

Verjetnost sekanja konveksne podmnožice -
Poročilo dela

Peter Dolenc, Jan Rems

2017

1 Uvod

2 Generiranje premic

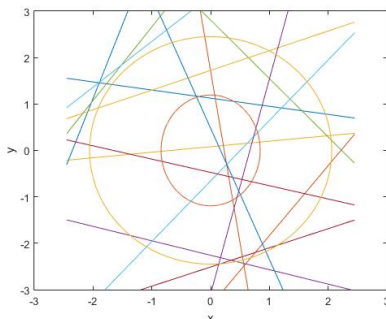
Pri projektu se je izkazalo, da je zelo pomembno, kako generiramo naključne premice. Da dobimo željeni rezultat, namreč da je verjetnost sekanja tudi male krivulje enak razmerju obsegov obeh krivulj, je potrebno generirati par (p, θ) iz enakomerno zvezne porazdelitve, kjer p predstavlja oddaljenost iz izhodišča in teče med 0 in najbolj oddaljeno točko večje konveksne krivulje od izhodišča, θ pa predstavlja kot med premico in x -osjo ter teče med 0 in 2π . Matriko velikosti $n \times 2$ s temi pari parametrov sva potem s pomožno funkcijo spremenila v matriko velikosti $n \times 2$, kjer je prvi stolpec vseboval koeficiente k , drugi pa začetne vrednosti n iz eksplcitnega zapisa premic $y = k \cdot x + n$. To nama je omogočilo, da sva določila daljice s krajišči izven večje krivulje, s katerimi sva sekala obe krivulji.

Poskusila sva tudi alternativno: generirala sva pare točk v večji konveksni množici in skozi njih potegnili premice. Izkazalo se je, da tak način generiranja premic ne pripelje do istega rezultata, saj je že za elipse v središču prišlo do velike napake. Da so premice generirane enakomerno glede na parametre p in θ se je torej izkazalo za ključno.

3 Generiranje konveksnih krivulj

3.1 Elipse

Eksperimentalni del projekta je potekal v dveh fazah. V prvi sva najino hipotezo želela pokazati na enostavnejših konveksnih krivuljah - elipsah. Napisala sva funkcijo "elipsa", ki za vhodne podatke dobi parametra $xmax$ in $ymax$, ki določata največji možni števili a in b , ki se pojavita v parametriziranem zapisu elipse $x = a \cos(t), y = b \sin(t), t \in [0, 2\pi)$. Kot izhod vrne dva poligona točk v obliki elipse z naključno izbranimi parametroma $a \in (0, xmax)$ in $b \in (0, ymax)$ iz enakomerne zvezne porazdelitve za veliko elipso in $a_m \in (0, a)$ in $b_m \in (0, b)$ za malo elipso. V zadnjem koraku sva združili funkciji, ki generirata premice in elipsi ter dodala funkcijo, ki šteje presečišča med krivuljami. Za lažjo predstavbo si oglejmo sliko 1:

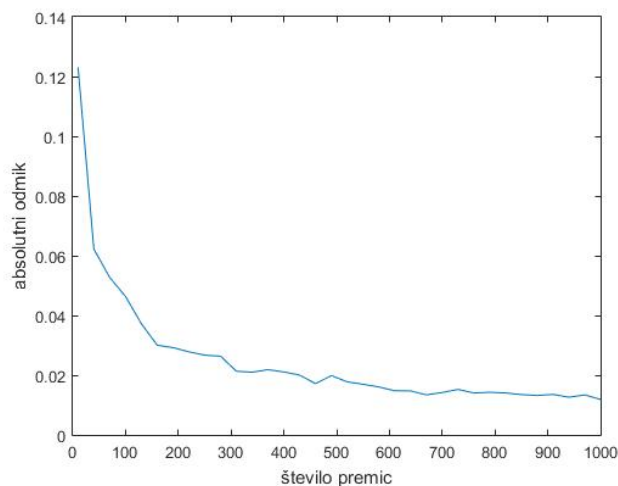


Slika 1: Primer generiranih elips in premic

Na sliki 1 vidimo 2 elipsi generirani s funkcijo “elipsa” s parametroma $xmax = 3$ in $ymax = 3$ in 15 premic. V zadnjem koraku sva uporabilo funkcijo, ki šteje presečišča premic z obema elipsama ter število presečišč z manjšo krivuljo delila s številom presečišč z večjo. Rezultat sva primerjala z oceno razmerij obsega.

Razmeroma zanesljive rezultate sva dobila, če sva preko posameznega para elips generirala več kot 300 premic. Zato sva generirala 1000 parov elips in za vsak par 1000 premic, da sva lahko ocenila rezultate v povprečju. Povprečje absolutnih vrednosti razlike med razmerjem obsegov in razmerjem premic, ki sekajo eno in drugo krivuljo je bila 0.0125, če pa dovolimo predznačeno razliko, se je v povprečju zmanjšala na 0.000046¹.

Zanimalo naju je tudi, kako hitra je konvergenca najine metode, oz. koliko premic je potrebno generirati za posamezen par elips, da bo rezultat točen. Rezultate sva dobila tako, da sva za vsak par elips primerjala absolutno razliko do točnega rezultata po vsakih 30 dodatnih premicah. Na vzorcu 100 elips je razmerje prikazano na grafu 1:



Slika 2: Graf napake v razmerju s številom premic

¹Celotna tabela rezultatov je dosegljiva na github repozitoriju