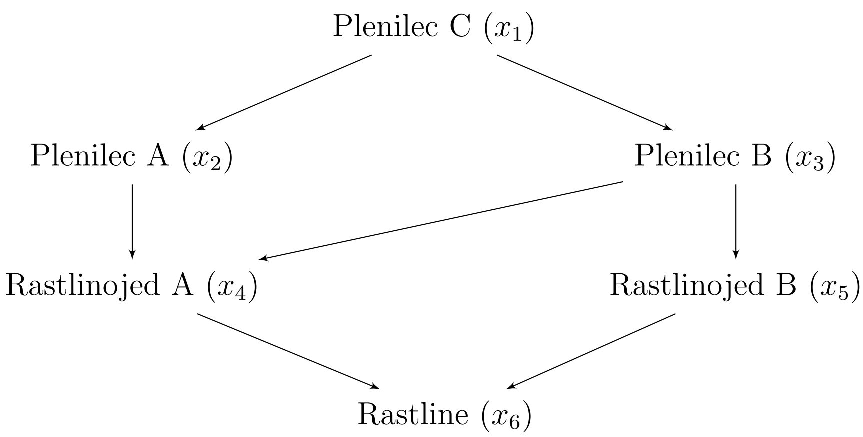
Dinamika populacij v prehranjevalni verigi

Domen Mohorčič, Larsen Cundrič, Mustafa Grabus

7. junij 2020

1 Problem

Modelirali bomo dinamiko populacij različnih vrst, odnosi med vrstami pa so predstavljeni z naslednjim grafom:



Puščice označujejo smer prehranjevanja (npr. Plenilec A se prehranjuje z Rastlinojedom A). Z x_i označimo velikost posamezne populacije. Sprememba velikosti posamezne populacije je tako odvisna od naravnega prirastka/smrtnosti (b_i) in smrtnosti zaradi ulova ter prirastka zaradi prehranjevanja z drugo vrsto (a_{ij}). Ulov in prehranjevanje sta odvisna od velikosti ustreznih drugih populacij, naravni prirastek in smrtnost pa sta odvisna samo od velikosti populacije. Dinamiko i-te populacije lahko opišemo tako:

$$\dot{x}_i = x_i \cdot b_i + \sum_{i \neq j} a_{ij} \cdot x_i \cdot x_j \tag{1}$$

kjer je x_i velikost i-te populacije, b_i je koeficient naravnega prirastka/smrtnosti, $a_{ij} \cdot x_i \cdot x_j$ pa je sprememba i-te populacije glede na interakcijo z j-to populacijo.

2 Reševanje

2.1 Naloga 1

Sistem diferencialnih enačb za podani sistem je naslednji:

$$\begin{aligned}
\dot{x}_1 &= x_1 \cdot (b_1 + a_{12} \cdot x_2 + a_{13} \cdot x_3) \\
\dot{x}_2 &= x_2 \cdot (b_2 + a_{21} \cdot x_1 + a_{24} \cdot x_4) \\
\dot{x}_3 &= x_3 \cdot (b_3 + a_{31} \cdot x_1 + a_{34} \cdot x_4 + a_{35} \cdot x_5) \\
\dot{x}_4 &= x_4 \cdot (b_4 + a_{42} \cdot x_2 + a_{43} \cdot x_3 + a_{46} \cdot x_6) \\
\dot{x}_5 &= x_5 \cdot (b_5 + a_{53} \cdot x_3 + a_{56} \cdot x_6) \\
\dot{x}_6 &= x_6 \cdot (b_6 + a_{64} \cdot x_4 + a_{65} \cdot x_5)
\end{aligned}$$

x_i predstavlja velikost i-te populacije, $\dot{x}_i$ pa predstavlja spremembo i-te populacije v nekem trenutku. Velikosti posameznih vrst bomo zapisali v vektor $X = [x_1, x_2, x_3, x_4, x_5, x_6]^T$. Sistem diferencialnih enačb pa bomo zapisali v vektor $\dot{X}$:

$$\dot{X} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} x_1 \cdot (b_1 + a_{12} \cdot x_2 + a_{13} \cdot x_3) \\ x_2 \cdot (b_2 + a_{21} \cdot x_1 + a_{24} \cdot x_4) \\ x_3 \cdot (b_3 + a_{31} \cdot x_1 + a_{34} \cdot x_4 + a_{35} \cdot x_5) \\ x_4 \cdot (b_4 + a_{42} \cdot x_2 + a_{43} \cdot x_3 + a_{46} \cdot x_6) \\ x_5 \cdot (b_5 + a_{53} \cdot x_3 + a_{56} \cdot x_6) \\ x_6 \cdot (b_6 + a_{64} \cdot x_4 + a_{65} \cdot x_5) \end{bmatrix} \tag{2}$$

V vektorju $b = [b_1, b_2, b_3, b_4, b_5, b_6]^T$ so koeficienti naravnega prirastka/smrtnosti, ki je za rastline (x_6) pozitiven, za vse ostale pa negativen.
Sistem enačb za dani primer lahko tako zapišemo kot

$$\dot{X} = X * (b + A \cdot X) \tag{3}$$

($*$ predstavlja množenje po elementih), kjer matrika A vsebuje koeficiente hranjenja in plenjenja a_{ij}:

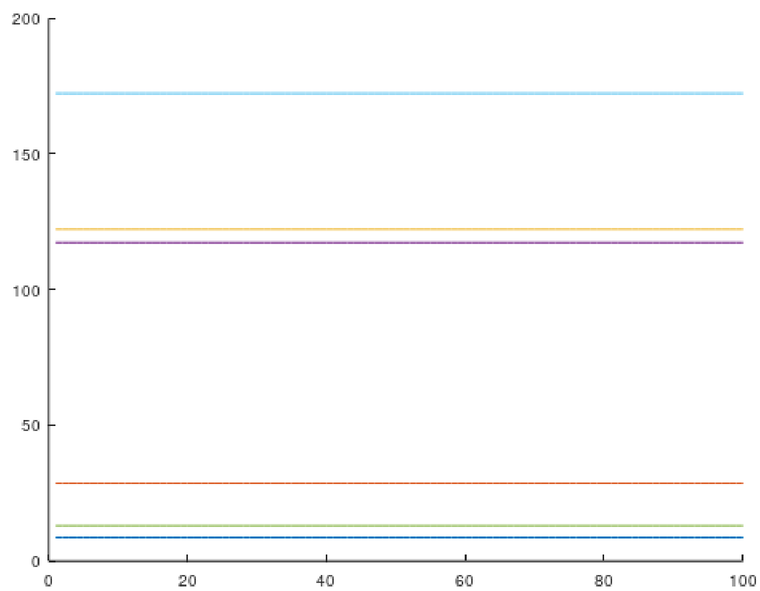
$$A = \begin{bmatrix} 0 & a_{12} & a_{13} & 0 & 0 & 0 \\ a_{21} & 0 & 0 & a_{24} & 0 & 0 \\ a_{31} & 0 & 0 & a_{34} & a_{35} & 0 \\ 0 & a_{42} & a_{43} & 0 & 0 & a_{46} \\ 0 & 0 & a_{53} & 0 & 0 & a_{56} \\ 0 & 0 & 0 & a_{64} & a_{65} & 0 \end{bmatrix} \tag{4}$$

Koeficient a_{ij} predstavlja interakcijo vrste i z vrsto j. Pozitivna vrednost pomeni, da se vrsta i prehranjuje z vrsto j, negativna pa, da je vrsta i plen vrste j. Vrsta s sabo nima take interakcije, zato so elemeniti, kjer je $i = j$, enaki $a_{ij} = 0$. Koeficienta a_{ij} in a_{ji} pa si nista nujno nasprotna, saj lahko plenilska vrsta poje več plena, kot pa ima koristi od tega.
Za začetek smo določili $b = [-0.1, -0.1, -0.1, -0.05, -0.05, 0.3]^T$ in

$$A = \begin{bmatrix} 0 & 0.5 & 0.7 & 0 & 0 & 0 \\ -2 & 0 & 0 & 1 & 0 & 0 \\ -8 & 0 & 0 & 1 & 4 & 0 \\ 0 & -6 & -1 & 0 & 0 & 2 \\ 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & -2 & -5 & 0 \end{bmatrix} \cdot 0.001 \tag{5}$$

2.2 Naloga 2

Stacionarna rešitev sistema je, ko je vektor $\dot{X}$ enak 0. To pomeni, da je $X * (b + A \cdot X) = 0$. Vektor $X = 0$ že zadosti našemu pogoju, vendar iščemo neničelno rešitev. Za naša A in b je to $X = [8.7, 28.7, 122.3, 117.4, 13.0, 172.3]^T$. Slika rešitve po 100 iteracijah izgleda tako:



2.3 Naloga 3

Napisali smo funcijo simulatePopulation.m, ki za vhod vzame matriko interakcij A, vektor naravnih prirastkov b in začetne velikosti populacij v vektorju

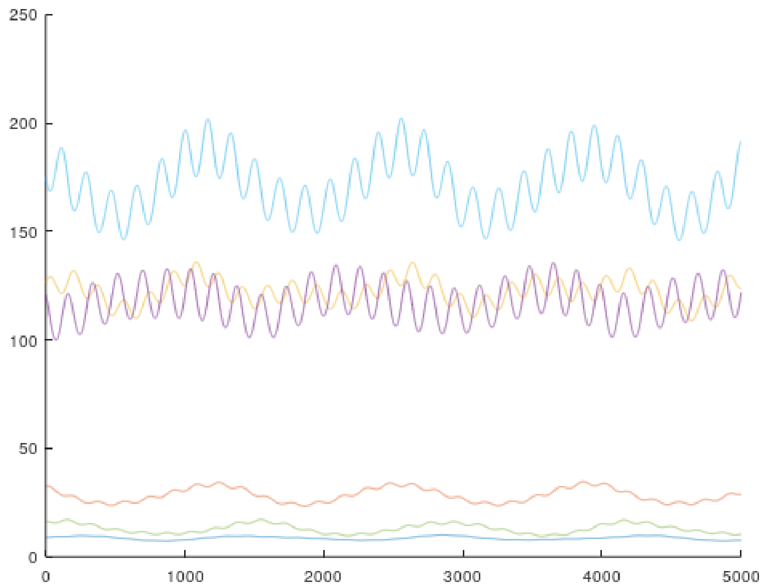
X. Funkcija za simulacijo uporabi metodo Runge-Kutta četrte stopnje, na koncu pa izriše gibanje sistema.

```
function lastX = simulatePopulation (x0, b, A, n, fig)
    F = @(X, b, A) X.*(b+A*X); %enacba prirastka za vsako vrsto
    h = 0.1; %dolzina koraka

    tocke = zeros(6, n);
    x = x0;

    %Simulacija
    for i = (1:n)
        x = x + rk4Step(F, x, h, b, A);
        tocke(:, i) = x;
    end

    figure(fig)
    hold on;
    for i = (1:6)
        vektor = zeros(1, n);
        vektor(1, :) = tocke(i, :);
        plot(vektor);
    endfor
    hold off;

    lastX = x;
endfunction

%%Korak rk4
function val = rk4Step(f, x0, h, b, A)
    k1=feval(f, x0, b, A);
    k2=feval(f, x0 + k1*h/2, b, A);
    k3=feval(f, x0 + k2*h/2, b, A);
    k4=feval(f, x0 + k3*h, b, A);
    val = h*(k1+2*k2+2*k3+k4)/6;
endfunction
```

2.4 Naloga 4

Naredili smo simulacijo za začetne pogoje, ki se malo razlikujejo od X. Določili smo $X_{nov} = [9.0, 32.6, 125.9, 121.0, 16.5, 175.4]^T$, slika simulacije po

5000 iteracijah pa izleda tako:



Sistem se za naše vrednosti (A, b in X_{nov}) obnaša ciklično.

2.5 Naloga 5

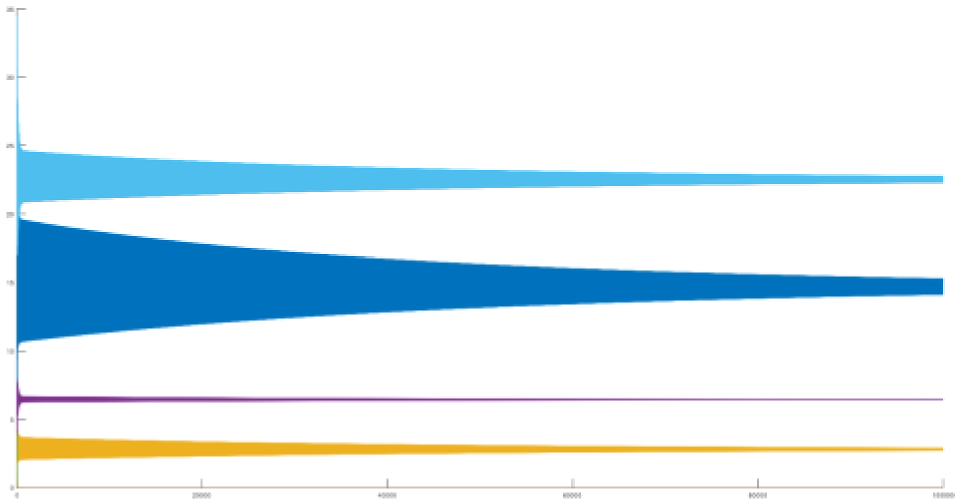
Preučili smo obnašanje sistema za različne vrednosti koeficientov in začetnih pogojev in našli ciklično obnašanje, asimptotično ciklično obnašanje in kaos. Pri iskanju teh sistemov smo si pomagali s funkcijo generateSystem.m, ki nam vrne matriko A in vektor b, oba generirana naključno.

Asimptotično ciklično obnašanje, kjer imamo spiralni lijak, smo dobili z naslednjimi začetnimi vrednostmi:

$$A = \begin{bmatrix} 0 & 0.37939 & 1.38080 & 0 & 0 & 0 \\ -1.10424 & 0 & 0 & 0.65246 & 0 & 0 \\ -0.30582 & 0 & 0 & 1.24509 & 0.40233 & 0 \\ 0 & -0.99713 & -0.73865 & 0 & 0 & 0.39387 \\ 0 & 0 & -0.35715 & -0 & 0 & 0.39208 \\ 0 & 0 & 0 & -1.50798 & -1.78165 & 0 \end{bmatrix},$$

$$b = \begin{bmatrix} -3.89109 \\ -0.27347 \\ -3.53131 \\ -6.79120 \\ -8.91734 \\ 9.72807 \end{bmatrix} \text{ in } X = \begin{bmatrix} 0.77988 \\ 1.22798 \\ 2.48059 \\ 1.73900 \\ 3.98827 \\ 25.00319 \end{bmatrix}.$$

Vektor X se od stacionarne rešitve sistema razlikuje samo za 0.00012384. Slika simulacije sistema s takimi podatki pa je naslednja:



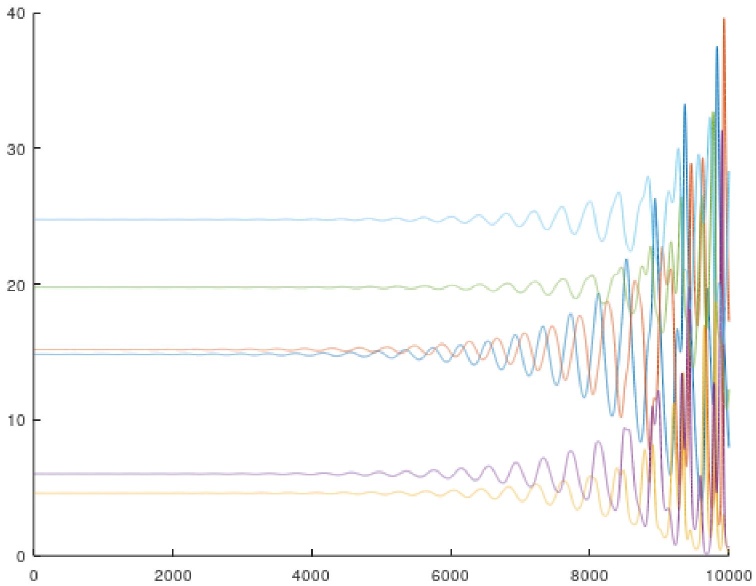
.

Asimptotično ciklično obnašanje s spiralnim izvorom pa smo dobili z naslednjimi začetnimi vrednostmi:

$$A = \begin{bmatrix} 0 & 0.000084 & 0.027628 & 0 & 0 & 0 \\ -0.011626 & 0 & 0 & 0.032503 & 0 & 0 \\ -0.033349 & 0 & 0 & 0.038057 & 0.028158 & 0 \\ 0 & -0.026921 & -0.033088 & 0 & 0 & 0.028894 \\ 0 & 0 & -0.008998 & 0 & 0 & 0.009693 \\ 0 & 0 & 0 & -0.004489 & -0.006963 & 0 \end{bmatrix},$$

$$b = \begin{bmatrix} -0.128572 \\ -0.023682 \\ -0.291557 \\ -0.154676 \\ -0.198708 \\ 0.164787 \end{bmatrix} \text{ in } X = \begin{bmatrix} 14.8432 \\ 15.1853 \\ 4.6090 \\ 6.0374 \\ 19.7722 \\ 24.7789 \end{bmatrix}.$$

Vektor X se od stacionarne rešitve razlikuje za 0.0023216. Slika simulacije sistema izgleda tako:



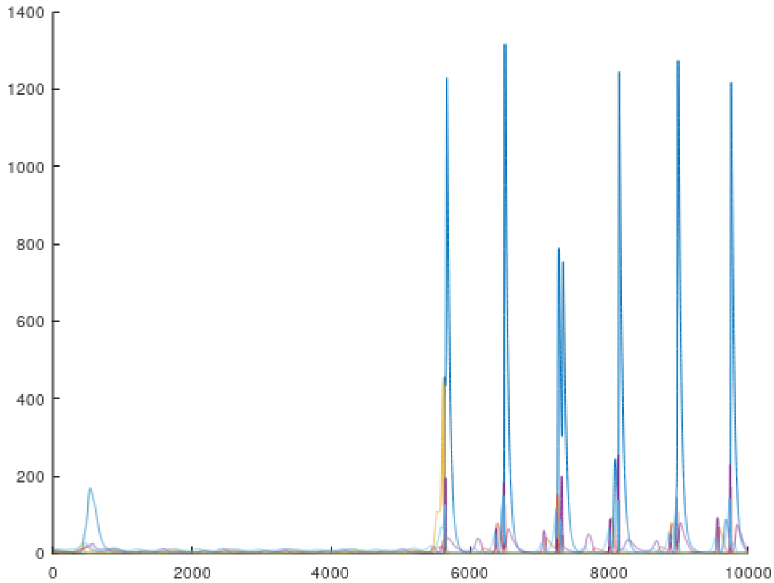
.

Kaos smo dobili z naslednjimi koeficienti:

$$A = \begin{bmatrix} 0 & 0.024710 & 0.024893 & 0 & 0 & 0 \\ -0.003471 & 0 & 0 & 0.024640 & 0 & 0 \\ -0.027545 & 0 & 0 & 0.036172 & 0.027497 & 0 \\ 0 & -0.034815 & -0.004165 & 0 & 0 & 0.028781 \\ 0 & 0 & -0.029746 & 0 & 0 & 0.018492 \\ 0 & 0 & 0 & -0.014320 & -0.039922 & 0 \end{bmatrix},$$

$$b = \begin{bmatrix} -0.245517 \\ -0.094402 \\ -0.114666 \\ -0.113336 \\ -0.150582 \\ 0.235885 \end{bmatrix} \text{ in } X = \begin{bmatrix} 7.2023 \\ 7.9662 \\ 1.7152 \\ 4.6668 \\ 4.4584 \\ 12.1101 \end{bmatrix}.$$

Vektor X se od stacionarne rešitve razlikuje za 1.9694, za sistem pa po 10000 iteracijah dobimo naslednjo naslednjo sliko:



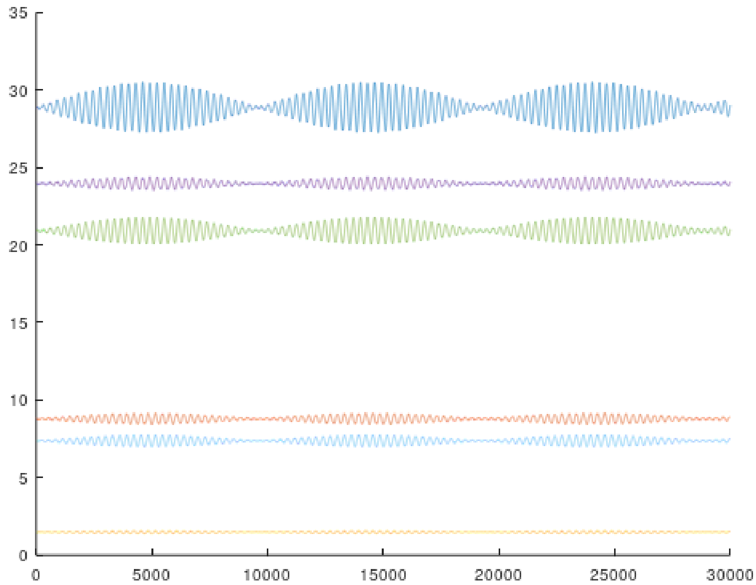
.

Ciklično obnašanje smo našli z naslednjimi koeficienti:

$$A = \begin{bmatrix} 0 & 0.021620 & 0.037477 & 0 & 0 & 0 \\ -0.011555 & 0 & 0 & 0.026287 & 0 & 0 \\ -0.026978 & 0 & 0 & 0.012645 & 0.027949 & 0 \\ 0 & -0.030686 & -0.019335 & 0 & 0 & 0.041368 \\ 0 & 0 & -0.033011 & 0 & 0 & 0.030491 \\ 0 & 0 & 0 & -0.000749 & -0.012079 & 0 \end{bmatrix},$$

$$b = \begin{bmatrix} -0.2454399 \\ -0.2962493 \\ -0.1088634 \\ -0.0064447 \\ -0.1757772 \\ 0.2704713 \end{bmatrix} \text{ in } X = \begin{bmatrix} 28.8557 \\ 8.7814 \\ 1.4681 \\ 23.9553 \\ 20.9203 \\ 7.3815 \end{bmatrix}.$$

Vektor X se od stacionarne rešitve razlikuje za 0.025147, slika simulacije pa je naslednja:



.

Obnašanje sistema pa je odvisno od lastnih vrednosti jakobijeve matrike sistema v stacionarni točki. Jakobijeva matrika je enaka:

$$J_{x_1} = \begin{bmatrix} b_1 + x_2a_{12} + x_3a_{13} + x_4a_{14} + x_5a_{15} + x_6a_{16} \\ x_2a_{21} \\ x_3a_{31} \\ x_4a_{41} \\ x_5a_{51} \\ x_6a_{61} \end{bmatrix} \tag{6}$$

$$J_{x_2} = \begin{bmatrix} x_1a_{12} \\ b_2 + x_1a_{21} + x_3a_{23} + x_4a_{24} + x_5a_{25} + x_6a_{26} \\ x_3a_{32} \\ x_4a_{42} \\ x_5a_{52} \\ x_6a_{62} \end{bmatrix} \tag{7}$$

$$J_{x_3} = \begin{bmatrix} x_1a_{13} \\ x_2a_{23} \\ b_3 + x_1a_{31} + x_2a_{32} + x_4a_{34} + x_5a_{35} + x_6a_{36} \\ x_4a_{43} \\ x_5a_{53} \\ x_6a_{63} \end{bmatrix} \tag{8}$$

$$J_{x_4} = \begin{bmatrix} x_1a_{14} \\ x_2a_{24} \\ x_3a_{34} \\ b_4 + x_1a_{41} + x_2a_{42} + x_3a_{43} + x_5a_{45} + x_6a_{46} \\ x_5a_{54} \\ x_6a_{64} \end{bmatrix} \tag{9}$$

$$J_{x_5} = \begin{bmatrix} x_1a_{15} \\ x_2a_{25} \\ x_3a_{35} \\ x_4a_{45} \\ b_5 + x_1a_{51} + x_2a_{52} + x_3a_{53} + x_4a_{54} + x_6a_{56} \\ x_6a_{65} \end{bmatrix} \tag{10}$$

$$J_{x_6} = \begin{bmatrix} x_1a_{16} \\ x_2a_{26} \\ x_3a_{36} \\ x_4a_{46} \\ x_5a_{56} \\ b_6 + x_1a_{61} + x_2a_{62} + x_3a_{63} + x_4a_{64} + x_5a_{65} \end{bmatrix} \tag{11}$$

$$JF = \begin{bmatrix} J_{x_1} & J_{x_2} & J_{x_3} & J_{x_4} & J_{x_5} & J_{x_6} \end{bmatrix} \tag{12}$$

Lastne vrednosti jacobijeve matrike v stacionarni točki so kompleksne. Imaginarni del skrbi za ciklično obnašanje, od njihovega realnega dela pa je odvisno obnašanje sistema. Če je realni del lastne vrednosti negativen, je stacionarna točka spiralno privlačna, če je pozitiven, je spiralno odbojna. Če je realni del enak 0, imamo krožnico.
Obnašanje sistema okoli točke, ki ima različno predznačene realne dele lastnih vrednosti, je težje opisati. Okoli take točke je več podprostorov, v katerih se sistem obnaša različno. Vektorski podprostor je lahko stabilen (spiralno privlačen proti točki), če imajo lastne vrednosti negativen realni del. Tak prostor razpenjajo lastni vektorji takih lastnih vrednosti. Vektorski podprostor je lahko tudi nestabilen (spiralno odbija stran od točke), če imajo lastne vrednosti pozitiven realni del. Tega razpenjajo pripadajoči lastni vektorji. Vektorski podprostor v okolici stacionarne točke pa je lahko tudi center (sistem kroži), če obstajajo lastne vrednosti brez realnega dela. Ta podprostor pa razpenjajo lastni vektorji, ki pripadajo tem lastnim vrednostim.
Sistem ima tako splošno rešitev, ki je naslednja:

$$x(t) = e^{\alpha t}((C_1 cos(\beta t) + C_2 sin(\beta t))u + (-C_1 sin(\beta t) + C_2 cos(\beta t))w) \tag{13}$$

Lastna vrenost Jacobijeve matrike v stacionarni točki takega sistema je oblike $\lambda_{1,2} = \alpha \pm \beta i$. V zgornji enačbi je α realni del lastne vrednosti in β imaginarni del lastne vrednosti. Vektor u je realni del lastnega vektorja, w pa je imaginarni del lastnega vektorja. Ker sta lastni vrednosti konjugirani, sta konjugirana tudi njuna lastna vektorja, zato v enačbi nastopa samo eden.
Ker pa imamo sistem šest spremenljivk, imamo tudi šest lastnih vrednosti $(\lambda_1, \dots, \lambda_6)$ in šest lastnih vektorjev $(v_1, \dots, v_6)$. Ker pa so med lastnimi vrednostmi in lastnmi vektorji konjugirani pari, ki jih lahko zapišemo tako:

$$\begin{aligned} \lambda_{1,2} &= \alpha_1 \pm \beta_1 i \\ \lambda_{3,4} &= \alpha_3 \pm \beta_3 i \\ \lambda_{5,6} &= \alpha_5 \pm \beta_5 i \end{aligned}$$

in

$$\begin{aligned} v_{1,2} &= u_1 \pm w_1 i \\ v_{3,4} &= u_3 \pm w_3 i \\ v_{5,6} &= u_5 \pm w_5 i \end{aligned}$$

Tako nam da enačba (13) sistem treh generalnih rešitev za naš problem:

$$x_1(t) = e^{\alpha_1 t}((C_1 cos(\beta_1 t) + C_2 sin(\beta_1 t))u_1 + (-C_1 sin(\beta_1 t) + C_2 cos(\beta_1 t))w_1) \tag{14}$$

$$x_3(t) = e^{\alpha_3 t}((C_3 cos(\beta_3 t) + C_4 sin(\beta_3 t))u_3 + (-C_3 sin(\beta_3 t) + C_4 cos(\beta_3 t))w_3) \tag{15}$$

$$x_5(t) = e^{\alpha_5 t}((C_5 cos(\beta_5 t) + C_6 sin(\beta_5 t))u_5 + (-C_5 sin(\beta_5 t) + C_6 cos(\beta_5 t))w_5) \tag{16}$$

splošna rešitev pa je linearna kombinacija le teh:

$$x(t) = x_1(t) + x_3(t) + x_5(t) \tag{17}$$

3 Zaključek in komentar

Za seminarsko nalogo smo modelirali obnašanje prehranjevalne verige, predstavljene v poglavju (1). Napisali smo sistem diferencialnih enačb za podan prehranjevalni sistem, ga simulirali pri različnih začetnih vrednostih in zarličnih koeficientih ter predstavili nekaj različnih obnašanj sistema. Na koncu smo še razložili, kako je gibanje sistema v okolici stacionarnih točk povezano z lastnimi vrednostmi jacobijeve matrike v tej točki.

4 Program

Napisali smo naslednje funkcije:

- generateSystem.m: Za parametre sprejme dva koeficienta, ki povesta, koliko so velike vrednosti matrike A in vektorja b. Funkcije vrne matriko koeficientov prehranjevanja A, vektor naravnega prirastka b in stacionarno rešitev podanega sistema X.

- jacobian.m: Funkcija sprejme matriko A in vektorja b in X. Vrne Jacobijevo matriko tega sistema.

- simulatePopulation.m: Funkcija sprejme matriko A, vektorja b in X, število iteracij n in število, kam naj izriše simulacijo sistema. Funkcija vrne zadnji X po simulaciji.

- eigValuesB.m: Funkcija sprejme matriko A in vektorja b in X. Vrne vektor lastnih vrednosti in matriko lastnih vektorjev za Jacobijevo matriko pri podanih argumentih.

- main.m: Funkcija deluje kot glavni program in povezuje vse prej naštete funkcije. Najprej zgenerira naključen sistem z generateSystem, nato pa izpiše rezultate A, b in $X0$. Kliče se funkcija eigValuesB, ki izpiše lastne vrednosti generiranega sistema. Naredi se spremenljivka X, ki je naključno odmaknjena od $X0$. Na koncu se kliče funkcija simulatePopulation, ki simulira prej generirani sistem pri novi začetni vrednosti X.

5 Avtorji

Domen Mohorčič je sestavil poročilo, napisal funkcijo generateSystem.m za naključno generiranje sistema in raziskal klasifikacijo stacionarne točke v večdimenzionalnih sistemih.
Larsen Cundrič je napisal funkcijo eigValuesB.m, ki vrne lastne vrednosti in lastne vektorje, ter napisal funkcijo main.m, ki povezuje vse napisane funkcije. Poiskal je tudi različne sisteme, ki so opisani v poglavju 2.5.
Mustafa Grabus je napisal funkcijo simulatePopulation.m, ki simulira sistem za dane podatke. Napisal je tudi funkcijo jacobian.m, ki vrne vrednost Jacobijeve matrike pri danih podatkih.

6 Viri

Z zunanjimi viri smo si pomagali pri klasifikaciji stacionarnih točk:

- https://math.stackexchange.com/questions/1348606/stability-of-equilibria-for-n-dimensional-nonlinear-systems-of-differential-eq
- https://en.wikipedia.org/wiki/Center_manifold