delta_{y_1}$  &  $\Delta_{y_2}$  fyrir minnandi gildi á  $n$ . Þar sem  $T = 50$ .  
Aðferð Euler

gildi	1	2	3	4	5	6	7	8	9	10	11
n	100	200	400	800	1600	3200	6400	12800	25600	51200	102400
$\Delta_{y_1}$		1.0023	1.0910	1.3991	1.6832	1.8442	1.9239	1.9626	1.9815	1.9908	1.9954
$\Delta_{y_2}$		29.990	25.833	0.40281	1.1588	1.6107	1.8093	1.9052	1.9527	1.9764	1.9882

Tafla 2: Hlutfallsleg skekkja milli ítrana  $\Delta_{y_1}$  &  $\Delta_{y_2}$  fyrir minnandi gildi á  $n$ . Þar sem  $T = 10$ .  
Aðferð Euler

gildi	1	2	3	4	5	6	7	8	9	10	11
n	100	200	400	800	1600	3200	6400	12800	25600	51200	102400
$\Delta_{y_1}$		1.9550	1.9789	1.9898	1.9950	1.9975	1.9988	1.9994	1.9997	1.9998	1.9999
$\Delta_{y_2}$		2.0634	2.0313	2.0155	2.0077	2.0039	2.0019	2.0010	2.0005	2.0002	2.0001

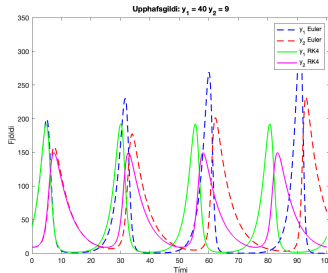
Bera má saman töflur 1 og 2 og sjá að mikill munur er á skekkju þegar  $T$  er hærra. En sjá má að ávalt er  $n$  verður hærra þá sjáum við að skekkjan verður alltaf sirka tvöfalt betri en á undan sem gefur til kynna að þetta sé fyrsta stigs aðferð þ.e.  $2^1 = 2$ .

## Dæmi 5

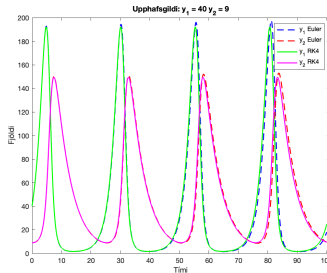
Notið aðferð Runge-Kutta (RK4) til að leysa diffurjöfnuhneppið í staðin fyrir aðferð Eulers. Berið saman við niðurstöður þar sem Euler var notað. Prófið ykkur áfram með mismunandi byrjunarskilyrði og með lausnir yfir lengri tíma. Sýnið að minnsta kosti þrjár mismunandi lausnir.

### Svar

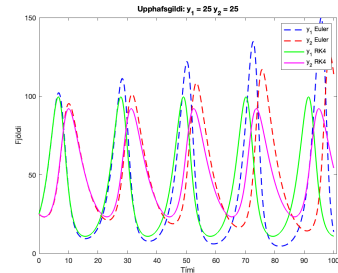
Notast var við skjölín **RKsolver.m** ásamt **lv.m** úr dæmi 4 og **eulersolve.m** úr dæmi 2 til þess að leysa eftirfarandi verkefni.



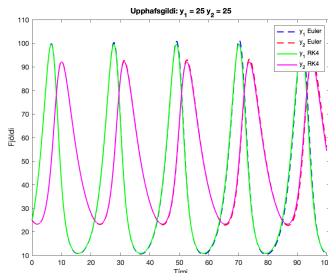
(a)  $y_0 = \begin{pmatrix} 40 \\ 9 \end{pmatrix}$  með  $n_{\text{euler}} = 1000$



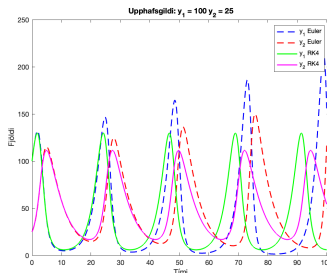
(b)  $y_0 = \begin{pmatrix} 40 \\ 9 \end{pmatrix}$  með  $n_{\text{euler}} = 16000$



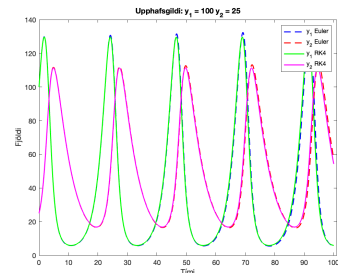
(c)  $y_0 = \begin{pmatrix} 25 \\ 25 \end{pmatrix}$  með  $n_{\text{euler}} = 1000$



(d)  $y_0 = \begin{pmatrix} 25 \\ 25 \end{pmatrix}$  með  $n_{\text{euler}} = 16000$



(e)  $y_0 = \begin{pmatrix} 100 \\ 25 \end{pmatrix}$  með  $n_{\text{euler}} = 1000$



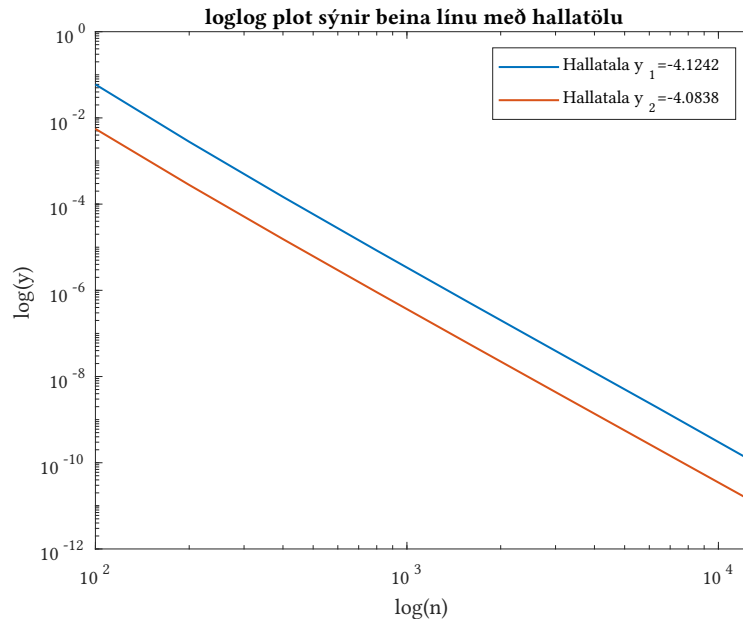
(f)  $y_0 = \begin{pmatrix} 100 \\ 25 \end{pmatrix}$  með  $n_{\text{euler}} = 16000$

Mynd 5: Gröf þriggja lausna yfir tímabil fyrir mismunandi upphafsgildi og mismunandi  $n$  sett inn í aðferð Eulers.

Vert er að taka eftir því að í aðferð Runge-Kutta var aðeins notast við  $n = 1000$  en  $n$  er breytilegt í aðferð Eulers. Það sýnir okkur hversu mun betri aðferð Runge-Kutta er fyrir minni gildi á  $n$  þ.e.a.s. Runge-Kutta er ber með sér töluvert minni skekkju en Euler aðferðin.

## Dæmi 6

Sýnið að Runge-Kutta er fjórða stigs aðferð.



Mynd 6: Hallatalan er nánast  $-4$  sem sýnir að aðferð Runge-Kutta er fjórða stigs aðferð.

Myndin sýnir hlutfall eftir því hversu mikið betri skekkjan verður við næsta  $n$  þ.e.  $2^4 = 16$  sinnum betri hverju sinni myndi skila hallatölu um það bil  $-4$  sem við fáum.

Kóðinn var framkvæmdur í MATLAB<sup>®</sup> og eins og sést erum við að skoða fyrir 8 mismunandi  $n$  og raungildi var reiknað með  $n = 1.000.000$ :

```
n = [100,200,400,800,1600,3200,6400,12800]; T=50;
y0 = [40;9];
real_n = 1e6; %1.000.000

[tvec_rk, yvec_rk_real] = RKsolver(y0,real_n,T); %Raungildi
error_rk = zeros(2,length(n));

for i = 1:length(n)
    [~, yvec_rk] = RKsolver(y0,n(i),T);
    error_rk(:,i) = abs(yvec_rk(:,end)-yvec_rk_real(:,end));
end

change_in_error = error_rk(:,1:end-1) ./ error_rk(:,2:end);

loglog(n,error_rk,LineWidth=1)
title("loglog plot sýnir beina línu með hallatölu")
p1 = polyfit(log(n), log(error_rk(1,:)),1);
p2 = polyfit(log(n), log(error_rk(2,:)),1);
slope1 = p1(1); slope2 = p2(1);
```

Út frá þessum niðurstöðum getum við séð að Runge-Kutta er klárlega fjórða stigs aðferð.

Tafla 3: Hlutfallsleg skekkja milli ítrana  $\Delta_{y_1}$  &  $\Delta_{y_2}$  fyrir minnandi gildi á  $n$ . Þar sem  $T = 50$ .  
Aðferð Runge-Kutta

<b>gildi</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>
<b>n</b>	100	200	400	800	1600	3200	6400	12800
$\Delta_{y_1}$		21.386	18.963	17.580	16.820	16.420	16.257	16.869
$\Delta_{y_2}$		19.977	18.052	17.055	16.536	16.270	16.136	16.092

Hér sést, líkt og í dæmi 4 að skekkjan minnkar um ~16 við hverja tvöföldun á 16. Önnur leið á að sýna að RK er 4.stigs aðferð. Aftur sést óreglueg skekkja þegar  $n$  er lítið en stefnir á væntanlegt gildi.

## Hluti 2: Holling-Tanner líkanið

Skoðum nú Holling-Tanner líkanið. Þetta líkan gefur samspil tveggja tegunda eins og Lotka-Volterra líkanið, en það er aðeins flóknara þar sem það tekur fleiri þætti með í reikninginn. Holling-Tanner líkanið er gefið með jöfnunum:

$$y_1' = r_1 y_1 \left(1 - \frac{y_1}{k}\right) - \frac{w y_1}{d + y_1} y_2 \quad (4.1)$$

$$y_2' = r_2 y_2 \left(1 - \frac{j y_2}{y_1}\right) \quad (4.2)$$

Notum fastana:

$$r_1 = 1, \quad r_2 = 0.1, \quad k = 7, \quad d = 1, \quad j = 1, \quad w = 0.3$$

### Dæmi 1

Finnið stöðuleikapunkta kerfisins.

#### Svar

Við framkvæmdum reikninga á blaði til þess að finna þessa stöðuleikapunkta:

$$\begin{aligned} y_2' = 0 &\Rightarrow \begin{cases} y_2 = 0 \\ y_1 = y_2 \neq 0 \end{cases} \text{ Sjáum frá jöfnu 4.2 að } y_1 \neq 0 \\ y_1' = 0 &\Rightarrow \begin{cases} y_2 = 0 &\Rightarrow r_1 y_1 \left(1 - \frac{y_1}{k}\right) = 0 \\ \hat{y} = y_1 = y_2 \Rightarrow r_1 \hat{y} \left(1 - \frac{\hat{y}}{k}\right) - \frac{w \hat{y}^2}{d + \hat{y}} = 0 \star \end{cases} \Rightarrow \begin{cases} y_1 = 0 \\ y_1 = k \end{cases} \xrightarrow{y_1 \neq 0} y_1 = k \end{aligned}$$

Gerðum þá reikninga fyrir  $y_1 = y_2$  og fáum:

$$\begin{aligned} \star \quad r_1 \hat{y} \left(1 - \frac{\hat{y}}{k}\right) &= \frac{w \hat{y}^2}{d + \hat{y}} \\ r_1 \left(1 - \frac{\hat{y}}{k}\right) &= \frac{w \hat{y}}{d + \hat{y}} \\ d + \hat{y} - \frac{\hat{y}^2}{k} - \frac{d}{k} \hat{y} &= \frac{w}{r_1} \hat{y} \\ \hat{y}^2 + \hat{y} \left(d - k + k \frac{w}{r_1}\right) - dk &= 0 \\ \hat{y} &= \frac{-dr_1 + kr_1 - kw \pm \sqrt{k^2 w^2 + r_1^2 (d^2 + 2dk + k^2) + r_1 (2dkw - 2k^2 w)}}{2r_1} \end{aligned} \quad (5)$$

$$\hat{y} \stackrel{\substack{r_1=1 \\ k=7 \\ d=1 \\ w=0.3}}{=} \frac{39 \pm \sqrt{4321}}{20} \approx \begin{cases} +5.23672 \\ -1.33672 \end{cases} \xrightarrow{(y_1, y_2 > 0)} \hat{y} \approx 5.23672$$

Þ.a.l. eru tveir stöðugleikapunktar:

$$y_{s_1} = \begin{pmatrix} k \\ 0 \end{pmatrix} = \begin{pmatrix} 7 \\ 0 \end{pmatrix}$$

$$y_{s_2} \approx \begin{pmatrix} 5.23672 \\ 5.23672 \end{pmatrix}$$

Næst skoðum við eigingildi Jacobi-fylkisins í þessum punktum til að skoða stöðugleika þeirra:

$$Dy = \begin{pmatrix} \nabla y_1 \\ \nabla y_2 \end{pmatrix} = \begin{pmatrix} -dy_2 \frac{w}{(d+y_1)^2} + r_1 - 2r_1 \frac{y_1}{k} & -y_1 \frac{w}{d+y_1} \\ jr_2 \frac{y_2^2}{y_1^2} & r_2 \frac{-2jy_2+y_1}{y_1} \end{pmatrix} \quad (6)$$

$$Dy \Big|_{y_{s_1}} = \begin{pmatrix} -1 & -\frac{21}{80} \\ 0 & \frac{1}{10} \end{pmatrix} \xRightarrow{\text{Eigen values}} \lambda = \begin{cases} -1 \\ \frac{1}{10} = 0.1 \end{cases}$$

$y_{s_1}$  er þar af leiðandi ekki aðfellustöðugur, því eitt af eigingildum Jacobi-fylkisins í þeim punkt hefur ekki neikvæðan raunhluta. Einnig sést að punkturinn er ekki sveiflukendur, því eigingildin hafa engann þverhluta.

Og eins með  $y_{s_2}$

$$Dy \Big|_{y_{s_2}} = \begin{pmatrix} \frac{3221-61\sqrt{4321}}{1470} & \frac{-101+\sqrt{4321}}{140} \\ \frac{1}{10} & -\frac{1}{10} \end{pmatrix}$$

$$\xRightarrow{\text{Eigen values}} \lambda = \frac{1537}{1470} + \frac{-61 * \sqrt{4321} \pm \sqrt{26798291 - 404722\sqrt{4321}}}{2940} \approx \begin{cases} -0.468176 \\ -0.168418 \end{cases}$$

Bæði eigingildin hafa neikvæðan raunhluta, þ.a.l. er  $y_{s_2}$  aðfellustöðugur stöðugleika punktur.

## Dæmi 2

Finnið lausnir á kerfinu með RK4 og lýsið hvað er að gerast.

### Svar

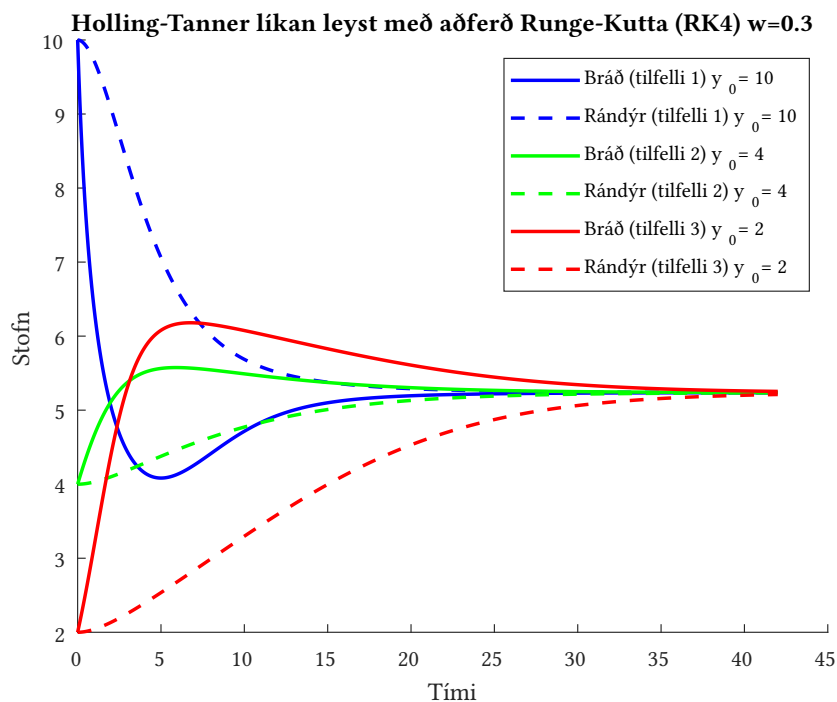
Hér er notast við **RKsolver2.m** skrána sem líkist **RKsolver.m** að miklu leyti nema í stað  $\mathbf{lv}$  fallsins þá notum við fallið **ht** (Holling-Tanner) sem finna má í skránni **ht.m**:

```
function [tvec, yvec] = RKsolver2(y0,n,T,w)
arguments
    y0; n; T; w = 0.3; % default gildi ef ekki er sett inn w
end
y = zeros(length(y0),n);
y(:,1)=y0;
h=T/n;
t=0;
yvec=[y0];tvec=[0];
for i=1:n
    t=t+h;
    k1=ht(t,y(:,i),w);
    k2=ht(t+h/2,y(:,i)+h/2.*k1,w);
    k3=ht(t+h/2,y(:,i)+h/2.*k2,w);
    k4=ht(t+h,y(:,i)+h.*k3,w);
    y(:,i+1)=y(:,i)+h/6*(k1+2.*k2+2.*k3+k4);
    tvec(i+1)=t;
    yvec(:,i+1)=y(:,i+1);
end
end
```

Holling-Tanner líkanið í sér skjali. Þar sem  $w$  er inntaksbreyta, fyrir næsta dæmi:

```
function z = ht(t,y,w)
    arguments
        t; y; w = 0.3; % default gildi ef ekki er sett inn w
    end
    r_1 = 1; r_2 = 0.1; k = 7;
    d = 1; j = 1;
    z = [
        r_1*y(1)*(1 - (y(1)/k)) - (w*y(1))/(d + y(1))*y(2);
        r_2*y(2)*(1 - (j*y(2))/y(1))
    ];
end
```

Hér eru 3 mismunandi upphafsgildi þegar  $w = 0.3$ . Sést að grafið leitar að aðfellustöðuga stöðugleikapunktinn  $y_{s_2} = \begin{pmatrix} 5.23672 \\ 5.23672 \end{pmatrix}$  fyrir öll upphafsgildi.



Mynd 7: Holling-Tanner módel

### Dæmi 3

Látum nú  $w = 1$  og leystu jöfnurnar fyrir 3 mismunandi upphafsskilyrði. Plottaðu lausnirnar og útskýrðu hvað er að gerast. Skoðaðu bæði tímaháðar lausnir og fasaferla.

#### Svar

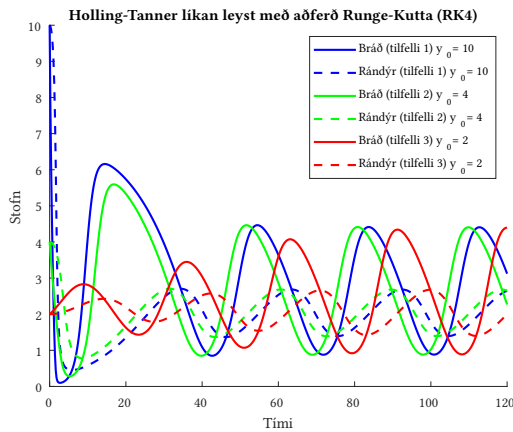
Hér þurfum við bara að beita störun á jöfnu 4 og nýta jöfnu 5 til þess að finna nýja stöðugleikapunkta. Við sjáum að  $y_{s_1} = \begin{pmatrix} 7 \\ 0 \end{pmatrix}$  er ennþá stöðugleikapunktur og við vitum að hann er ekki aðfellustöðugur. Síðan notum við jöfnu 6 til þess að smíða nýtt Jacobi-fylki út frá nýja stöðugleikapunktinum og athugum síðan eigingildi þess.

$$\hat{y} = \frac{-1 \pm \sqrt{29}}{2} \quad (y_1, y_2 > 0) \quad \underline{\underline{=}} \quad \frac{-1 + \sqrt{29}}{2} \approx 2.19258$$

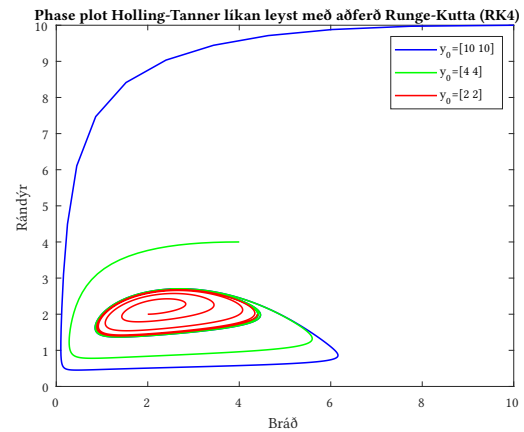
$$Dy|_{y_{s_2}} = \begin{pmatrix} \frac{67-11\sqrt{29}}{49} & \frac{-15+\sqrt{29}}{14} \\ \frac{1}{10} & -\frac{1}{10} \end{pmatrix}$$

$$\begin{aligned} \text{Eigen values} \\ \Rightarrow \lambda &= \frac{-110\sqrt{29} + 621 \pm \sqrt{3}\sqrt{254987 - 50440\sqrt{29}}}{980} \\ &\approx 0.02922 \pm 0.22799i \end{aligned}$$

Nú fyrst að eigingildi Jacobi-fylkisins er hefur jákvæðan raunhluta eru hvorugir stöðugleikapunktanna aðfellustöðugir, en einn af þeim er núna orðinn sveiflukenndur vegna þverhluta, eins og sést á mynd 8.



(a) Holling-Tanner módel línurit



(b) Holling-Tanner módel fasarit

Mynd 8: Holling-Tanner módel með  $w = 1$

Hluti af kóðanum er eftirfarandi (án plot-skipana):

```
% Veljum skrefafjölda og lokatíma
n = 600; T=120;

% Upphafsgildi
y0_1 = [10; 10]; y0_2 = [4; 4]; y0_3 = [2; 2];

% Breytum w = 0.3;
w = 1;

% Faum lausnir á diffurjöfnunum með RK4
[t,y1] = RKsolver2(y0_1,n,T,w);
[t,y2] = RKsolver2(y0_2,n,T,w);
[t,y3] = RKsolver2(y0_3,n,T,w);
```

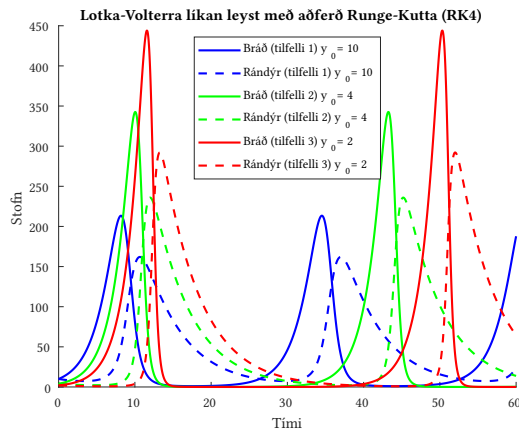


## Dæmi 4

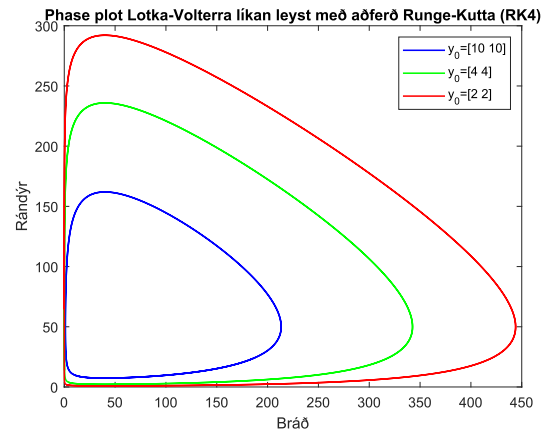
Berðu saman lausnir úr Holling-Tanner líkaninu og lausnir úr Lotka-Volterra líkaninu. Er eitthvað sem er líkt með lausnunum? Er eitthvað sem er öðruvísi?

### Svar

Ef borið er saman myndir 8b og 9b þá sést að bæði ferðast í hringrás en það er mikill munur á hvernig líkönin bregðast við mismunandi upphafs skilyrðum. Fyrir Holling-Tanner er ljóst að aðeins ein hringrás er „stöðug“, en í Lotka-Volterra eru margar hringrásir mögulegar. Fyrir mismunandi upphafsgildi breytist útslag og lota Lotka-Volterra, á meðan útslag og lota Holling-Tanner nálgast alltaf sama gildi eftir ákveðinn tíma „time to steady state“




(a) Lotka-Volterra módel línurit.



(b) Lotka-Volterra módel fasarit.

Mynd 9: Lotka-Volterra módel fyrir mismunandi upphafsgildi

Hér er  MATLAB kóðinn sem tekur 3 mismunandi upphafsgildi og notar aðferð Runge-Kutta með til að leysa fyrir Lotka-Volterra móleðið, þar sem  $n = 3200$  og  $T = 60$  til að sýna 2 lotur:

```
y0_1=[10;10]; y0_2=[4;4]; y0_3=[2;2];  
n=3200;T=60;
```

```
[t, rk_1]=RKsolver(y0_1,n,T);  
[time2, rk_2]=RKsolver(y0_2,n,T);  
[time3, rk_3]=RKsolver(y0_3,n,T);
```

Gerðir voru reikningar fyrir Holling-Tanner með þessum upphafsgildum í dæmi 3 sem má sjá á myndum 8a og 8b.

### Hluti 3: Holling-Tanner líkanið með 3 tegundum

Við skoðum nú Holling-Tanner líkanið úr hluta 2 en ætlum að bæta við einni tegund og skoða samspil milli þriggja tegunda. Látum  $X$  vera plöntur,  $Y$  vera plöntuætur og  $Z$  vera kjötætur, þá höfum við samspilið að  $Z$  borðar  $Y$  og  $Y$  borðar  $X$ . Þá má stilla upp jöfnunum

$$X' = X(1 - X) - \frac{a_1 X}{1 + b_1 X} Y \quad (7.1)$$

$$Y' = \frac{a_1 X}{1 + b_1 X} Y - d_1 Y - \frac{a_2 Y}{1 + b_2 Y} Z \quad (7.2)$$

$$Z' = \frac{a_2 Y}{1 + b_2 Y} Z - d_2 Z \quad (7.3)$$

Fastar líkansins segja til um misunandi eiginleika í samspili tegundanna. Til dæmis er  $d_1$  náttúrulegur dauði plönuætna og  $a_2$  er hámarkshlutfall af plöntuætum sem kjötætur geta neytt. Við ætlum að skoða fastann  $b_1$  sérstaklega, sem stjórnar hversu mikið plöntuæturnar geta neytt af plöntum.


Við skulum skoða lausnir þar sem fastarnir eru

$$a_1 = 5, \quad b_1 = 3, \quad a_2 = 0.1, \quad b_2 = 2, \quad d_1 = 0.4, \quad d_2 = 0.01$$

#### Dæmi 1

Finnið lausnir á diffurjöfnuhneppinu fyrir  $t \in [0, 400]$  og plottið graf þeirra sem fall af tíma með byrjunarskilyrðum  $X(0) = 0.8, Y(0) = 0.1$  og  $Z(0) = 8$ .

#### Svar

Notast er við **RKsolver3.m** skjal sem líkist þeim **RKsolver.m** og **RKsolver2.m** sem eru notaðar í hlutum 1 og 2 en nú smíðum við nýtt fall fyrir Holling-Tanner líkanið í  MATLAB:

xxx glósur um hvað kóðinn gerir xxx

$y_0$ : Upphafsgildi tegundanna sem vigurinn  $[y_1; y_2; y_3]$  þar sem raðast plöntur, plöntuætur og kjötætur. Stofnstærðirnar eru upphafsstilltar með  $y_0$  og tímaskrefið  $h$  er reiknað sem  $h = \frac{T}{n}$ . Í hverju skrefi eru fjórir stuðlar  $(k_1, k_2, k_3, k_4)$  reiknaðir sem meta breytingar á stofnum.  $k_1$ : Breytingarhraði við upphaf skrefsins.  $k_2$ : Breytingarhraði við miðpunkt skrefsins, byggt á  $k_1$ .  $k_3$ : Breytingarhraði aftur við miðpunkt, byggt á  $k_2$ .  $k_4$ : Breytingarhraði við lok skrefsins. Lausnin uppfærir með vegnu meðaltali þessara stuðla:  $y_i + 1 = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$  Lausnir fyrir allar tegundir og tíma eru geymdar í fylkjunum  $yvec$  og  $tvec$ .

Hjálparfallið `ht2` tekur inn tíma  $t$ ,  $y$  vigur með gildum fyrir  $y_1, y_2, y_3$  á tíma  $t$  og fastagildið  $b_1$  sem stýrir áhrifum plöntustofnsins á plöntuætur. Fallið reiknar breytingarhraða tegundanna og skilar vigrinum  $(d\frac{y_1}{dt}, d\frac{y_2}{dt}, d\frac{y_3}{dt})$ .

```
function [tvec, yvec] = RKsolver3(y0,n,T,b_1)
    arguments
        y0; n; T; b_1 = 3; % default gildi ef ekki er sett inn b_1
    end
    y = zeros(length(y0),n);
    y(:,1)=y0;
    h=T/n;
    t=0;
    yvec=[y0]; tvec=[0];
    for i=1:n
        t=t+h;
```

```

        k1=ht2(t,y(:,i),b_1);
        k2=ht2(t+h/2,y(:,i)+h/2.*k1,b_1);
        k3=ht2(t+h/2,y(:,i)+h/2.*k2,b_1);
        k4=ht2(t+h,y(:,i)+h.*k3,b_1);
        y(:,i+1)=y(:,i)+h/6*(k1+2.*k2+2.*k3+k4);
        tvec(i+1)=t;
        yvec(:,i+1)=y(:,i+1);
    end
end

function z = ht2(t,y,b_1)
    arguments
        t; y; b_1 = 3;
    end
    a_1 = 5; a_2 = 0.1; % b_1 = 3;
    b_2 = 2; d_1 = 0.4; d_2 = 0.01;
    z = [
        y(1)*(1 - y(1)) - (a_1*y(1)*y(2))/(1 + b_1*y(1));
        (a_1*y(1)*y(2))/(1 + b_1*y(1)) - d_1*y(2) - (a_2*y(2)*y(3))/(1 + b_2*y(2));
        (a_2*y(2)*y(3))/(1 + b_2*y(2)) - d_2*y(3)
    ];
end

```

Það er síðan geymt í sér skrá að nafni **ht2.m** og í aðal forritinu er skilgreint byrjunarskilyrði, þrepafjölda og lokatíminn. Kóðinn er eftirfarandi:

```

clc; clear; close all

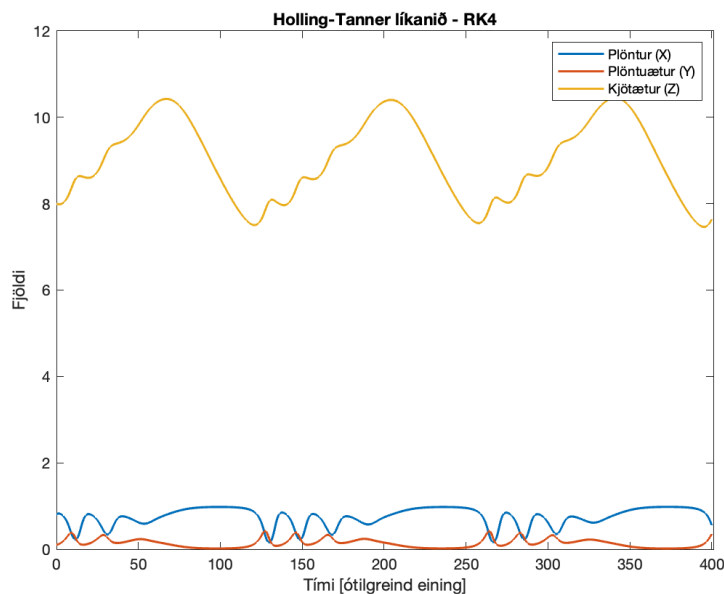
% Fastar:
a_1 = 5; b_1 = 3; a_2 = 0.1;
b_2 = 2; d_1 = 0.4; d_2 = 0.01;

% Gefum okkur lokatíma, skrefastaerð og upphafsgildi
T = 400; n = 100000; y0_1 = [0.8; 0.1; 8];

% Leysum með RK4
[t1,y1] = RKsolver3(y0_1,n,T);

% Plottum graf breytanna sem fall af tíma
plot(t1,y1, LineWidth=1.5)
xlabel("Tími [ótilgreind eining]")
ylabel("Fjöldi")
title("Holling-Tanner líkanið - RK4")
legend("Plöntur (X)", "Plöntuætur (Y)", "Kjötætur (Z)")

```



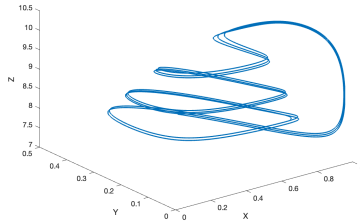
Mynd 10: Lausnir á diffurjöfnuhneppinu fyrir  $t \in [0, 400]$

Grafið á mynd 10 sýnir samspil hjá plöntum og plöntuætum í sveiflum yfir tíma, plöntur stjórna bæði fjölgun og fækkun plöntuæta svo þær tengja saman plöntuætur og kjötætur í sveiflum. Plöntuætur eru einnig undir áhrifum frá kjötætum og þeir stofnar sveiflast í takt, en kjötætur með minni sveifluhraða þar sem fjölgun þeirra er háð tveimur stigum fæðukeðjunnar.

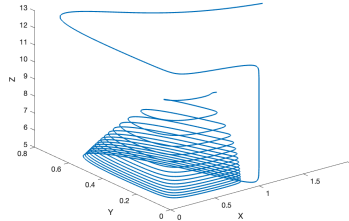
## Dæmi 2

Teiknið fasaferla lausnanna  $(X, Y, Z)$  fyrir nokkur mismunandi upphafsskilyrði.

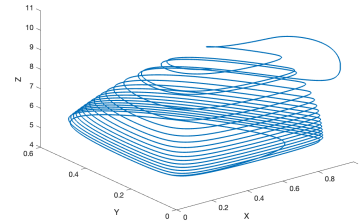
Svar



$$(a) y_0 = \begin{pmatrix} 0.8 \\ 0.1 \\ 8 \end{pmatrix}$$



$$(b) y_0 = \begin{pmatrix} 2 \\ 0.5 \\ 12 \end{pmatrix}$$



$$(c) y_0 = \begin{pmatrix} 0.1 \\ 0.1 \\ 5 \end{pmatrix}$$

Mynd 11: Fasarit af lausnum af jöfnu 7 með mismunandi upphafsgildi

Hér sjáum við að fasaferlar eru allir lotubundnir óháð upphafsgildum. Lotur eru mismunandi en þeir verða allir lotubundnir eftir einhvern tíma. Það sést vel á mynd 11b þar sem upphafsgildi eru langt frá lotubundna ferlinum. Hluti af kóðanum er eftirfarandi (án plot-skipana):

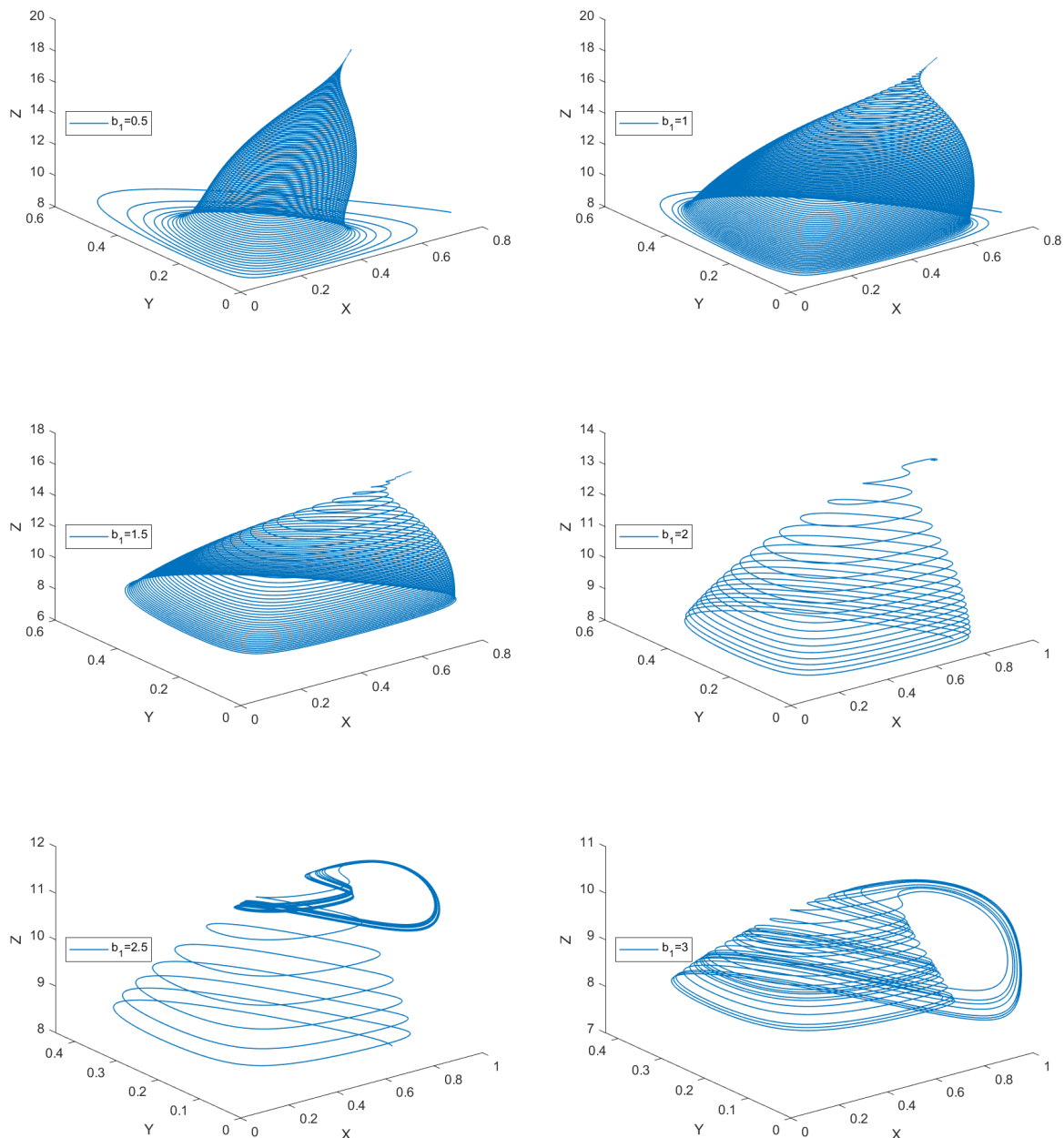
```
T=800;n=100000;  
% Skilgreinum ny upphafsgildi  
y0_1 = [0.8; 0.1; 8];  
y0_2 = [2; 0.5; 12];  
y0_3 = [0.1; 0.1; 5];  
  
% Leysum með RK4  
[t1, y1] = RKsolver3(y0_1,n,T);  
[t2, y2] = RKsolver3(y0_2,n,T);  
[t3, y3] = RKsolver3(y0_3,n,T);
```

### Dæmi 3

Skoðið lausnir fyrir mismunandi  $b_1$  gildi á bilinu  $[0.5, 3]$ . Þið ættuð að sjá nokkrar mismunandi tegundir af lausnum. Hér gæti verið áhugavert að teikna fasaferla og lausn fyrir  $Z$  sem fall af tíma fyrir mismunandi  $b_1$ .

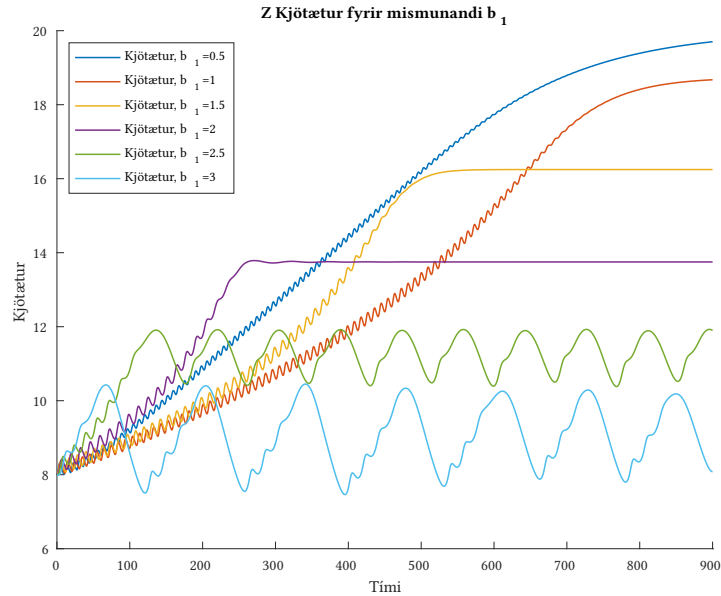
#### Svar

Hér má sjá fasarit fyrir plöntur ( $X$ ), plöntuætur ( $Y$ ) og kjötætur ( $Z$ ) fyrir mismunandi gildi  $b_1$  en upphafsgildin eru  $y_0 = \begin{pmatrix} 0.8 \\ 0.1 \\ 8 \end{pmatrix}$  sem notuð voru í dæmi 1. Notaður var **RKsolver3.m** til að leysa diffurjöfnuhneppin fyrir öll gildi  $b_1$



Mynd 12: Fasarit af lausnum af jöfnu 7 með mismunandi gildi á  $b_1$

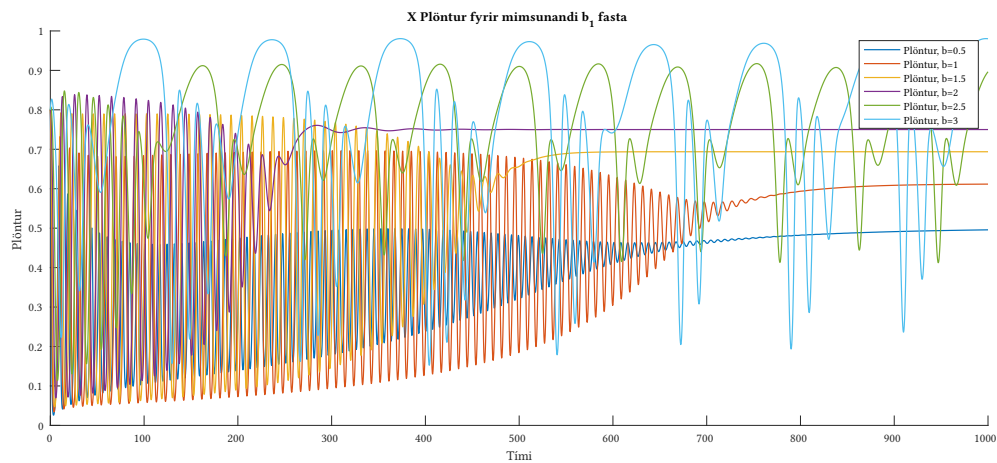
Við skoðum líka lausnir fyrir  $Z$  sem fall af tíma á eftirfarandi grafi:



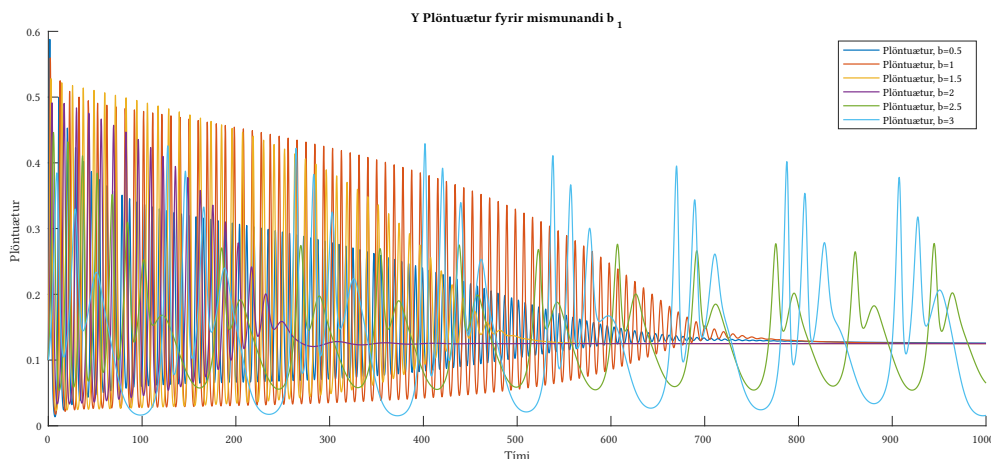
Mynd 13: Lausnir á diffurjöfnuhneppinu fyrir  $t \in [0, 900]$

Línuritið hér að ofan mætti túlka þannig að þegar  $b_1$  stækkar frá 0.5 upp í 3 þá fer  $Z$  úr því að nálgast stöðugleikapunkt kerfisins, í að sveiflast í kringum stöðuleikapunktinn á lotubundinn hátt. Slík hegðun gerist þegar  $b_1 = 2.5$  og  $b_2 = 3$

Einnig er áhugavert að skoða lausnir á  $X$  og  $Y$  diffurjöfnuhneppisins sem fall af tíma



Mynd 14: Lausnir á diffurjöfnuhneppinu fyrir  $t \in [0, 1000]$



Mynd 15: Lausnir á diffurjöfnuhneppinu fyrir  $t \in [0, 1000]$

mynd 14 og 15 eru gríðalega óskýrar en sýna sömu hegðun og mynd 13, þar sem  $b_1 = 2.5$  og  $b_1 = 3.0$  gefa lotubundinn föll en  $b_1 \leq 2$  gefur föll sem eru samleitin („they converge“). Þessi hegðun sést **einnig** á mynd 12 þar sem fasarit sýnir fallið stefna á fast gildi en þegar  $b_1$  stækkar þá verður það lotubundið. Kóðann fyrir myndir 13, 14 og 15 má finna í skjalinu `h3_d3_plot.m` sem notar `RKsolver3.m` og fylgir með skilum.

## Sjálfstæð vinna

### Dæmi 4

Vinnið áfram með líkön fyrir samspil tegunda. Þið megið skoða fleiri eiginleika við þau líkön sem hér hafa verið kynnt, t.d. áhrif breytinga á ákveðnum þáttum. Einnig er hægt að finna og skoða önnur líkön og bera saman við þau sem við höfum skoðað.

### Svar

Það var ákveðið að skoða [Competitive Lotka–Volterra](#) líkan fyrir  $N$ -tegundir, og valið var  $N = 3$ .

Í Competitive Lotka–Volterra líkaninu þá eru diffurjöfnur kerfisins einfaldlega:

$$\frac{dx_i}{dt} = r_i x_i \left( 1 - \sum_{j=1}^N \alpha_{ij} x_j \right) \quad (8)$$

Fyrst að þetta líkan hefur gríðarlega mikið af breytilegum gildum, þá var reint að besta kerfið með tölvu, en besta fyrir hvað? Til þess að besta notum við kostnaðarföll, kostnaðarfall greinir gæðinn á ílaginu sínu, yfirleitt með einni tölu. Í þessu tilveki var reint að besta fyrir sveiflur, fyrst að eftir þínu prófun fannst að flest gildi á  $\alpha$  og  $r$  gefa flatar lausni í enda hermunar.

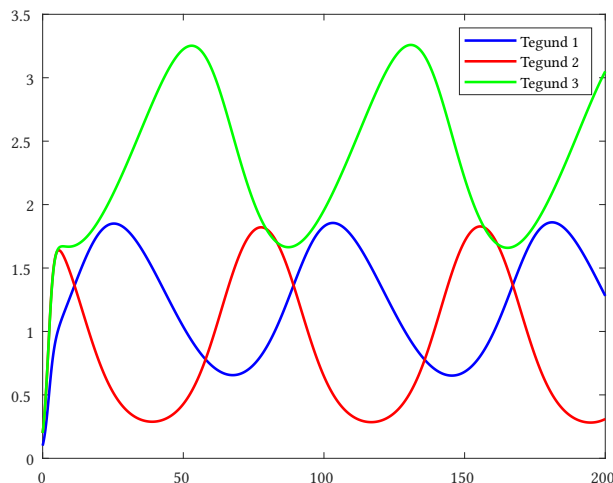
Mörg kostnaðar föll voru prófuð, meðal annars vegðar summur, lægstu og hæstu afleiður og tvíafleiður og svo framleiðis, en ekkert þessara aðferða gáfu sig. Þar afleiðandi var handvirkt fundið stuðla, gefna í jöfnu 9:

$$\alpha = \begin{pmatrix} 0.238238 & 0.195195 & 0.219219 \\ 0.348348 & 0.244244 & 0.145145 \\ 0.232232 & 0.252252 & 0.200200 \end{pmatrix} \quad r = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad (9)$$

Sem gefur eftirfarandi lausn á mynd 16 sem var framleidd með `RK4solveCLV.m` og `skilaverkefni3hluti4.m` skjölunum.



Í samanburð við Lotka–Volterra líkanið og Holling–Tanner líkanið er Competitive Lotka–Volterra mikið meira sveigjanlegt og viðkvæmt, þegar maður fíktar með föstum þá sér maður stundum gríðarlegar breytingar fyrir mjög litlar breytingar á föstum, t.d. gat jafnvægis punktur einnar tegundar farið frá hæsta stigi í lágsta við aðeins 0.01 skref í gildi eins þáttar í  $\alpha$ .



Mynd 16: Lausn á 8 með föstum gefnum í 9 og upphafs gildum  $x_0 = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.2 \end{pmatrix}$

Fyrst jafna 8 hefur nákvæmlega einn stöðugleika punkt sem hefur ekki 0 sem eitt af stökunum þá er hægt að gera eigingildis greiningu í þeim punkt, hinsvegar lítill árangur náðist í þeirri greiningu, því eingin greinamunur gat verið gerður á milli mismunandi gildum á  $\alpha$ .

Kóðinn til að reikna lausn á jöfnu 8 er eftirfarandi:

```
function [tvec, yvec] = RK4solveCLV(y0,n,T)
    y = zeros(length(y0),n);
    y(:,1)=y0;
    h=T/n;
    t=0;
    yvec=[y0];tvec=[0];
    for i=1:n
        t=t+h;
        k1=f(t,y(:,i));
        k2=f(t+h/2,y(:,i)+h/2.*k1);
        k3=f(t+h/2,y(:,i)+h/2.*k2);
        k4=f(t+h,y(:,i)+h.*k3);
        y(:,i+1)=y(:,i)+h/6*(k1+2.*k2+2.*k3+k4);
        tvec(i+1)=t;
        yvec(:,i+1)=y(:,i+1);
    end
end

function z=f(t,x)
    % fallið í diffurjöfnu
    alpha = [
        0.238238, 0.195195, 0.219219;
        0.348348, 0.244244, 0.145145;
        0.232232, 0.252252, 0.200200
    ];
    r = ones(3,1);
```

```
% Differential equations
z = r .* x .* ( 1 - alpha * x);
end
```