Diferencne jednačine

Posmatramo Košijev zadatak

$$\frac{\partial y}{\partial t} = f(t, y), \quad y(t_0) = a. \tag{*}$$

Prvi izvod jednačine (*) može se aproksimirati podeljenom razlikom unapred sa korakom Δt:

$$\frac{y(t+\Delta t) - y(t)}{\Delta t} = f(t, y).$$

Uvođenjem smene $t_k = t_0 + k \cdot \Delta t$ i $y_k = y(t_k)$ dobija se izraz

$$y_{n+1} = y_n + \Delta t \cdot f(t_n, y_n), \quad y_0 = a \tag{**}$$

koji predstavlja Ojlerovu eksplicitnu metodu za rešavanje Košijevog zadatka za diferencijalne jednačine. Jednačina (**) predstavlja linarnu *diferencnu jednačinu prvog reda* koja se u opštem obliku zapisuje korišćenjem poznatih nizova realnih brojeva $\{a_n\}$ i $\{b_n\}$ na sledeći način:

$$y_{n+1} = a_n y_n + b_n. \tag{***}$$

Linearna diferencna jednčina I reda je homogna ukoliko je $b_n = 0$, inače je nehomogena.

U zavisnosti od vrednosti nizova brojeva $\{a_n\}$ i $\{b_n\}$ rešenje diferencne jednačine (***) se može zapisati kao:

$$\begin{aligned} y_{n+1} &= \alpha a^n + \sum_{k=0}^{n-1} a^{n-k-1} b_k, \quad n = 1,2 \ldots, \\ a_n &= const = a, \\ y_0 &= \alpha. \end{aligned} \tag{1}$$

$$\begin{aligned} y_{n+1} &= \left(\prod_{i=0}^{n-1} a_i\right)\alpha + \sum_{r=0}^{n-1}\left(\prod_{i=r+1}^{n-1} a_i\right) b, n = 1,2 \ldots, \\ b_n &= const = b, \\ y_0 &= \alpha \end{aligned} \tag{2}$$

$$y_{n+1} = \begin{cases} a + bn & , a_n = a = 1 \\ \left(\alpha - \frac{b}{1-a}\right) a^n + \frac{b}{1-a} & , a_n = a \neq 1 \end{cases}, \quad b_n = b, n = 0,1,2 \ldots \tag{3}$$

MEDICINSKA PRAKSA - LEKOVI:

Poznato je da se određeni lekovi pacijentima daju na svakih h sati. Obeležimo sa S_n količinu odgovarajućeg leka u krvnom sistemu pacijenta nakon n-tog unošenja. Pretpostavimo da između dva unošenja leka organizam pacijenta eliminiše određeni deo unete količine leka (na primer, eliminiše se p procenata leka od ukupne unete količine). Ukoliko je početna doza leka S_0, odrediti količinu leka nakon n unošenja (S_n), kao i količinu leka u krvotoku organizma ako takav lek treba da se unosi doživotno ($\lim_{n\to\infty} S_n$). Rešiti problem u slučaju unosa $2mg$ leka ako se zna da se 40% količine leka koja se nalazi u krvnom sistemu razgradi u roku od 6h.

Rešenje:

Ako je količina leka u krvnom sistemu u $(n+1)$-om intervalu jednaka količini leka u krvnom sistemu u n-tom intervalu umanjena za p-ti deo i uvećana za dozu S_0, možemo da formiramo diferencnu jednačinu

$$S_{n+1} = (1-p)S_n + S_0, n = 1,2,\dots$$

Rešavanjem jednačine (slučaj (3), $a_n = 1-p, b_n = S_0$) dobija se sledeći izraz:

$$S_n = \left[S_0 - \frac{S_0}{p}\right](1-p)^n + \frac{S_0}{p} = S_0\left[\left(1-\frac{1}{p}\right)(1-p)^n + \frac{1}{p}\right] = \frac{S_0}{p}\left[1-(1-p)^{n+1}\right].$$

Granična vrednost unete količine leka iznosi $S^* = \lim_{n\to\infty} S_n = \frac{S_0}{p}$.

Neka je, sada $S_0 = 2\,mg$ i neka na svakih $6h$ ($h = 6$) telo eliminiše 40% leka ($p = 0.4$), tada se dobija sledeća jednačina:

$$S_{n+1} = 0.6\,S_n + 2, S_0 = 2\,, n = 1,2,\dots$$

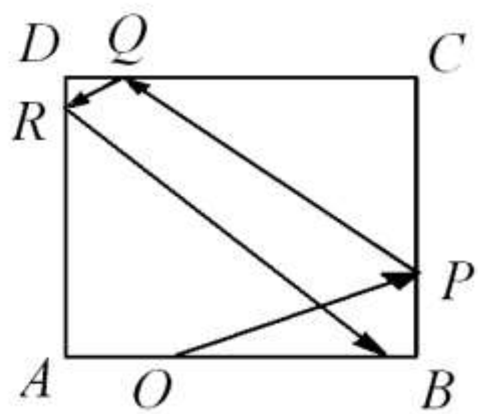
U tabeli je prikazano stanje unetog leka u krvotoku organizma nakon $n = 1,2,3,\dots,7$ unošenja:

N	0	1	2	3	4	5	6	7
S_n	2	3.2	3.92	4.352	4.6112	4.7667	4.86	4.916

Sa obzirom da je $S^* = \lim_{n\to\infty} S_n = \frac{2}{0.4} = 5$ može se zaključiti da opisanom dinamikom količina unetog leka neće prelaziti 5mg. Dakle, ako se na svakih 6h unosi $2mg$ leka, tokom dužeg vremenskog perioda, količina unetog leka neće preći $5mg$. Korišćenjem opisanog metoda možemo rešavati obrnuti problem, tj. možemo odrediti količinu leka koja treba da se unosi tokom nekog perioda kako bi u organizmu bila konstantna doza leka S^*.

KRETANJE LOPTICE PO BILIJARSKOM STOLU:

Posmatrajmo bilijarski sto kvadratnog oblika, čija je jedna stranica dužine a. Obeležimo temena tog kvadrata sa A, B, C i D. Pretpostavimo da se bilijarska loptica može udariti dovoljno jako da se može duže vreme kretati po stolu. Ukoliko se loptica pozicionirana u tački O, na stranici AB, udari pod uglom α, odrediti poziciju koju će loptica udariti na stranici AB nakon n-tog obilaska po stolu. Pretpostaviti da se loptica odbija po pravilu „ulazni ugao je jednak izlaznom uglu" i da će nakon odbijanja od stranice AB udariti stranicu BC, potom stranice CD i DA (tim redom) pre nego što ponovo udari stranicu AB. Odrediti ugao pod kojim je potrebno da se udari loptica tako da se nakon obilaska svih drugih stranica ponovo vrati u tu tačku.



Rešenje:

Obeležimo sa O_n i O_{n+1} mesto koje loptica udara na stranici AB nakon n-tog i $(n+1)$-og obilaska po stolu. Označimo dalje, sa $ax_n = AO_n$, odnosno $ax_{n+1} = AO_{n+1}$. Tada je

$$O_nB = a - ax_n = a(1-x_n)$$

Ako su P_n, Q_n i R_n tačke udara loptice u BC, CD i DA redom, nakon n udara, imamo

$$P_nB = a(1-x_n)\tan\alpha$$

$$P_nC = a - a(1-x_n)\tan\alpha$$

$$Q_nC = a\cot\alpha - a(1-x_n)$$
$$Q_nD = a(1-\cot\alpha) + a(1-x_n)$$
$$R_nD = a(\tan\alpha - 1) + a(1-x_n)\tan\alpha$$
$$R_nA = a(2-\tan\alpha) - a(1-x_n)\tan\alpha$$
$$AQ_{n+1} = a(2\cot\alpha - 1) - a(1-x_n) = ax_{n+1}$$

Dakle, dobija se da je

$$x_{n+1} = x_n + 2(\cot\alpha - 1)$$

odnosno

$$x_n = x_0 + 2n(\cot\alpha - 1), n = 1,2,\ldots$$

gde je $AO = ax_0$ a O startna tačka na stranici AB bilijarskog stola.

Loptica će se kretati uvek po istoj putanji ukoliko važi $\cot\alpha - 1 = 0$, odnosno za početni ugao $\alpha = \pi/4$. Primetimo da u tom slučaju putanja ne zavisi od početne pozicije.
Diskutovati kretanje loptice za $\alpha \in [0, \pi]$.

EKONOMSKI MODELI

PRIMER 1

Odrediti broj godina potrebnih da se određena suma novca uložena u banku udvostruči, ako se na nju primenjuje složeno ukamaćivanje (kamata na kamatu) s kamatnom stopom od $p = 2\%$ godišnje.
Rešenje:
Označimo sa A_n iznos novca na kraju *n*-te godine. Matematički model formiramo na sledeći način:

$$A_n = A_{n-1} + pA_{n-1} = (1+p)A_{n-1}$$

gde je p kamatna stopa. Rešavamo poslednju diferencnu jednačinu, dobijamo da je

$$A_n = (1+p)^n A_0$$

gde je A_0 uloženi iznos novca. Prema uslovima zadatka, $A_n = 2A_0$ se dobija za

$$2A_0 = (1+p)^n A_0 \quad \rightarrow \quad n = \frac{log2}{\log(1+r)} = \frac{log2}{\log(1+0.02)} = 35.0027$$

Dakle, za 35 godina suma uloženog novca će se odvostručiti.

PRIMER 2

Pretpostavimo da se kamatna suma novca R deponuje na kraju svakog obračunskog perioda u nekoj banci, pri čemu se na taj novac primenjuje složeni kamatni račun (kamata na kamatu) sa stopom *r* po svakom obračunskom periodu. Koliko novca banka duguje na kraju svakog obračunskog perioda?
Rešenje:
Iznos koji banka duguje na kraju (n+1)og obračnunskog perioda jednak je zbiru iznosa novca koji banka duguje na kraju *n*-tog perioda, kamate obračunate na taj iznos po stopi *r* i novca u iznosu *R* koji se uplaćuje za svaki obračunski period, tj. $A_{n+1} = (1+r)A_n + R, n = 1,2,..$ gde je $A_0 = 0$. Rešenje ove diferencne jednačine je

$$A_n = R\,\frac{(1+r)^n - 1}{r}.$$

PRIMER 3

Napraviti amortizacioni plan po principu mesečne otplate zajma od 100 evra uz kamatnu stopu od 5% mesečno. Amortizacioni plan treba da sadrži mesec (tj. redni broj rate), neplaćeni deo glavnice početkom meseca, iznos rate otplate duga na kraju meseca (anuitet), strukturu anuiteta koja pordazumeva iznos kamate obračunate na neplaćeni deo duga na kraju obračunskog presioda (tj. meseca) i deo otplate glavnice.

Rešenje:

Neka je, nakon *n*-te uplate g_n neoptplaćeni deo duga p_n. Formiramo sledeću jednačinu:

$$p_{n+1} = p_n + rp_n - g_n = (1+r)p_n - g_n, \ n = 0,1,2 \ldots$$

Rešavanjem jednačine dobijamo da je

$$p_n = (1+r)^n p_0 - \sum_{k=0}^{n-1}(1+r)^{n-k-1} g_k.$$

Kako je otplata duga najčešće u jednakim mesečnim ratama, npr. $g_n = const = G$ zamenom u poslednjoj jednalini dobijamo

$$p_n = (1+r)^n p_0 - (1+r)^n G \sum_{k=0}^{n-1}(1+r)^{-k-1}$$
$$= (1+r)^n p_0 - [(1+r)^n - 1]\frac{G}{r}$$

Ukoliko želimo da otplatimo dug kroz *n* rata, tada možemo da izračunamo visinu rate, pa se za $p_n = 0$ dobija da je

$$G = p_0 \left[\frac{r}{1-\frac{1}{(1+r)^n}}\right]$$

Izračunajmo, sada, iznos mesečne rate duga

$p_0 = 100 \ evra$

$r = 5\%$

$$G = 100 \left[\frac{0.05}{1-\frac{1}{(1+0.05)^5}}\right] = 23.09748 \ evra \sim 23.10 \ evra$$

Amortizacioni plan:

Mesec	Neplaćeni deo glavnice, mesečno	Anuitet	Kamata od 5%	Otplata glavnice
1	100 evra	23.10 evra	5 evra	18.1 evra
2	81.90	23.10	4.1	19
3	62.90	23.10	3.14	19.96
4	42.94	23.10	2.15	20.95
5	21.99	23.10	1.1	22
6	0			
Ukupno		115.50	15.49	100.01

PRIMER 4

Napraviti plan za otplatu stambenog kredita od 50 000 eur koji želimo da isplatimo nakon 20 godina (30 godina). Kamatna stopa je fiksna i iznosi 5%. Da li će se više para uložiti na kamatu ukoliko je period otplate duga 20 ili 30 godina. Pretpostavimo da nakon 15 godina plaćanja duga možemo da prevremeno otplatimo 10 000 evra duga. Da li nam se prevremana otplata duga više „isplati“ kada je period otplate duga 20 ili kada je period otplate duga 30 god?

Rešenje:

U slučaju perioda otplate duga od 20 godina imamo sledeći amortizacioni plan:

Rata $G_{20} = 4012.129 \sim 4012 eur$ godišnje

Amortizacioni plan bez prevremene otplate:

Godina	Neplaćeni deo glavnice, mesečno	Anuitet	Kamata od 5%	Otplata glavnice
1	50000	4012.13	2500	1512.13
2	48487.9	4012.13	2424.39355	1587.74
3	46900.1	4012.13	2345.006778	1667.12
4	45233	4012.13	2261.650666	1750.48
5	43482.5	4012.13	2174.12675	1838
6	41644.5	4012.13	2082.226637	1929.9
7	39714.6	4012.13	1985.731519	2026.4
8	37688.2	4012.13	1884.411645	2127.72
9	35560.5	4012.13	1778.025777	2234.1
10	33326.4	4012.13	1666.320616	2345.81
11	30980.6	4012.13	1549.030197	2463.1
12	28517.5	4012.13	1425.875257	2586.25
13	25931.3	4012.13	1296.56257	2715.57
14	23215.7	4012.13	1160.784248	2851.34
15	20364.3	4012.13	1018.21701	2993.91
16	17370.4	4012.13	868.521411	3143.61
17	14226.8	4012.13	711.3410315	3300.79
18	10926	4012.13	546.3016331	3465.83
19	7460.21	4012.13	373.0102648	3639.12
20	3821.09	4012.13	191.054328	3821.07
21	0.01189	4012.13	0.000594418	4012.13
Ukupno		80 244.50	30 242.59	50 000.00

Amortizacioni plan sa prevremenom otplatom:

Godina	Neplaćeni deo glavnice, mesečno	Anuitet	Kamata od 5%	Otplata glavnice
1	50000	4012.13	2500	1512.13
2	48487.9	4012.13	2424.39355	1587.74
3	46900.1	4012.13	2345.006778	1667.12
4	45233	4012.13	2261.650666	1750.48
5	43482.5	4012.13	2174.12675	1838
6	41644.5	4012.13	2082.226637	1929.9
7	39714.6	4012.13	1985.731519	2026.4
8	37688.2	4012.13	1884.411645	2127.72
9	35560.5	4012.13	1778.025777	2234.1
10	33326.4	4012.13	1666.320616	2345.81
11	30980.6	4012.13	1549.030197	2463.1
12	28517.5	4012.13	1425.875257	2586.25
13	25931.3	4012.13	1296.56257	2715.57
14	23215.7	4012.13	1160.784248	2851.34
15	20364.3	14012.13	1018.21701	12993.91
16	3726.82	3913.16	186.3410315	3726.82
Ukupno		74 095.11	27 738.70	50 000.00

U slučaju perioda otplate duga od 20 godina imamo sledeći amortizacioni plan:

Rata $G_{30} = 3252.571754 \sim 3252.57 eur$ godišnje

Amortizacioni plan bez prevremene otplate:

Godina	Neplaćeni deo glavnice, mesečno	Anuitet	Kamata od 5%	Otplata glavnice
1	50000	3252.57	2500	752.57
2	49247.43	3252.57	2462.3715	790.1985
3	48457.2315	3252.57	2422.861575	829.708425
4	47627.5231	3252.57	2381.376154	871.193846
5	46756.3292	3252.57	2337.816461	914.753539
6	45841.5757	3252.57	2292.078785	960.491215
7	44881.0845	3252.57	2244.054224	1008.51578
8	43872.5687	3252.57	2193.628435	1058.94157
9	42813.6271	3252.57	2140.681357	1111.88864
10	41701.7385	3252.57	2085.086925	1167.48308
11	40534.2554	3252.57	2026.712771	1225.85723
12	39308.3982	3252.57	1965.419909	1287.15009
13	38021.2481	3252.57	1901.062405	1351.5076
14	36669.7405	3252.57	1833.487025	1419.08298
15	35250.6575	3252.57	1762.532876	1490.03712
16	33760.6204	3252.57	1688.03102	1564.53898
17	32196.0814	3252.57	1609.804071	1642.76593
18	30553.3155	3252.57	1527.665775	1724.90423
19	28828.4113	3252.57	1441.420563	1811.14944
20	27017.2618	3252.57	1350.863091	1901.70691
21	25115.5549	3252.57	1255.777746	1996.79225
22	23118.7627	3252.57	1155.938133	2096.63187
23	21022.1308	3252.57	1051.10654	2201.46346
24	18820.6673	3252.57	941.033367	2311.53663
25	16509.1307	3252.57	825.4565354	2427.11346
26	14082.0172	3252.57	704.1008621	2548.46914
27	11533.5481	3252.57	576.6774052	2675.89259
28	8857.65551	3252.57	442.8827755	2809.68722
29	6047.96829	3252.57	302.3984143	2950.17159
30	3097.7967	3252.57	154.889835	3097.68017
Ukupno		97 577.1	47 420.22	50 000.00

Amortizacioni plan sa prevremenom otplatom:

Godina	Neplaćeni deo glavnice, mesečno	Anuitet	Kamata od 5%	Otplata glavnice
1	50000	3252.57	2500	752.57
2	49247.43	3252.57	2462.3715	790.1985
3	48457.2315	3252.57	2422.861575	829.708425
4	47627.5231	3252.57	2381.376154	871.193846
5	46756.3292	3252.57	2337.816461	914.753539
6	45841.5757	3252.57	2292.078785	960.491215
7	44881.0845	3252.57	2244.054224	1008.51578
8	43872.5687	3252.57	2193.628435	1058.94157
9	42813.6271	3252.57	2140.681357	1111.88864
10	41701.7385	3252.57	2085.086925	1167.48308
11	40534.2554	3252.57	2026.712771	1225.85723
12	39308.3982	3252.57	1965.419909	1287.15009
13	38021.2481	3252.57	1901.062405	1351.5076
14	36669.7405	3252.57	1833.487025	1419.08298
15	35250.6575	13252.57	1762.532876	11490.03712
16	23760.6204	3252.57	1188.03102	2064.53898
17	21696.0814	3252.57	1084.804071	2167.76593
18	19528.3155	3252.57	976.4157746	2276.15423
19	17252.1613	3252.57	862.6080633	2389.96194
20	14862.1993	3252.57	743.1099665	2509.46003
21	12352.7393	3252.57	617.6369648	2634.93304
22	9717.80626	3252.57	485.890313	2766.67969
23	6951.12657	3252.57	347.5563287	2905.01367
24	4046.1129	3252.57	202.3056451	3050.26435
25	995.848547	1045.64	49.79242737	946.057573
Ukupno		89 107.32	39 107.32	49 999.99

POPULACIONI MODELI (OPET)

Posmatrajmo problem u kome određena populacija ima prirodni priraštaj β izražen u procentima. Označimo sa x_n broj jedinki te vrste za n godina. Tada važi

$$x_{n+1} - x_n = \beta x_n, n = 0,1,2, \ldots.$$

Pretpostavimo još da zbog prirodnog izvora hrane, maksimalni broj jedinki ove popupalcije ne može preći broj M. Ako uzmemo da će se u momentu kada broj jedinki ove populacije dostigne broj M, promena broja jedinki biti jednaka nuli, odnosno opadati (jer nema dovoljno hrane). Može se formirati sledeća jednačina

$$x_{n+1} - x_n = \beta x_n \frac{M - x_n}{M}$$

odnosno

$$x_{n+1} = x_n + \beta x_n \frac{M - x_n}{M}.$$

PRIMER 5:

Pretpostavimo da je u nekom regionu prebrojano 100 sova i da je prirodni priraštaj sova u tom regionu 4%. Pretpostavimo još da je zbog ograničenja u zalihama hrane maksimalni broj sova u tom regionu 500. Ako sa x_n označimo broj sova za n godina, tada se korišćenjem modela iz prethodnog zadatka može izvesti sledeća diferencna jednačina:

$$x_{n+1} = x_n + \frac{0.04}{500} x_n(500 - x_n) = x_n + 0.00008(500 - x_n)x_n, \quad n = 0,1,2, \ldots.$$

Korišćenjem dobijene iterativne formule, možemo odrediti broj sova u narednih 10 godina

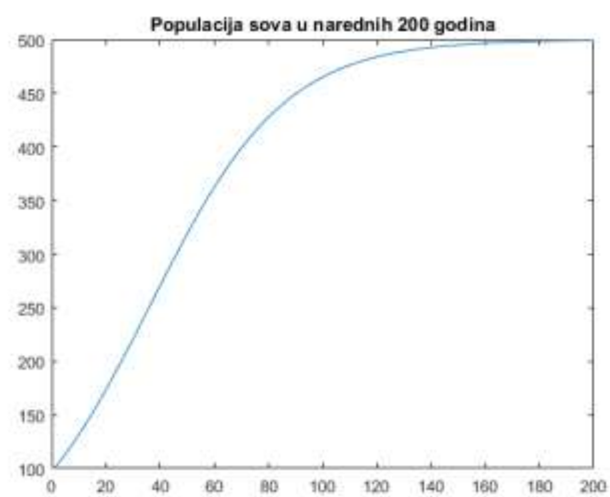
godina	0	1	2	3	4	5	6	7	8	9	10
populacija	100	103.2	106.476	109.828	113.2562	116.7603	120.3401	123.9951	127.7249	131.5289	135.406

Prikazaćemo grafički kako se broj jedinki ove vrste menja i koliko je vremena potrebno pre nego što se dostigne maksimalni broj jedinki.

Primera radi, možemo da odredimo koliko će sova biti za 150 godina.

```
x(1) = 100;
for i = 2:150
    x(i) = x(i-1)+0.00008.*(500-x(i-1)).*…
            x(i-1);
end
plot(x);
title('Populacija sova u narednih  godina');
```

>> x(150)
 495.1708



PRIMER 6:

Vesti se šire gradom tako da je brzina širenja vesti proporcionalna broju stanovnika koji su čuli vest y i međusobnog kontakta sa delom stanovništva koji nisu čuli vest. Ako gradić ima 1000 stanovnika i ako je u 8 sati ujutru vest čulo 80 ljudi, a do podne je čulo pola grada

a) Napisati diferencijalnu jednačinu koja zadovoljava uslove zadatka i rešiti je
b) U kom trenutku će 90% stanovništva ovog gradića čuti vest?

Rešenje:

a) Važi sledeća jednačina

$$y' = ky(1000 - y)$$

Rešavanjem diferencijalne jednačine dobija se da je

$$\frac{dy}{y(1000-y)} = kdt$$

$$\frac{1}{1000}\left(\frac{1}{y} + \frac{1}{1000-y}\right) = kdt$$

Integraljenjem jednačine dalje se dobija da je

$$\frac{1}{1000}\ln\frac{y}{1000-y} = kt + \ln C$$

$$\ln\frac{y}{1000-y} = 1000kt + 1000\ln C$$

$$\frac{y}{1000-y} = Ae^{1000kt}$$

b) Zamenom početnih uslova

$$y(0) = 80, \qquad y(4) = 500, \quad y(t_0) = 900$$

dobija se da je $A = 0.08696$, $k = 0.00061$ odnosno da će 90% stanovništva čuti vest za $t_0 = 7.6$h sati
Dakle, 90% stanovništva će čuti za vest do 8h + 7.6 = 15h i 36min.

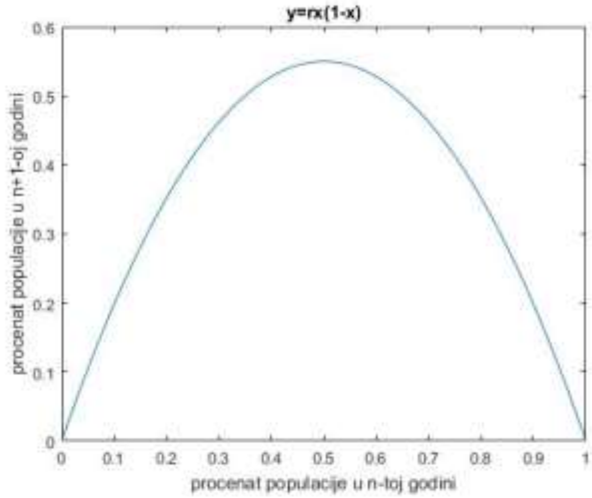
Promene koje se tiču broja jedinki neke populacije se najjednostavnije mogu prikazati korišćenjem jednačine

$$x_{n+1} = rx_n(1 - x_n) \qquad (*)$$

koja je u literaturi poznata pod nazivom logistička jednačina.
Ovakvu jednačinu su ekolozi koristili sredinom prošlog veka kako bi opisali promenu broja jedinki određenih populacija. Ako se oslobodimo zagrada, dobićemo izraz $x_{n+1} = rx_n - rx_n^2$.
Nacrtajmo grafik jednačine $y = rx(1-x)$ za $r = 2.2$. Primetićemo da jednačina dostiže maksimum nakon čega opada. Dakle, nakon što se dostigne određeni maksimalni procenat priraštaja broja jedinki te vrste, procenat priraštaja broja jedinki će opadati kao posledica ograničene količine hrane, tj. broj jedinki te vrste će biti "konstantan".

Uzmimo da je $r = 2.2$ i $x_0 = 0.03$, zamenom u logističku jednačinu dobija se sledeći niz vrednosti:

$x_1 = 0.064$
$x_2 = 0.1318$
$x_3 = 0.2518$
$x_4 = 0.4145$
$x_5 = 0.5339$
$x_6 = 0.5475$
$x_7 = 0.5450$
$x_8 = 0.5455$
$x_9 = 0.5454$
$x_{10} = 0.5455$



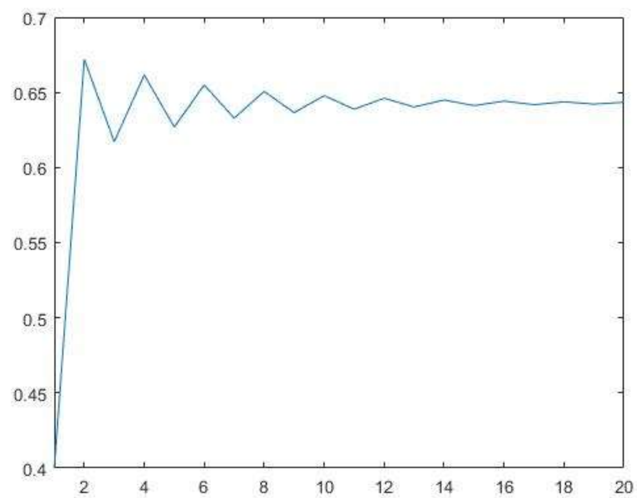
Vidimo da se vrednost elemenata niza prati nagli porast nakon čega blago varira oko vrednosti 0.54. Lako se možemo uveriti da bi se ovakva stabilizacija dogodila kada bi za početnu vrednost uzeli 0.5 ili 0.9. Ovakav konretan model opisuje čestu pojavu u prirodi, a to je da na određenom području obitava stalan broj jedinki neke vrste (ukoliko nema nekih značajnijih spoljnih uticaja).

Primer:

Uzmimo da je $r = 2.8$ i da je $x_0 = 0.4$. Recimo da želimo da odredimo broj populacija za 20 godina.

```
function x = logisticka(r,x0,n)
x(1)= x0;
for i =1:n-1
    x(i+1) = r.*x(i).*(1-x(i));
end
plot(x);
axis([1 20 0.4 0.7]);
end
```

>> x = logisticka(2.8,0.4,20);

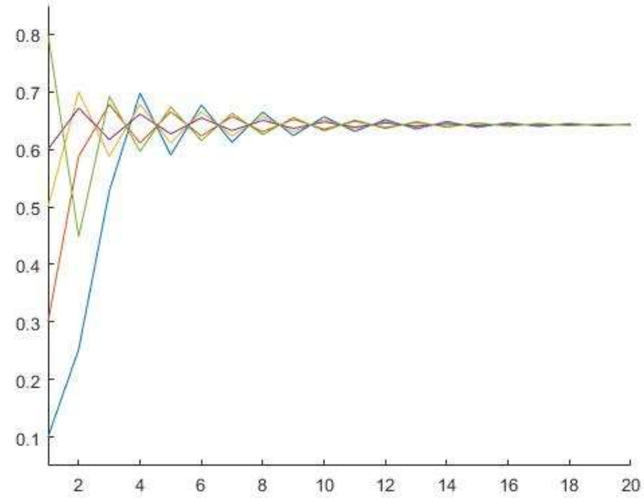


>> [ind x']

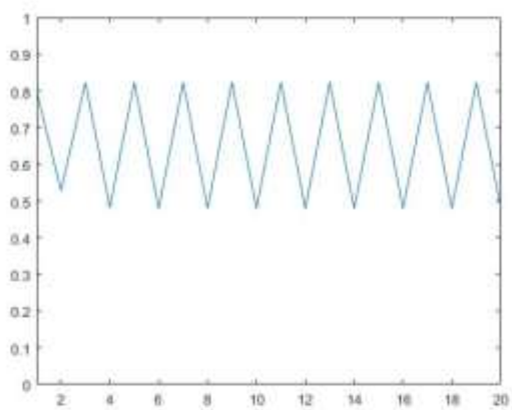
1.0000	0.4000	8.0000	0.6506	15.0000	0.6412
2.0000	0.6720	9.0000	0.6365	16.0000	0.6442
3.0000	0.6172	10.0000	0.6478	17.0000	0.6418
4.0000	0.6616	11.0000	0.6388	18.0000	0.6437
5.0000	0.6269	12.0000	0.6461	19.0000	0.6422
6.0000	0.6549	13.0000	0.6403	20.0000	0.643
7.0000	0.6328	14.0000	0.6449		

Uzmimo sada da je $r = 2.8$ i nekoliko različitih početnih vrednosti, na primer: $x_1 = 0.1$, $x_1 = 0.3$, $x_1 = 0.5$, $x_1 = 0.6$ i $x_1 = 0.8$. Primetićemo da za različite početne vrednosti niz teži ka istoj graničnoj vrednosti (promenićemo red koji se tiče granica osa, zbog lepšeg prikaza `axis([1 20 0.05 0.85]);`)

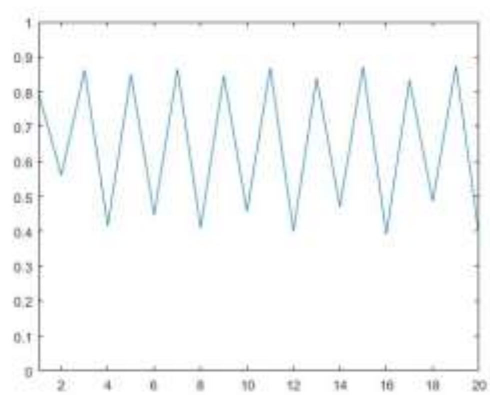
```
hold on
logisticka(2.8,0.1,20);
logisticka(2.8,0.3,20);
logisticka(2.8,0.5,20);
logisticka(2.8,0.6,20);
logisticka(2.8,0.8,20);
hold off
```



- Dakle, za velike vrednosti broja n rešenja logističke jednačine teže ka istoj vrednosti nezavisno od početne vrednosti.
- Računski rezultati su pokazali da je za $r < 3$ rešenje x_n logističke jednačine (*) stabilno, tj. da oscilira oko jedne vrednosti.
- Međutim, za $r > 3$ i $r < 3.5$ rešenje x_n oscilira između dve vrednosti (ponovo nezavisno od početnih vrednosti), videti Sliku 1 ($r = 3.3$ i $80 \leq n \leq 100$). Period oscilacija je $T = 2$.

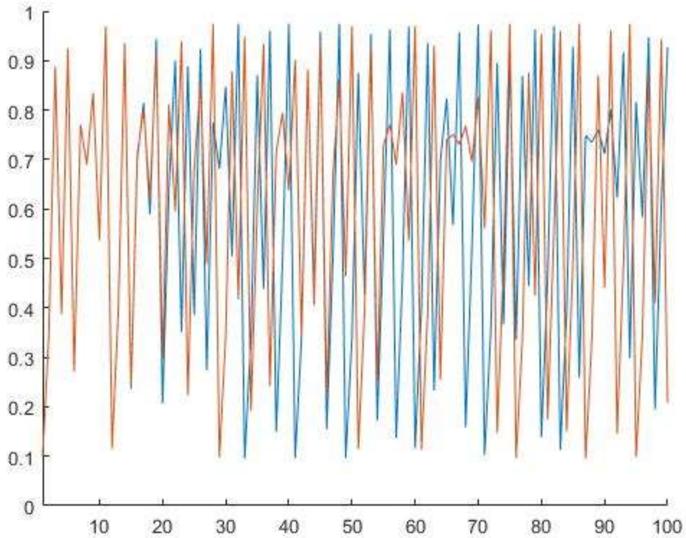


Slika 1.



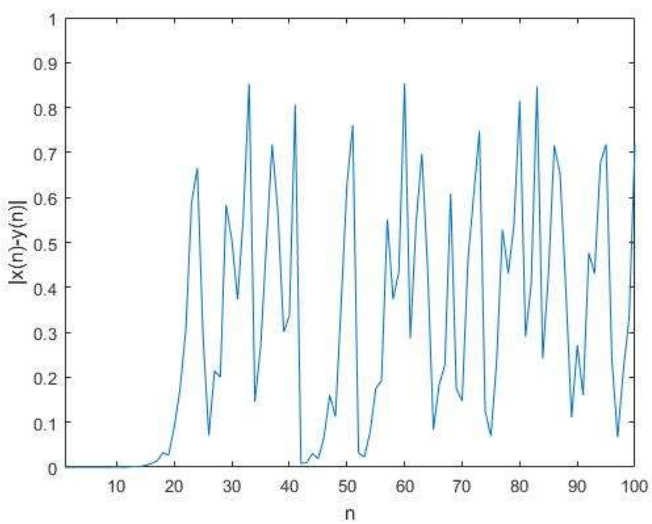
Slika 2.

- Slično, za $r = 3.5$ se dobija da je period oscilovanja $T = 4$ (Slika 2)
- Na početku se verovalo da se ovakve oscilacije dešavaju kao statističke greške. Međutim, ne radi se ni o kakvoj grešci, već o sastavu koji opisuje deterministički haos, tj. o modelu koji za male promene početnih uslova uzrokuje drstične promene u rezultatima. U literaturi se spominje čuveni leptirov efekat koji kaže da zamah krilima leptira u Pekingu izaziva oluju u Parizu.
- Sa tim u vezi, ako nastavimo da uvećavamo vrednost parametra r primetićemo da se period oscilovanja menja, i da za $r = 3.569946$ period T teži ka beskonačnosti (nastaje haos).
- Na slici 3 je prikazano rešenje za $r = 3.9$ i za dva početna uslova $x_1 = 0.1$ i $x_1 = 0.100001$. Primetimo da se rešenje ne može razlikovati do $n = 20$, dok su nakon $n = 25$ razlike drastične.



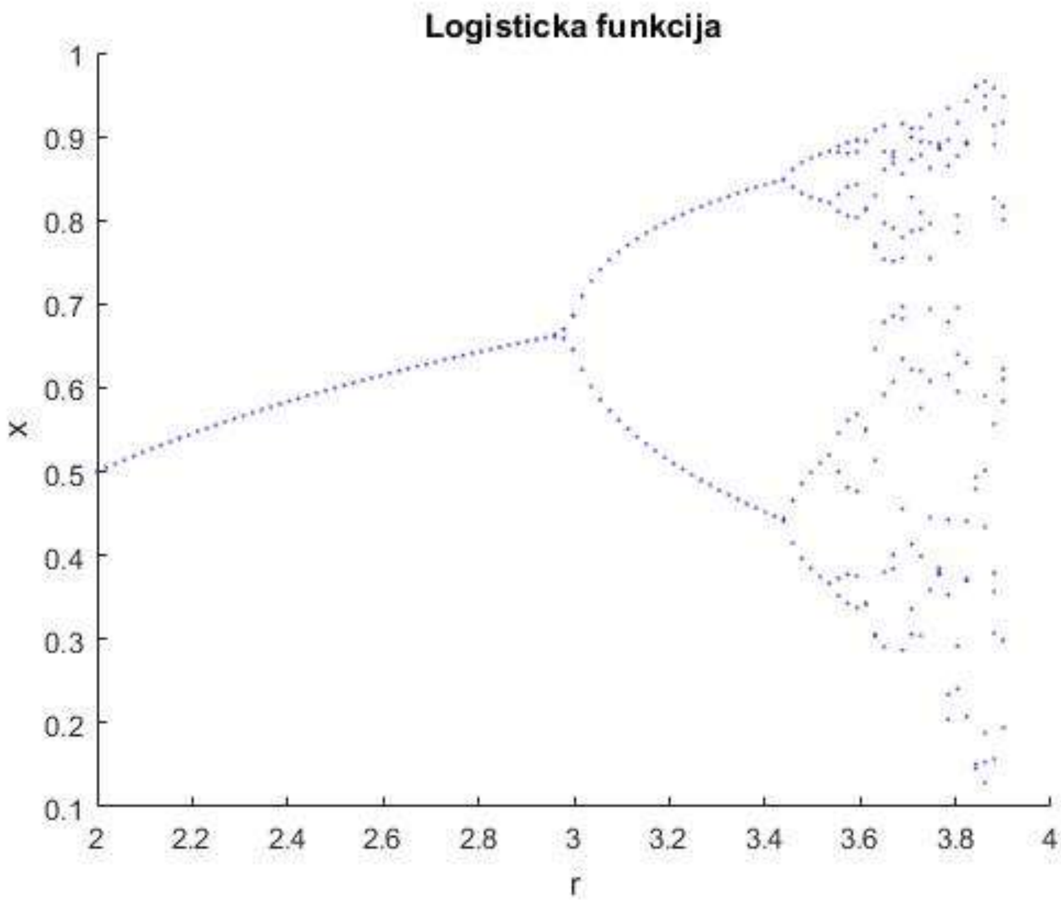
Slika 3.

Nacrtaćemo i grafik funkcije $d = |x_n - y_n|$ kako bi se ove razlike jasnije videle (Slika 4).



Slika 4.

Nacrtajmo sada rešenje logističke jednačine u zavisnosti od vrednosti parametra $r = [2,3.9]$ ako je $x(1) = 0.1$ i $n = 100$.



haos.m

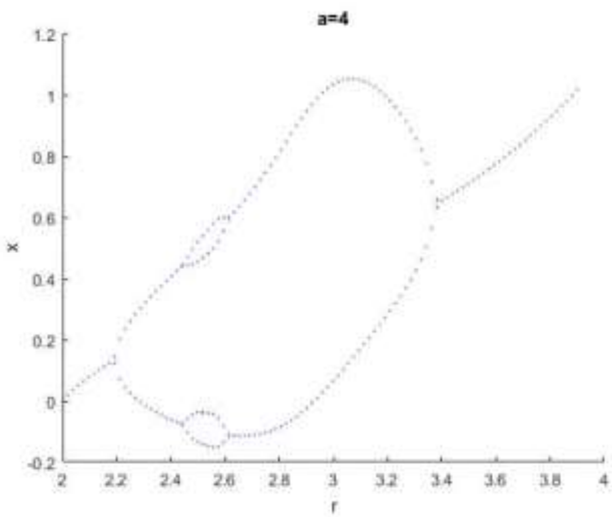
```
n = 100;
r = linspace(2,3.9,100);
hold on
for k = 1: length(r)
    x(1) = 0.1;
    for j = 1:n-1
        x(j+1) = r(k).*x(j).*(1-x(j));
        if (j>0.9*n)
            plot(r(k),x(j),'b.','MarkerSize', 1);
        end
    end
end
title('Logisticka funkcija');
xlabel('r');
ylabel('x');
```

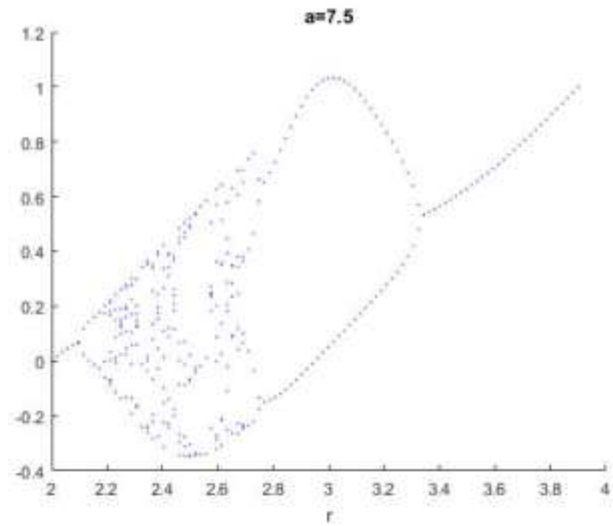
Postupak udvajanja tačaka nagomilavanja niza se naziva *bifurkacija*.

Zadatak

Napraviti bifurkacione dijagrame za sledeća preslikavanja

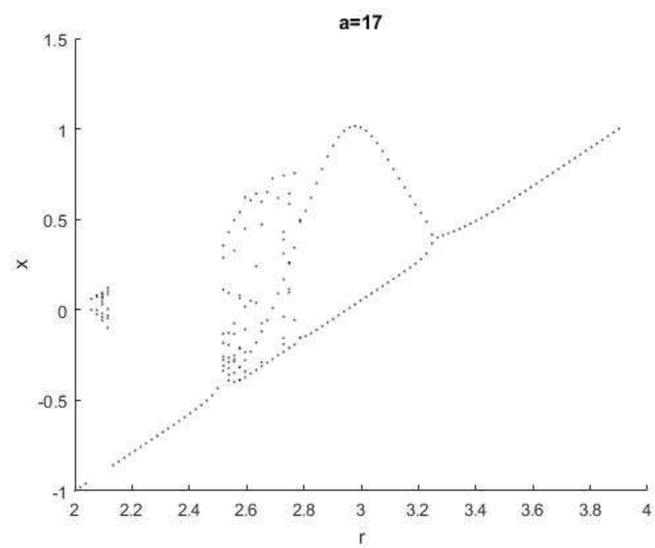
1. $x_{n+1} = b + e^{-ax_n^2}$ ako je $a = 4$, $a = 7.5$ i $a = 17$.
   Dijagrame prikazati za $-1 \leq b \leq 1$.
2. $x_{n+1} = b - x_n(1 - x_n) + e^{-ax_n^2}$ ako je $a = 7.5$, $-2 \leq b \leq 0.7$.





Zad1.m

```
n = 100;
a = 17;
b = linspace(-1,1,100);
hold on
for k = 1: length(r)
    x(1) = 0.1;
    for j = 1:n-1
        x(j+1) = b(k) + exp(-a.*x(j).^2);
        if (j>0.9*n)
            plot(r(k),x(j),'k.','MarkerSize', 1);
        end
    end
end
title('a=17');
xlabel('r');
ylabel('x');
```



Zadatak 2 se slično rešava.

NELINEARNE DIFERENCNE JEDNAČINE DRUGOG REDA

Rešavamo diferencnu jednačinu drugog reda

$$x_{n+1} = bx_{n-1} + 1 - ax_n^2$$
$$x(0) = x_0, x(1) = x_1$$

Neka su $a = 1.4$ i $b = 0.3$. Uz početni uslov $x_0 = x_1 = 0$ primetićemo da se niz x_n ponaša haotično.
Ako uvedemo smenu

$$y_n = bx_{n-1}$$

posmatrani problem se svodi na sistem dve diferencne jednačine prvog reda

$$x_{n+1} = y_n + 1 - ax_n^2$$
$$y_n = bx_{n-1}$$

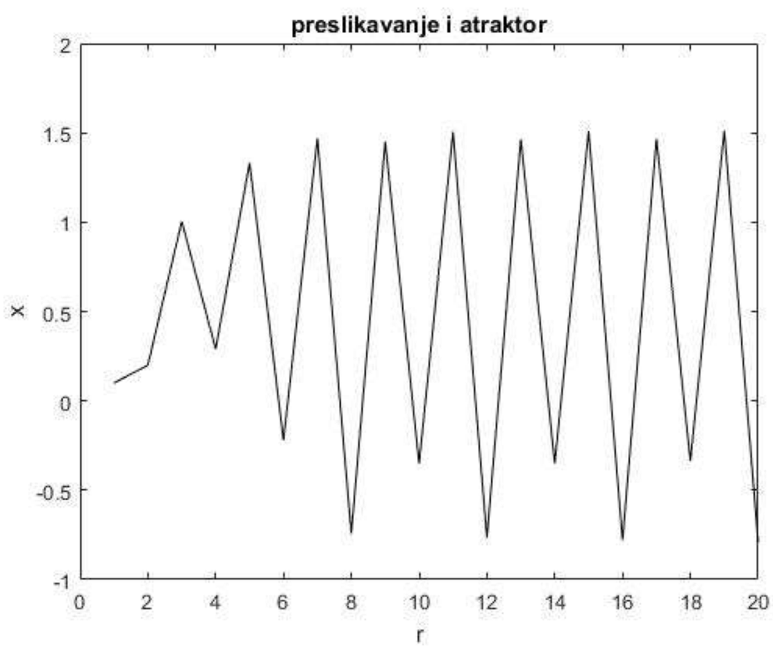
sa početnim uslovima $x_1 = 0, y_1 = bx_0 = 0$.

- Nacrtati atraktor i nekoliko prvih preslikavanja za

$$x_{n+1} = bx_{n-1} + 1 - ax_n^2 e^{-ax_{n-1}^2}$$

za $a = 0.8, b = 0.36$.

```
clear;
n = 20;
a = 0.8;
b = 0.36;
x(1) = 0.1;
x(2) = 0.2;
for j = 2:n-1
    x(j+1) = b.*x(j-1)+1-a.*x(j).^2.*exp(-a.*x(j-1).^2);
end
plot(x,'k');
title('preslikavanje i atraktor');
xlabel('r');
ylabel('x');
```



FRAKTALI

Fraktali su geometrijski likovi koji liče na sebe koliko god ih uvećavali.
Fraktalne krive najčešće nemaju konačnu dužinu.
Krajem 19og i početkom 20og veka, fraktali se predstavljaju kao nova grana matematike. Oni daju opis mnogih prirodnih oblika i prožimaju se kroz mnoge oblasti nauke (kompjuterska grafika, biologija, medicina).
Fraktali imaju dve osnovne osobine:

- Samosličnost
    - Potpuno slični fraktali (oni koji sadrže kobije koje su slične celom fraktalu) su Kohova pahulja, trougao Sierpinskog i tako dalje.
    - Kvazi samoslični fraktali su oni koji sadrže male kopije sebe koje nisu slične ccelom fraktalu. Na primer Mandelbrotov skup, Julijin skup i tako dalje
    - Statistički samoslični fraktali ne sadrže kopije samog sebe, ali neke njegove osobine (fraktalna dimenzija) ostaju iste pri različitim procenama (Peanov šum na primer)
- Fraktalnu dimenziju koja nije ceo broj

Kohova pahulja

Kohova pahulja predstavlja jedan od prvih i najpoznatijih fraktala. Kod Kohove krive se kreće od duži, dok se kod Kohove pahulje polazi od trougla. Kod Kohove krive u svakoj iteraciji se njena dužina povećava za faktor 4/3.

```
function Kohova_pahulja(Iter)

if nargin ~= 1
    error('Nema dovoljno ulaznih podataka');
end

xl = zeros(10,1);
xr = xl;
yl = xl;
yr = yl;
xr(Iter) = 1;
r = sqrt(1/3^2 - 1/6^2);
clf;
% podesavanje prikaza ...
set(gca,'FontSize',14);
set(gcf,'Color',[1,1,1]);
hold on;
pahuljica(xl,xr,yl,yr,Iter,r);
title('Kohova pahulja');
text(0.5,-0.05,(['Broj iteracija: ' num2str(Iter)]), ...
         'HorizontalAlign','center','FontSize',12);
hold off;
axis equal; axis tight; axis off;

%--------------------------------------------------------------
```

```
function pahuljica(xl,xr,yl,yr,Iter,r)

if (Iter < 2)
    plot([xl(1) xr(1)],[-yl(1) -yr(1)],'b-')
    return
end

Iter = Iter-1;

xl(Iter) = xl(Iter+1);
yl(Iter) = yl(Iter+1);
xr(Iter) = 1/3 * xr(Iter+1) + 2/3 * xl(Iter+1);
yr(Iter) = 1/3 * yr(Iter+1) + 2/3 * yl(Iter+1);
pahuljica(xl,xr,yl,yr,Iter,r);

xl(Iter) = xr(Iter);
yl(Iter) = yr(Iter);
xr(Iter) = .5 * xr(Iter+1) + .5 * xl(Iter+1) - r * (yl(Iter+1) - yr(Iter + 1));
yr(Iter) = .5 * yr(Iter+1)+.5*yl(Iter+1)+r*(xl(Iter+1)-xr(Iter+1));
pahuljica(xl,xr,yl,yr,Iter,r);

xl(Iter) = xr(Iter);
yl(Iter) = yr(Iter);
xr(Iter) = 2/3 * xr(Iter+1) + 1/3 * xl(Iter+1);
yr(Iter) = 2/3 * yr(Iter+1) + 1/3 * yl(Iter+1);
pahuljica(xl,xr,yl,yr,Iter,r);

xl(Iter) = xr(Iter);
yl(Iter) = yr(Iter);
xr(Iter) = xr(Iter+1);
yr(Iter) = yr(Iter+1);
pahuljica(xl,xr,yl,yr,Iter,r);
Iter = Iter + 1;
return;
```

Mandelbrotov skup

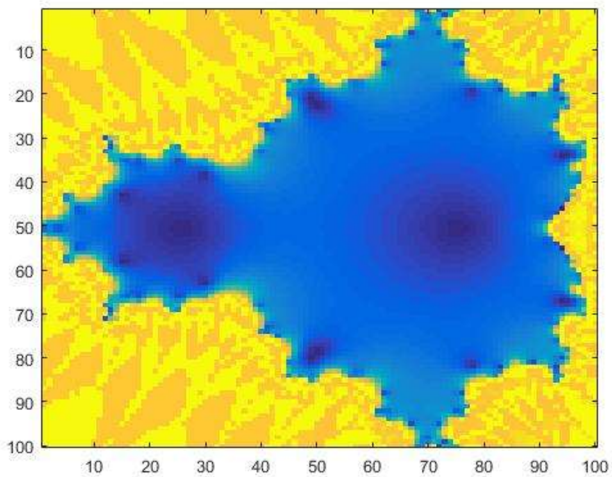
Mandelbrotov skup predstavlja najsavršeniji od svih fraktala
Ima svojstvo samosličnosti
Predstavlja skup tačaka u kompleksnoj ravni pri čemu fraktal formira granicu tog skupa.
Za formiranje Mandelbrotovog skupa se koristi algoritam zasnovan na rekurentnj formuli

$$z_{n+1} = z_n^2 + C$$

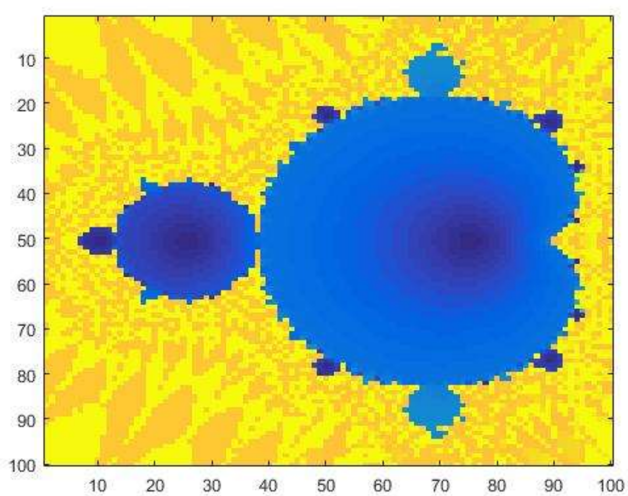
gde je $C = a + ib$ kompleksan broj i $z_0 = 0$, koja razdvaja dve tačke kompleksne ravni u dve kategorije:

- Tačke unutar Mandelbrotovog skupa
- Tačke izvan Mandelbrotovog skupa

U zavisnosti od broja tačaka Mandelbrotovi skupovi imaju sledeće oblike:
>> Mandelbrotov_skup(100,10)



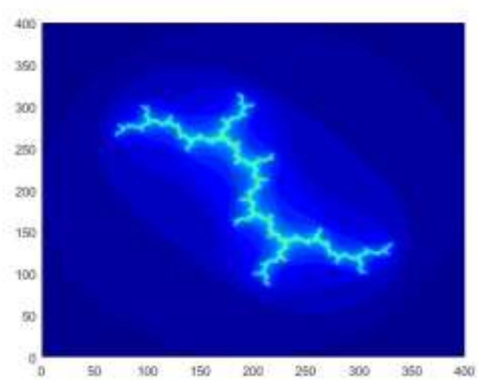
>> Mandelbrotov_skup(100,10)



Moguća su dva ishoda, ili će niz z_n ostati ograničen, ili će težiti bekonačnosti.

```
function Mandelbrotov_skup(brTc, brIt)

% brIt - broj iteracija u Mandelbrotovom skupu
% brTc - broj tacaka unutar slike
% test:
%           Mandelbrotov_skup(1000,51)
% Generisemo z = 0 sa njegovim realnim i imaginarnim delom
% z(n+1) = z(n)^2 + c

clf;
zRe = zeros(brTc,brTc);
zIm = zeros(brTc,brTc);

% Generisemo konstantu c sa njenim realnim i imaginarnim delom
cRe = repmat(linspace(-1.5,0.5,brTc),brTc,1);
cIm = repmat(linspace(-1,1,brTc)',1,brTc);

% Iterativna formula
for j = 1 : brIt
    % Racunamo q = z*z + c u kompleksnom prosotoru
    % q je konanca promenljiva u koju pakujemo rezultate
    qRe = zRe .* zRe - zIm .* zIm + cRe;
    qIm = 2 .* zRe .* zIm + cIm;
    % Da bismo izbegli numericku divergenciju ogranicicemo  q na [-5 5]
    zRe = qRe;
    q_veci_od_pet =  find(qRe > 5.);
    zRe(q_veci_od_pet) = 5.;
    q_manji_od_pet = find(qRe < -5.);
    zRe(q_manji_od_pet) = -5.;

    zIm = qIm;
    z_veci_od_pet = find(qIm > 5.);
    zIm(z_veci_od_pet) = 5.;
    z_manji_od_pet = find(qIm < -5.);
    zIm(z_manji_od_pet) = -5.;
end

% Graficki prikazujemo
slika = log( sqrt(zRe.*zRe+zIm.*zIm) + 1);
imagesc(slika);                          % crta sliku u bojama
```

Julija skupovi



Julija skupovi su definisani u kompleksnoj ravni slično kao i Mandelbrotov skup.
Posmatrajmo niz

$$z_{n+1} = z_n^2 + C$$

gde je $C = a + ib$ kompleksan broj i $z_0 = 0$.
Ako fiksiramo konstantu C i variramo kompleksnu iteraciju z_0 po istom kriterijumu kao za Mandelbratov skup, dobićemo Julia skupove za ovo preslikavanje.

```
function Julijini_skupovi(c,k,Xr,Yr)
%
% Crtamo Julijine skupove za dato c
%
% c - parametar jednacine z = z^2 + c
% k - broj iteracija
% Xr - funkcija se crta na intervalu [Xr(1,1) Xr(1,2)]
% Yr - funkcija se crta na intervalu [Yr(1,1) Yr(1,2)]
%
% test:
% Julijini_skupovi(i,100,[-2 2],[-2 2])
clf;

n = 400;
x = linspace(Xr(1,1),Xr(1,2),n);
y = linspace(Yr(1,1),Yr(1,2),n);

% pravimo mrezu po kojoj ce se crtati skup
[X,Y] = meshgrid(x,y);

% M je matrica sa podacima iz Julijinih skupova
M = zeros(length(X),length(Y));

for l = 1 : size(X,2)
    for j = 1 : size(Y,2)
       [w Iter] = Julia(X(l,j)+Y(l,j)*i , c, k);
       M(l,j) = M(l,j) + Iter;
    end
end
hold on;
%axis off;

colormap(jet);
pcolor(M);
shading interp;

hold off;
end


%*********************%
function [pri,it] = Julia(z,c,k)

R = max(abs(c),2);
i = 0;
while i < k
    if abs(z) > R
        pri = 1;
        it = i;
        return;
    end
    z = z^2 + c;
    i = i + 1;
end
pri = 0;
it = i;
end
```

ITERACIJE SISTEMA FUNKCIJA

Posmatramo preslikavanja $F_1(x,y), F_2(x,y), \dots, F_k(x,y)$ zajedno sa verovatnoćama $p_1, p_2, \dots, p_k$ čija je suma 1 i neka je poznato (x_0, y_0). Ako je (x_n, y_n) već sračunato, tada na slučajan način biramo jedno do mogućih k preslikavanja koje primenjujemo za dobijanje sledeće iteracije.
Posmatramo sledeća preslikavanja:

TROUGAO SIERPINSKOG

Spajamo središte svake stranice jednakostraničnog trougla.

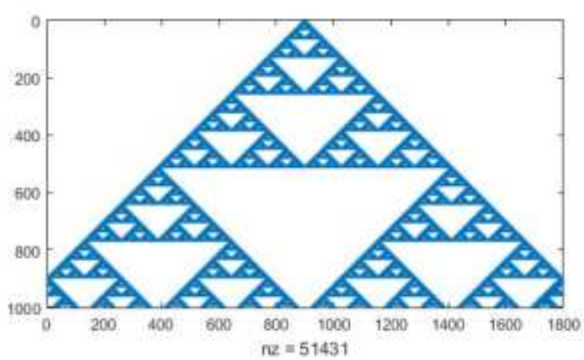
$$F_1(x,y) = \left(\frac{x}{2}, \frac{y}{2}\right)$$

$$F_2(x,y) = \left(\frac{2x+1}{4}, \frac{y+1}{2}\right)$$

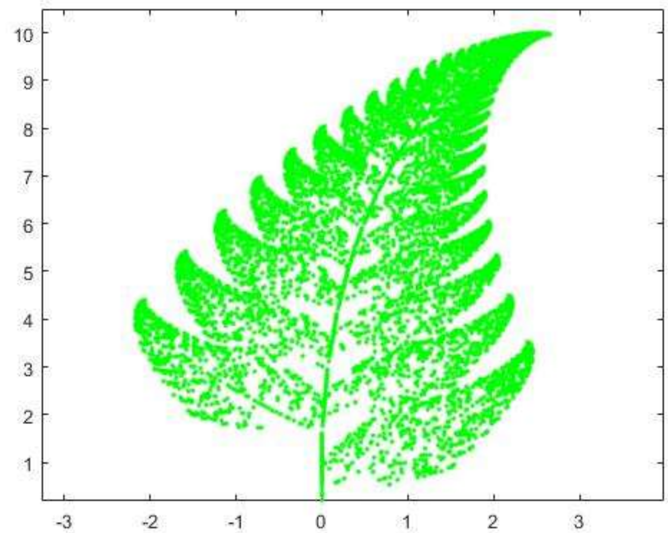
$$F_3(x,y) = \left(\frac{x+1}{2}, \frac{y}{2}\right)$$

Uzimajući da je $p_1 = p_2 = p_3 = 1/3$ i startujući iz tačke $(1,1)$ dobija se trougao Sierpisnkog
Tepih Sierpinskog je sličan, počinje se od kvadrata.

```
function sierpinski(duzina, visina)
% test:  sierpinski(1001,1801)
a = zeros(duzina,visina); % Sirina i visina grafika
a(1,(visina-1)/2) = 1;
for i = 2 : duzina
    for j = 2 : visina-1
        if a(i-1,j-1) == 0 && a(i-1,j) == 0 && a(i-1,j+1) == 1
            a(i,j) = 1;
        else
            if a(i-1,j-1) == 1 && a(i-1,j) == 0 && a(i-1,j+1) == 0
                a(i,j) = 1;
            end
        end
    end
end
spy(a); % crta retku matricu
end
```



BARNSLIJEVA PAPRAT



```
NumOfPts = 10000;
iter = 50;
pts = zeros(NumOfPts,2);
for j = 1:NumOfPts
    x = rand(1);
    y = rand(1);
    for i = 1:iter
        p = rand(1);
        if p < 0.01
            xn = 0;
            yn = .16*y;
            x = xn;
            y = yn;
        elseif p < .08
            xn = .2*x-.26*y;
            yn = .23*x+.22*y+1.6;
            x = xn;
            y = yn;
        elseif p < .15
            xn = -.15*x+.28*y;
            yn = .26*x+.24*y+.44;
            x = xn;
            y = yn;
        else
            xn = .85*x+.04*y;
            yn = -.04*x+.85*y+1.6;
            x = xn;
            y = yn;
        end
    end%i
    pts(j,1) = x;
    pts(j,2) = y;
end%j
xs = pts(:,1);
ys = pts(:,2);
plot(xs,ys,'.g')
axis([min(xs)*1.5,max(xs)*1.5,min(ys)*1.05,max(ys)*1.05]);
```

IMPRESIONISTIČKO DRVO

```
NumOfPts = 10000;
iter = 50;
pts = zeros(NumOfPts,2);
for j = 1:NumOfPts
    x = rand(1);
    y = rand(1);
    for i = 1:iter
        p = [0.11 0.09 0.18 0.19 0.21 0.22];
        izbor = round(10*rand());
        while ((izbor > 6) || (izbor == 0))
            izbor = round(10*rand());
        end
        izbor = p(izbor);
        if izbor == 0.11
            xn = 0.5*x;
            yn = 0.6*y;
            x = xn;
            y = yn;
        elseif izbor == 0.09
            xn = 0.05*x;
            yn = -0.5*y;
            x = xn;
            y = yn;
        elseif izbor == 0.18
            xn = 0.46*x - 0.15*y;
            yn = 0.39*x + 0.38*y + 0.6;
            x = xn;
            y = yn;
        elseif izbor == 0.19
            xn = 0.47*x - 0.15*y;
            yn = 0.17*x + 0.42*y + 1.1;
            x = xn;
            y = yn;
        elseif izbor == 0.21
            xn = 0.43*x  + 0.28*y;
            yn = -0.25*x + 0.45*y + 1;
            x = xn;
            y = yn;
        elseif izbor == 0.22
            xn = 0.42*x  + 0.26*y;
            yn = -0.35*x + 0.31*y + 0.7;
            x = xn;
            y = yn;
        end
end%i
    pts(j,1) = x;
    pts(j,2) = y;
end%j

xs = pts(:,1);
ys = pts(:,2);

plot(xs,ys,'.g')
axis([min(xs)*1.5,max(xs)*1.5,min(ys)*1.05,max(ys)*1.05]);
```