

Metoda končnih elementov: Poissonova enačba

Andrej Kolar-Požun, 28172042

12. april 2018

1 Polkrožna cev

Spet bomo reševali Poissonovo enačbo $\nabla^2 u = -f$, tokrat jo bomo prevedli na variacijski problem. Iščemo ekstrem sledečega funkcionala

$$S(u) = 1/2 \langle \nabla u, \nabla u \rangle - \langle f, u \rangle$$

Kjer skalarni produkt definiramo na sledeč način

$$\langle g, h \rangle = \int_{\Omega} dx g(x) h(x)$$

Po metodi Galerkina rešitev naše enačbe zapišemo kot linearno kombinacijo trial funkcij

$$u = \sum_{i=1}^N a_i w_i$$

Za funkcije w_i bom vzel funkcijo, ki je v x_i enaka ena, na sosedih enaka nič, vmes pa linearno narašča. S tako izbiro nam bodo koeficienti a_i povedali kar vrednost rešitve u v točki x_i .

Če zgornji funkcional variramo dobimo:

$$\begin{aligned} \sum_{j=1}^N \langle \nabla w_i, \nabla w_j \rangle a_j &= \langle f, w_i \rangle \\ \sum_{j=1}^N A_{ij} a_j &= b_i \end{aligned}$$

V tej nalogi bom na mojih problemih izvajal triangulacijo domene. To pomeni, da bom domene razdelil na majhne trikotničke v nasprotju z kvadratno mrežo s katero smo imeli do sedaj opravka.

r -ti trikotnik bo podan z koordinatami treh oglišč $((x_i, y_i), (x_j, y_j), (x_k, y_k))_r$. Ploščina r -tega trikotnika je $S_r = 1/2(x_i(y_j - y_k) + \text{cikl.perm})$

Po nekaj računih dobimo skalarne produkte, katere potrebujemo:

$$\begin{aligned} A_{ij} &= \sum ((x_j - x_k)(x_k - x_i) + (y_j - y_k)(y_k - y_i)) / 4S_r \\ b_i &= -1/3 \sum S_r \end{aligned}$$

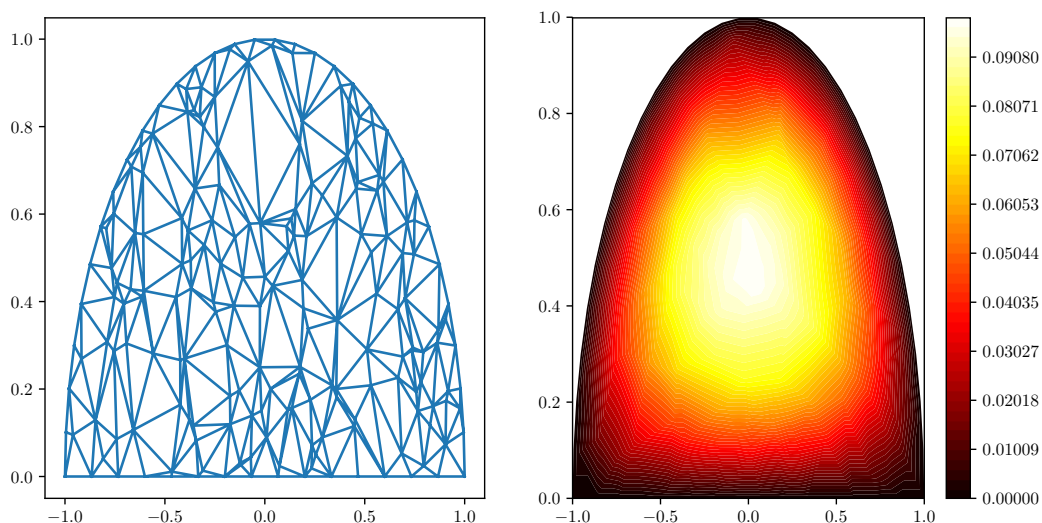
Prva vsota gre po vseh trikotnikih ob zveznici i - j , druga pa gre po vseh trikotnikih ob točki i . Izkazalo se je, da algoritem deluje veliko bolje in hitreje, če namesto po točkah iteriram po trikotnikih. Formule so podobne.

2 Polkrožna cev

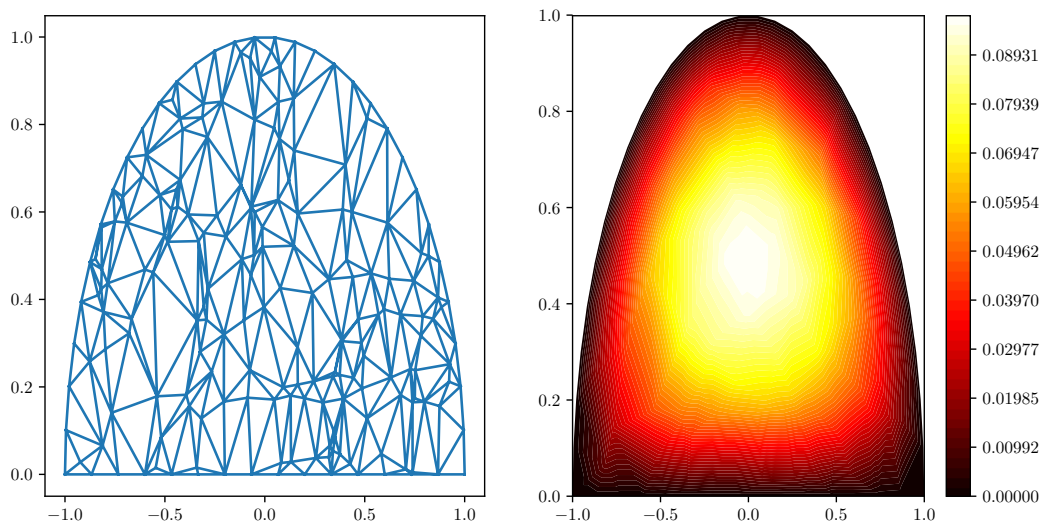
Najprej bom narisal nekaj slik, pri katerih sem povsod risal z 200 točkami, od tega jih je četrtnina bila na robu. Triangulacijo sem delal z scipy-jevo metodo Delaunay, matrični sistem pa na koncu rešil z `linalg.solve`. Poiseuillov koeficient sem računal tako, da sem povprečil vrednost pretoka po vseh 3 ogliščih trikotnika, pomnožil s ploščino trikotnika in to seštel po vseh trikotnikih.

Začnimo z naključnimi točkami porazdeljeni po enakomerni porazdelitvi po polkrogu:

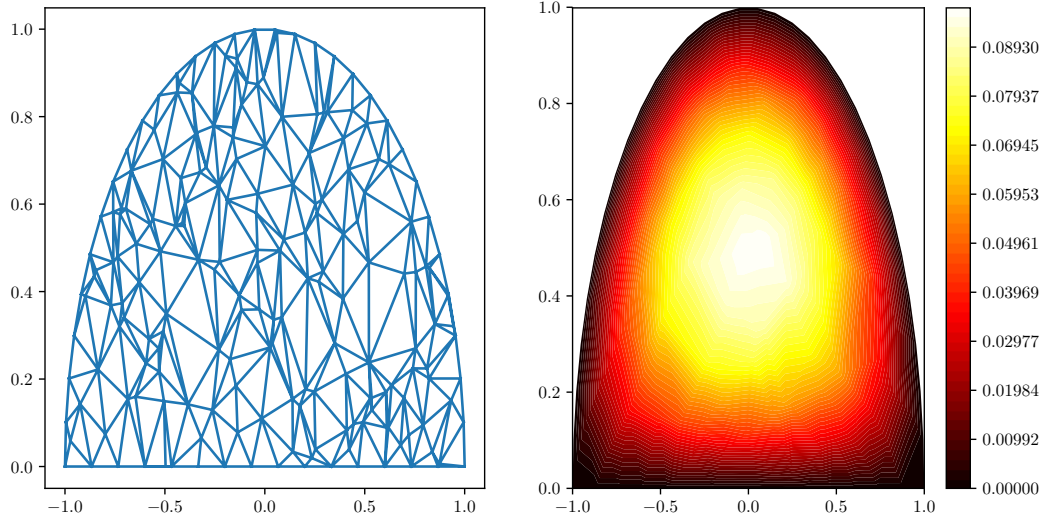
Random, Najmanjši kot je 4.13 stopinj, Pois. koef. je 0.735



Random, Najmanjši kot je 2.17 stopinj, Pois. koef. je 0.7332



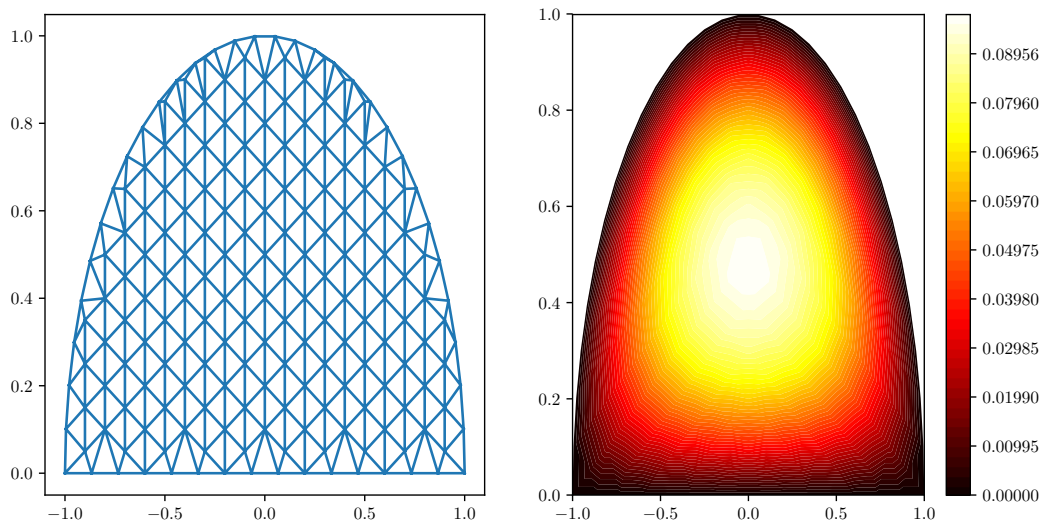
Random, Najmanjši kot je 0.04 stopinj, Pois. koef. je 0.7369



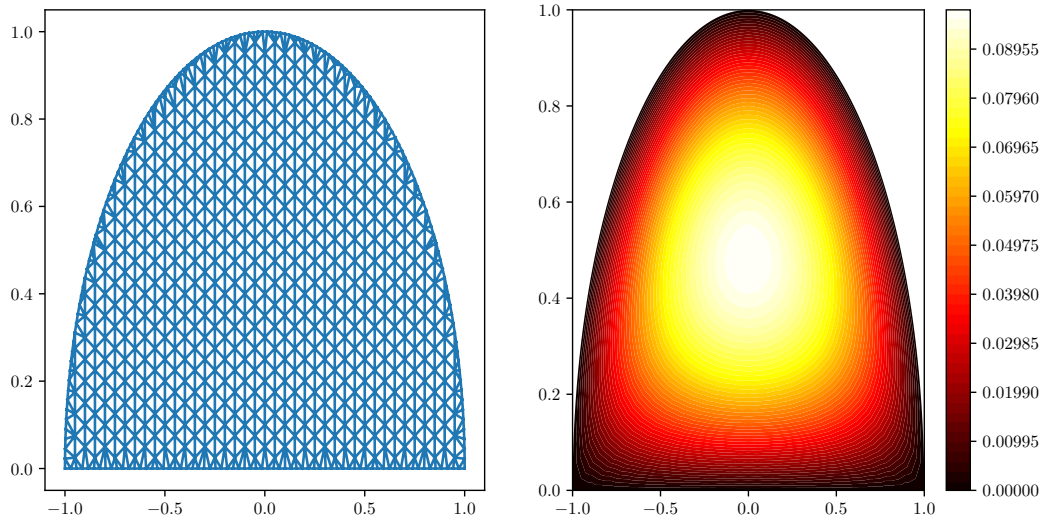
Presenetljivo sta profil in Poiseuillov koeficient zelo podobna za vse 3 primere, čeprav so najmanjši koti zelo različni. Spomnem naj, da je pravi C za polkrog približno 0.758

Poglejmo si še primer, kjer notranjost pokroga zapolnem z bolj simetrično mrežo in dodam še rob:

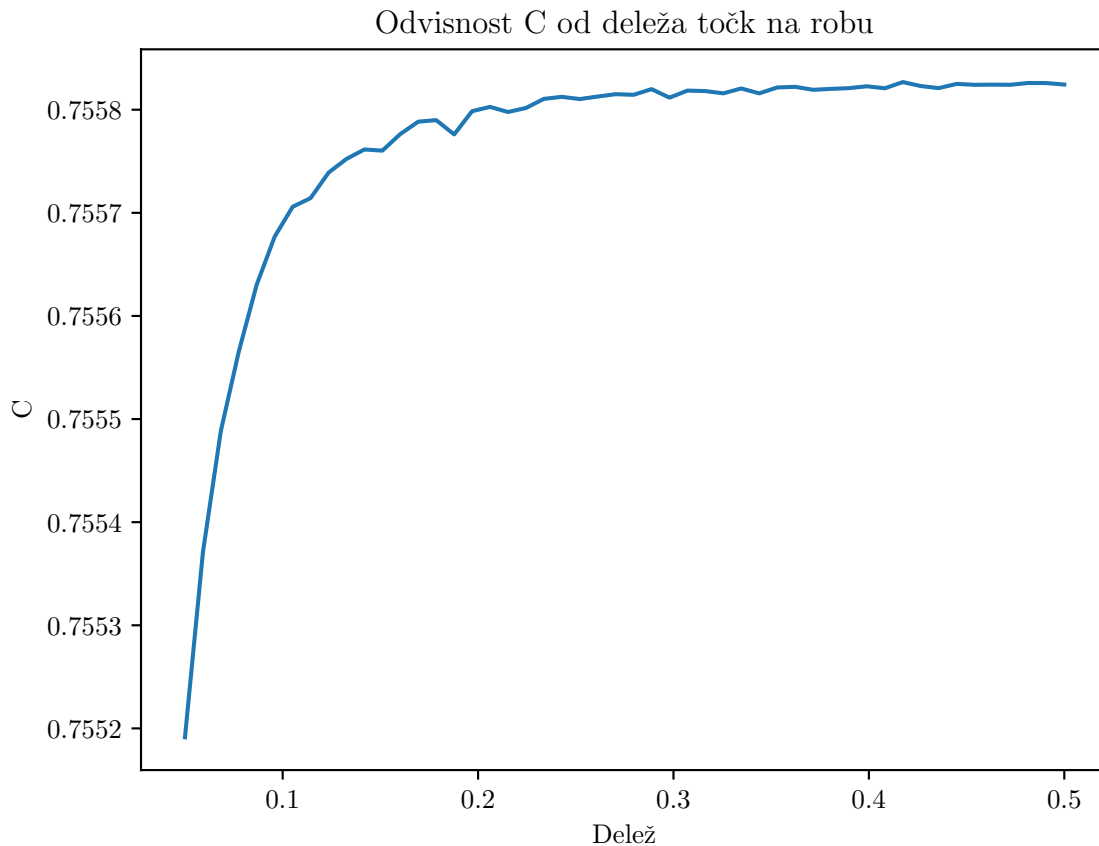
Enakomerna, Najmanjši kot je 2.7 stopinj, Pois. koef. je 0.7476



Enakomerna z 800 točkami, Najmanjši kot je 4.26 stopinj, Pois. koef. je 0.7557

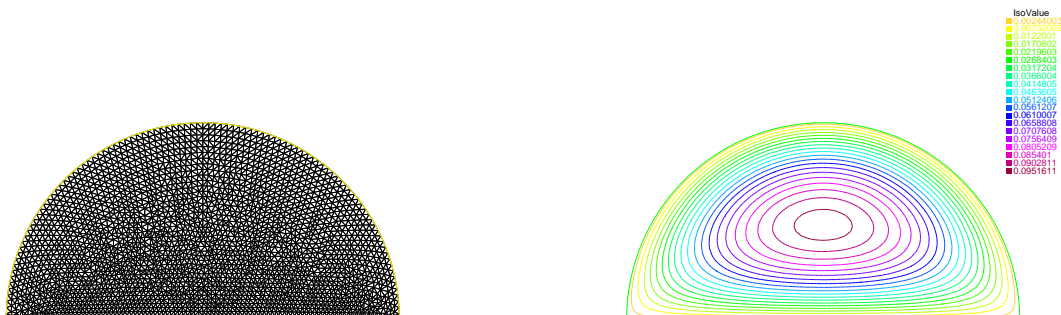


Taka triangulacija seveda še vedno ni optimalna, vendar vidimo, da je boljša kot random, saj dobimo Poiseuillov koeficient za kake 0.01 bližje pravi vrednosti. Hkrati na spodnji sliki vidimo, da izbira več točk po pričakovanju da boljši rezultat, ki je že zelo blizu pravi vrednosti. Spomnim naj, da sem v peti nalogi z relaksacijo na navadni mreži dobil za koeficient 0.779 kar je slabše.

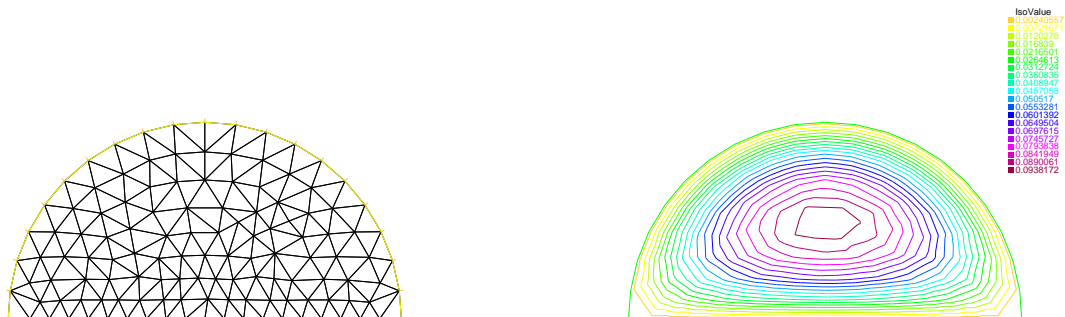


Videl sem, da vpliv najmanjšega kota ni viden na Poiseuillovem koeficientu, zato sem tukaj preveril še vpliv deleža točk na robu. V notranjosti polkroga sem imel fiksno 600 točk na rob pa sem jih pol dodal toliko da dobim primeren delež. Graf je zato pristranski saj število točk ni fiksno, pa tudi ker so trikotniki zelo različni. Delal sem sicer vedno z enakomernimi(nenaključnimi) točkami. Vseeno vidimo, da se koeficient tudi tukaj komaj spreminja.

V naslednjih slikah sem pretok narisal še z programom FreeFem++, ki sam pametneje izbere triangulacijo:



Tukaj je na robu 200 točk

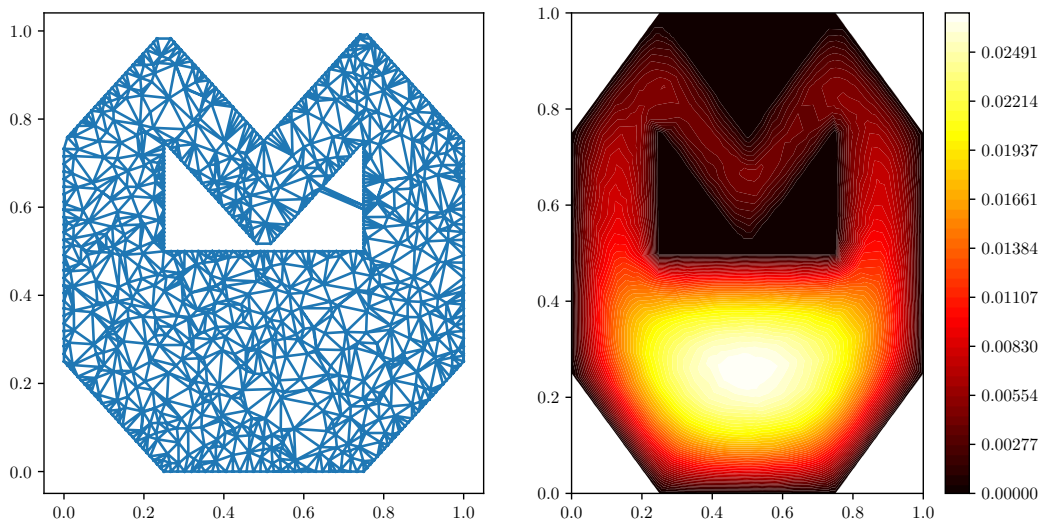


Tukaj je na robu 40 točk

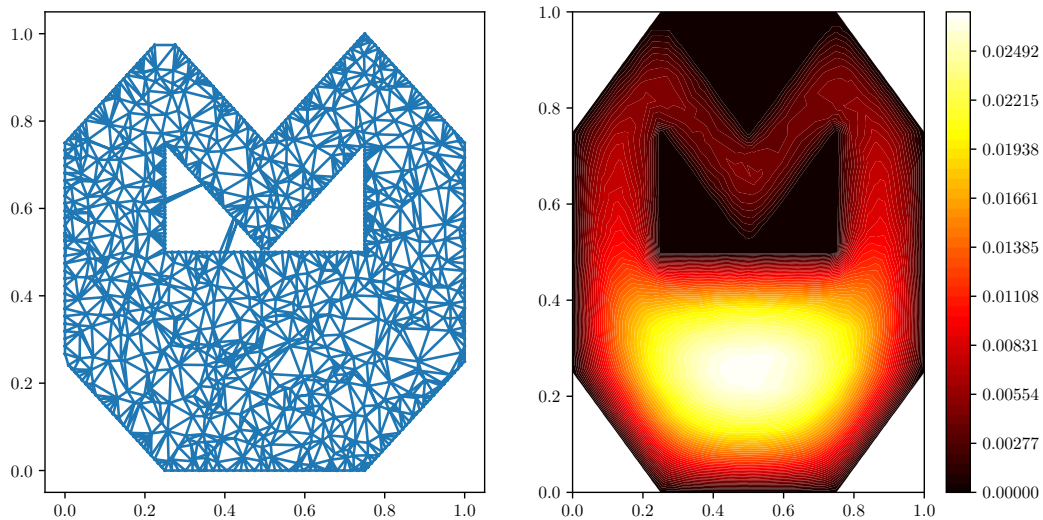
3 Mačkasta cev

V tem primeru se zapolnitev cevi z pravilnimi trikotnimi ni obnesla (matrika je bila singularna). Vedno sem torej izbral random točke. Poiseuillov koeficient za mačko iz pete naloge je 0.34585.

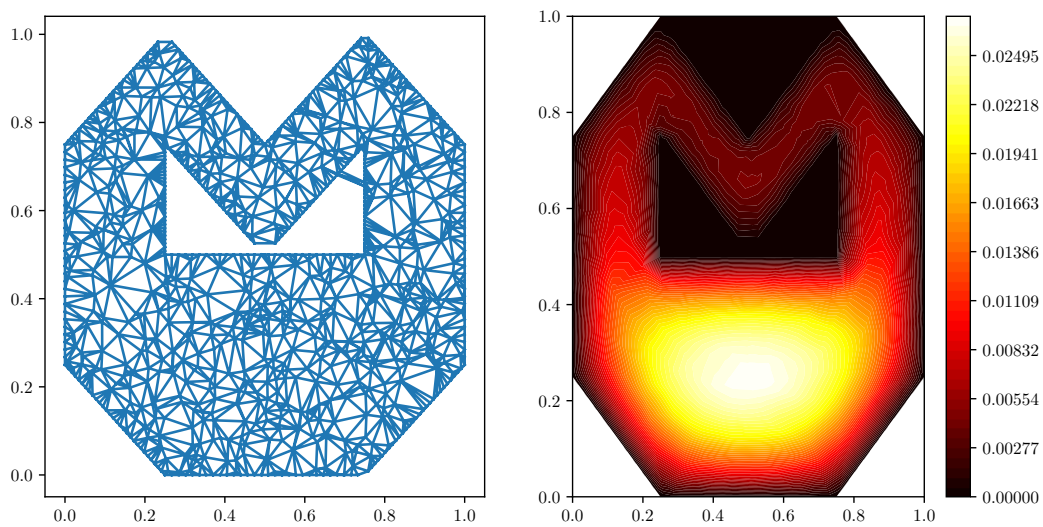
Random, Najmanjši kot je 0.75 stopinj, Pois. koef. je 0.3299



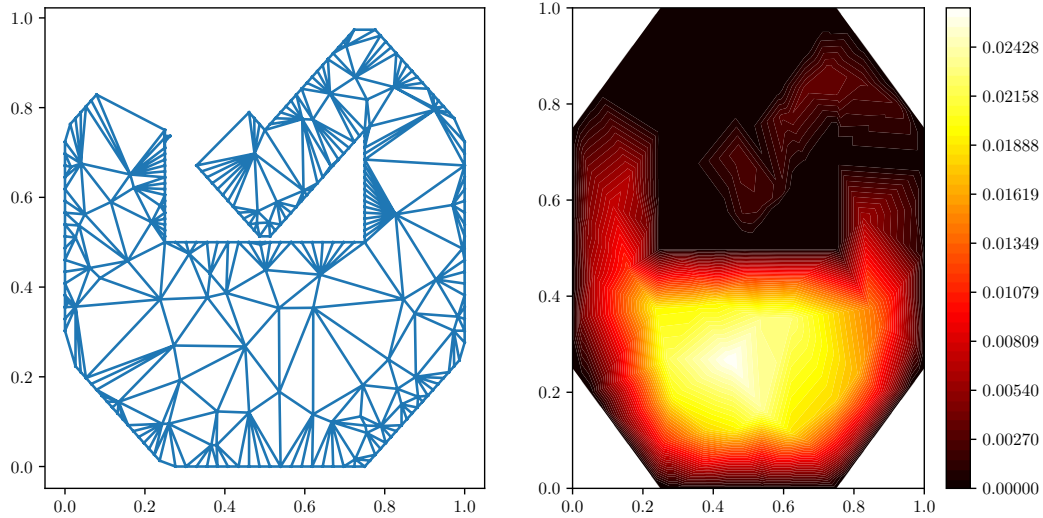
Random, Najmanjši kot je 0.28 stopinj, Pois. koef. je 0.3308



Random, Najmanjši kot je 0.8 stopinj, Pois. koef. je 0.3289

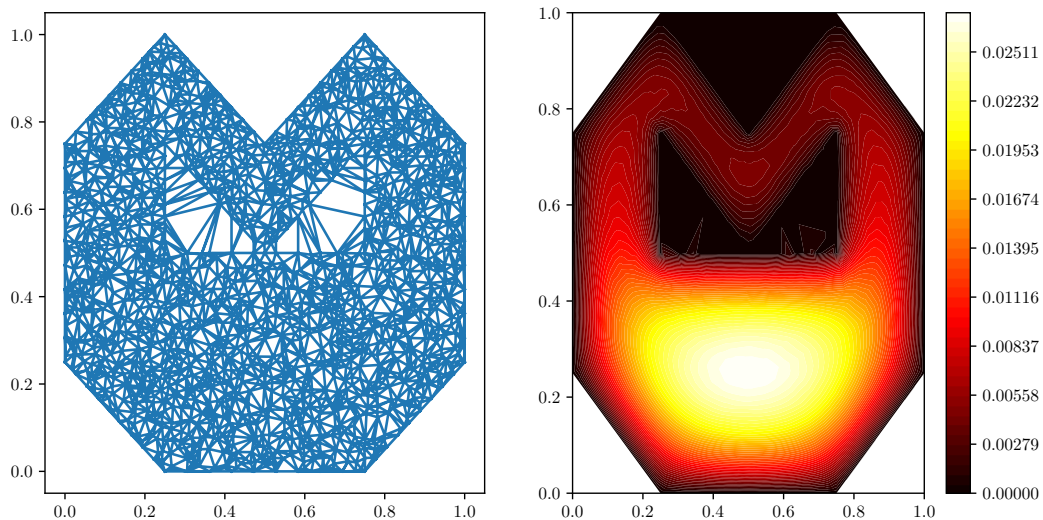


Random, Najmanjši kot je 1.88 stopinj, Pois. koef. je 0.3228

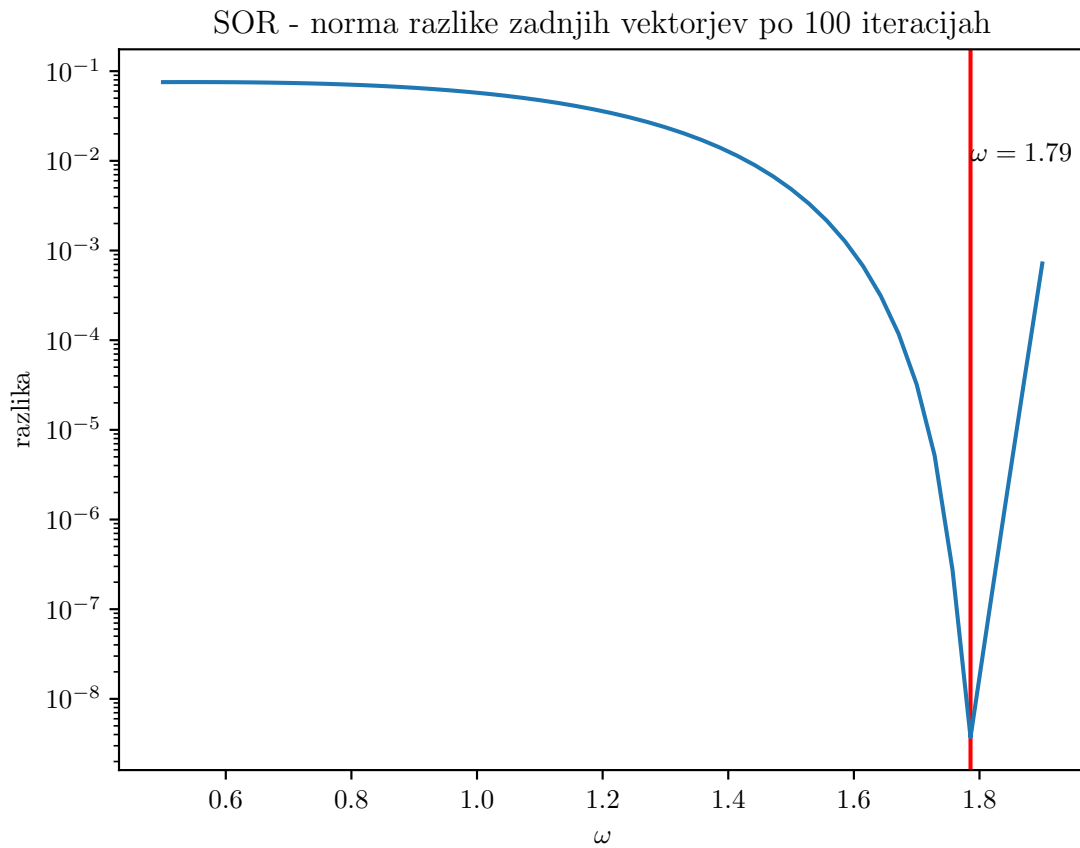


Če izberemo premalo točk seveda sploh ne moremo dobro opisati mačke. (Na levi nekje delno ni roba, ker sem trikotnike, ki imajo vse točke na robu odstranil, da jih Delaunay ne poveže zunaj domene

Random, Najmanjši kot je 0.25 stopinj, Pois. koef. je 0.3052

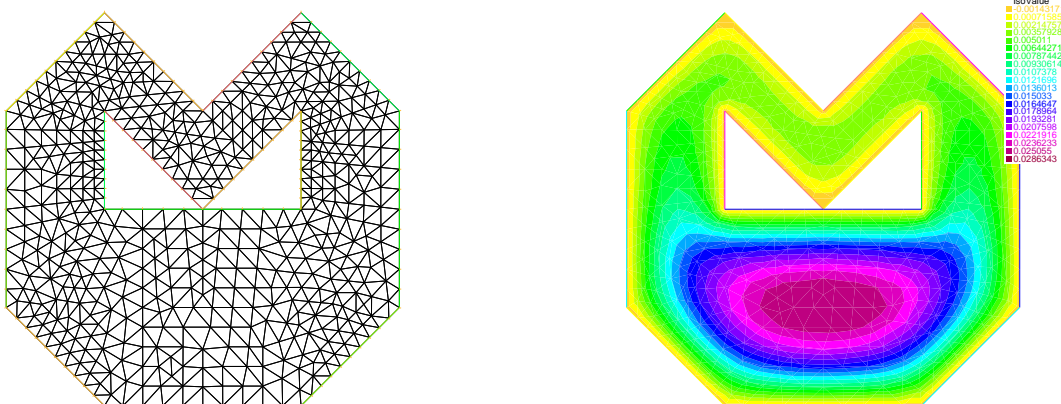


To se je v manjši meri pojavilo že pri prejšnjih primerih, vendar če je točk v mački veliko več kot na robu, jih lahko poveže v trikotnike preko območij, ki so izven domene, kar ni dobro (v tem primeru kakšna velika napaka sicer ni opazna)



Metoda SOR na matriki za triangulirano mačko z random točkami. Gledal sem normo razlike med zadnjim in predzadnjim vektorjem pri iteraciji.

Še rešitve z freefem:



Tukaj je na robu 140 točk

