

PRESEK

List za mlade matematike, fizike, astronome in računalnikarje

ISSN 0351-6652

Letnik 26 (1998/1999)

Številka 5

Strani 274-278

Matija Lokar:

KOCHOVA SNEŽINKA

Ključne besede: računalništvo, programska oprema, simbolno računanje, DERIVE, Kochova krivulja.

Elektronska verzija: <http://www.presek.si/26/1381-Lokar.pdf>

© 1999 Društvo matematikov, fizikov in astronomov Slovenije

© 2010 DMFA - založništvo

Vse pravice pridržane. Razmnoževanje ali reproduciranje celote ali posameznih delov brez poprejšnjega dovoljenja založnika ni dovoljeno.

KOCHOVA SNEŽINKA

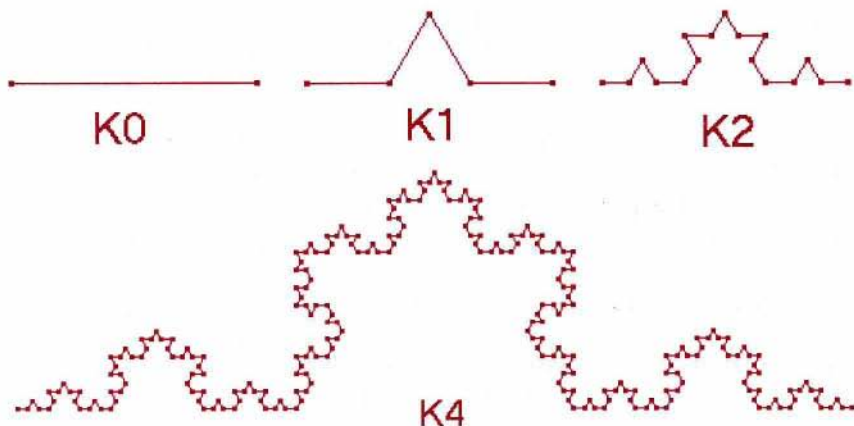
Zima se je poslovila in snežinke bodo kmalu le še oddaljeni spomin. Da ne bo prehod v poletno soparo prehiter, si z računalnikom pričarajmo še malo zimskega vzdušja in narišimo snežinko ali dve.

Naše snežinke bodo seveda nekoliko matematično obarvane. Za osnovo bomo vzeli t.i. Kochovo krivuljo. Ta spada med rekurzivne objekte. Ti nastanejo tako, da z določenim postopkom preoblikujemo osnovni objekt v več enakih objektov. Osnovni objekt je lahko trikotnik, črta, kocka ali kaj drugega. Dobljene objekte z enakim postopkom preoblikujemo naprej. Tako poznamo Peanovo krivuljo, zmajnico, Hilbertovo krivuljo, trikotnik Sierpinskega. Več o teh krivuljah si lahko preberete v članku Cirila Pezdirja, *Nenavadne krivulje*, Presek 18 (1990/91), str. 56–64.

Kochova krivulja

Kochovo krivuljo dobimo tako, da ravno črto (Kochovo krivuljo ničtega reda) razdelimo na tretjine. Srednji del izrežemo in ga nadomestimo z dvema črtama, ki bi skupaj z zbranim delom tvorili enakostranični trikotnik. Dobimo Kochovo krivuljo prvega reda.

Če postopek ponovimo na vseh štirih dobljenih črtah, nastane Kochova krivulja drugega reda. Od tod do Kochove snežinke ni več daleč. Vzamemo enakostranični trikotnik in mu namesto stranic narišemo Kochove krivulje ustreznega reda.



Za risanje rekurzivnih objektov je zelo primeren programski jezik logo. Če ste tega jezika vsaj malo vešč, bo snežinka kaj hitro pred vami. Zanimiv članek o risanju Kochove snežinke najdete tudi na Internetu

na naslovu <http://www.eurologo.org/papers/logomov.html>. A v logu smo v Preseku že veliko programirali. Zato si bomo tokrat ogledali, kako bi snežinko narisali s pomočjo programa DERIVE. Ta program je sicer namenjen predvsem simbolnemu računanju. Ker pa zna narisati točke in v njem lahko definiramo tudi rekurzivne funkcije, bo za našo nalogo povsem primeren.

Risanje točk s programom DERIVE

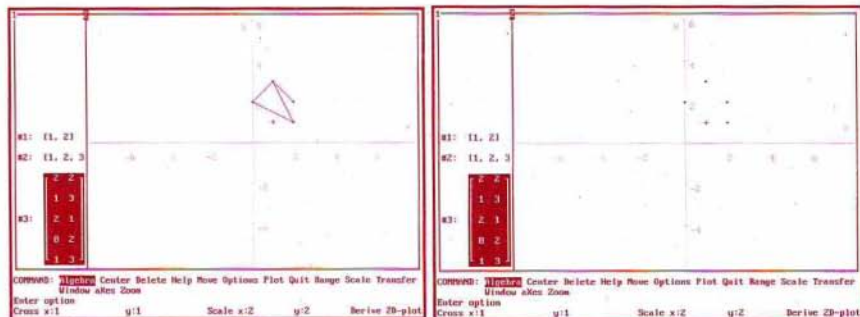
Točko v programu DERIVE podamo kot par koordinat v oglatih oklepajih, pri čemer sta koordinati ločeni z vejico. Tako je koordinatno izhodišče točka $[0,0]$, točko s koordinato $x = 1$ in koordinato $y = 2$ pa v programu DERIVE predstavimo kot $[1,2]$. Če tak izraz (npr. $[1,2]$) narišemo, dobimo narisano točko $(1,2)$.

Če želimo narisati več točk, sestavimo vektor, katerega komponente so spet dvodimenzionalni vektorji – torej matriko velikosti $n \times 2$, kjer je n število točk. Kadar imamo več točk, se lahko odločimo, ali bomo pri risanju točke med sabo povezali ali ne. Ko smo v oknu za risanje (ukaz Plot) z izbiro Options/State, nastavimo Mode v Connected ali pa v Discrete.

```

OPTIONS STATE: Coordinates:(Rectangular)Polar Mode: Connected Discrete
                Size:(Large)Small Follow:(Yes)No Trace: Yes(No)
Select plot point mode
Cross x:1      y:1      Scale x:2      y:2      Derive 2D-plot
  
```

Prva nastavev pomeni, da DERIVE takrat, ko riše več točk, te med sabo poveže, druga nastavev pa, da črt med zaporednimi točkami ne bo.



Če narišemo matriko velikosti 2×2 , dobimo daljico. Začetek daljice določa prva, konec pa druga vrstica matrike.

Kochova krivulja s programom DERIVE

Izračunajmo tri nove točke, ki nastanejo po prvem koraku. Definirajmo tri funkcije. Te vrnejo točke, ki skupaj s krajišči prvotne črte določajo novo črto. Parametra vseh treh funkcij sta začetna in končna točka. Vnesimo z ukazom Author vse te tri funkcije. Izrazi z lihimi številkami so komentarji, ki jih lahko izpustimo.

```
#1: "Točka na tretjini med A in B."
```

```
#2: TT(a, b) := a +  $\frac{b - a}{3}$ 
```

```
#3: "Točka na dveh tretjinah med A in B."
```

```
#4: TDT(a, b) := TT(b, a)
```

```
#5: "Točka na osti."
```

```
#6: TO(a, b) :=  $\frac{a + b}{2} + \frac{\sqrt{3}}{6} \begin{bmatrix} a - b & b - a \\ 2 & 1 & 1 \end{bmatrix}$ 
```

```
COMMAND: Author Build Calculus Declare Expand Factor Help Jump soLve Manage
Options Plot Quit Remove Simplify Transfer Unrename moVe Window approx
Enter option
User Free:100% Ins Derive Algebra
```

Nekaj težav imamo lahko pri vnosu indeksov. Če želimo dobiti koordinato x prve točke, moramo vzeti prvo komponento iz a . To storimo z `a SUB 1`. Zadnjo funkcijo torej vnesemo kot

$$TO(a,b) := (a+b)/2 + \text{SQRT}(3)/6 [(a \text{ SUB } 2 - b \text{ SUB } 2), (b \text{ SUB } 1 - a \text{ SUB } 1)]$$

Da bomo videli, ali delamo prav, narišimo tako dobljeno črto. Naj bo u začetna točka (denimo $[-1, 1]$) in v končna točka ($[1, 1]$). Narisati moramo torej poligonsko črto $[u, TT(u,v), TO(u,v), TDT(u,v), v]$. Žal DERIVE izraza ne zna narisati kar v taki obliki, ampak ga moramo prej poenostaviti z ukazom `Simplify`. Ne pozabimo, da moramo pri `Options/State` v risalnem oknu nastaviti povezovanje točk.

Za računanje točk, ki določajo krivuljo višjega reda, uporabimo rekuzijo. Če je stopnja krivulje enaka 0, vrnemo krajišči, drugače pa oglišča poligonske črte, ki jo sestavljajo Kochove krivulje, eno nižje stopnje nad odseki: od u do $TT(u, v)$, od $TT(u, v)$ do $TO(u, v)$, od $TO(u, v)$ do $TDT(u, v)$ in od $TDT(u, v)$ do v . Dobimo torej poligonsko črto, sestavljeno iz množice daljic.

<pre> #4: TDT(a, b) := TT(b, a) #5: "Točka na osti." #6: TO(a, b) := $\frac{a+b}{2} + \frac{\begin{bmatrix} a & -b \\ 2 & 2 \end{bmatrix}}{6}$ #7: u := [-1, -1] #8: v := [1, 1] #9: [u, TT(u, v), TO(u, v), TDT(u, v), #10: <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>-1</td><td>-1</td></tr> <tr><td>-0.33</td><td>-0.33</td></tr> <tr><td>-0.57</td><td>0.57</td></tr> <tr><td>0.33</td><td>0.33</td></tr> <tr><td>1</td><td>1</td></tr> </table> </pre>	-1	-1	-0.33	-0.33	-0.57	0.57	0.33	0.33	1	1	
-1	-1										
-0.33	-0.33										
-0.57	0.57										
0.33	0.33										
1	1										
<p>COMMAND: Algebra Center Delete Help Move Options Plot Quit Range Scale Transfer Window axes Zoom</p> <p>Enter option</p> <p>Cross x:1 y:1 Scale x:1 y:1 Derive 2D-plot</p>											

Ves naš program za risanje Kochove krivulje je torej:

```

"Točka na tretjini med A in B."
TT(a,b) := a + (b - a) / 3
"Točka na dveh tretjinah med A in B."
TDT(a,b) := TT(b,a)
"Točka na osti."
TO(a,b) := (a+b)/2 + SQRT(3)/6 [(a SUB 2-b SUB 2),(b SUB 1-a SUB 1)]
"Kochova crta."
CRTA(a,b,n) := IF(n=0, [a,b], [CRTA(a,TT(a,b),n-1),
                             CRTA(TT(a,b),TO(a,b),n-1),
                             CRTA(TO(a,b),TDT(a,b),n-1),
                             CRTA(TDT(a,b),b,n-1)])

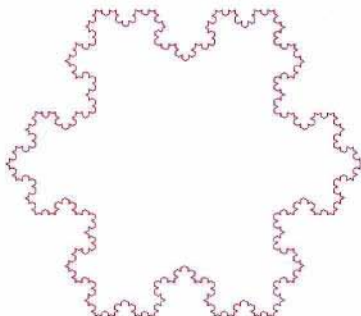
```

Seveda zadnjo funkcijo vpišemo vso v eni vrsti. Sedaj lahko vnesemo izraz

```
CRTA([-2,0],[2,0],3)
```

in ga poenostavimo s Simplify. Dobimo dolg vektor točk. Če jih narišemo (Plot/Plot), dobimo Kochovo krivuljo tretjega reda med točkama [-2,0] in [2,0].

Če narišemo sedaj Kochovo črto nad oglišči trikotnika, dobimo snežinko. Večjo "nabrazdanost" (stopnjo krivulje) bomo uporabili, lepša bo.



Za bolj zimsko vzdušje pa še nekaj snežink, dobljenih tako, da smo med sabo povezali nekaj Kochovih črt.

