Zlepki nad triangulacijami - DN 7

April 24, 2023

Uporabljeni MATLAB fajli iz prejšnjih nalog:

- bpolyval.m - izračuna vrednost polinoma v Bernstein-Bezier repr.
- changeBasis.m - Izračuna koeficiente polinoma v drugem baricentričnem ogrodju.
- checkSmoothnessSpline.m - Izračuna red gladkosti med trikotniki.
- coeffSmoothness.m - Določi koeficiente potrebne za dan red gladkosti.
- constructSpline.m - iz podatkov (α) izračuna zlepek (koeficiente nad vsakim trikotnikom)
- decasteljau.m - decasteljau-ev algoritem (splošen)
- evaluateSpline.m - izračuna vrednost zlepka nad triangulacijo v dani točki.
- interpolation.m - interpolira podatke (vrednosti + grad) na trikotniku.

Novi fajli:

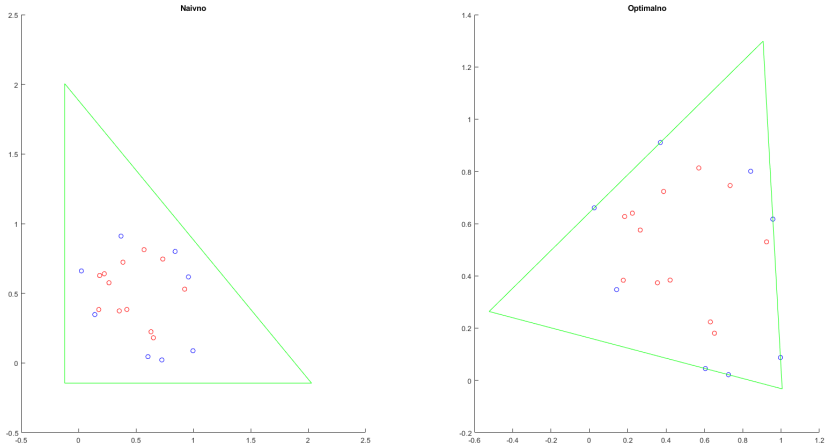
- barycentricMap.m - točko iz $\mathbb{R}^2$ predstavi z baricentričnimi koordinatami
- getPointCloud.m - za dano triangulacijo + vertex vrne ta vertex + primerne točke na vseh incidenčnih povezavah, kot opisano v navodilih.
- getTriangleNaive.m - naivni algoritem za generacijo trikotnika, ki vsebuje vse dane točke.
- getTriangleOptimal.m - algoritem za generacijo optimalnega trikotnika, ki vsebuje vse dane točke.
- testEnclosingTriangle.m - skripta, ki zgenerira naključne točke in prikaže trikotnik, ki jih vsebuje.
- testHW7.m - skripta, ki naredi korake, opisane v navodilih domače naloge (zgradi zlepek za nek i,r).

Za določanje trikotnika, ki vsebuje vse dane točke, sem najprej ubral zelo naiven pristop (za katerega vemo, da vsebuje vse točke, lahko pa ima zelo preveliko ploščino): Iz danih točk (x_i, y_i) $(i = 1, \dots N)$ izračunam njihovo povprečje (x_c, y_c). To povprečje potem vzamem kot center krožnice, katere radij je tak, da zavzame vse točke (x_i, y_i). Z osnovno geometrijo lahko potem določim stranice

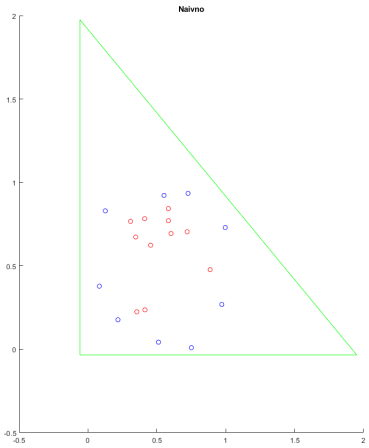
pravokotnega (zaradi enostavnosti) trikotnika, katerega včrtana krožnica je pravkar konstruirana krožnica.

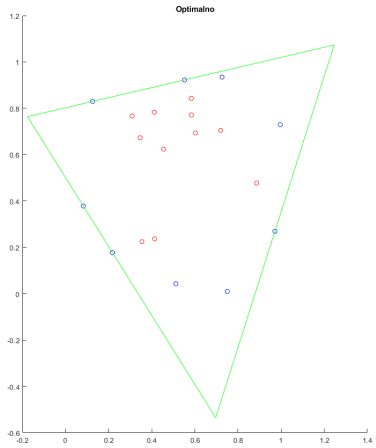
Optimalen trikotnik (v smislu najmanjše ploščine) pa dobimo tako, da iz točk najprej izračunamo njihovo konveksno ogrinjačo in potem najdemo najmanjši trikotnik, ki jo vsebuje. Za ta problem sem rešitev našel v članku "O'Rourke, A. Aggarwal, S. Maddila, and M. Baldwin, An optimal algorithm for finding minimal enclosing triangles". Ta pravi, da je optimalen trikotnik tak, v katerem dve stranici sovpadata s stranicama konveksne ogrinjače, tretja pa lahko ali sovpada s stranico konveksne ogrinjače ali pa se je na sredini dotika. Brute force poiščem vse take trikotnike in vrnem tistega, ki ima najmanjšo ploščino. (v članku je sicer opisan še računsko manj zahteven algoritem, katerega mi pa ni uspelo implementirati)

Najprej ta algoritem preverimo na 20 naključnih točkah iz kvadrata $[0, 1] \times [0, 1]$:



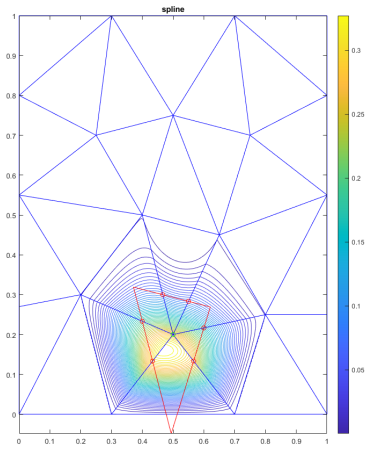
Rdeča barva - generirane točke, modra barva - generirane točke, ki so del konveksne ogrinjače. Levo - naiven trikotnik, desno - optimalen trikotnik. Primer, ko vse tri stranice sovpadajo s stranico konv. ogrinjače.

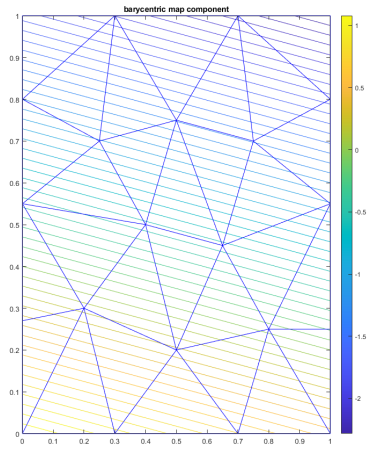




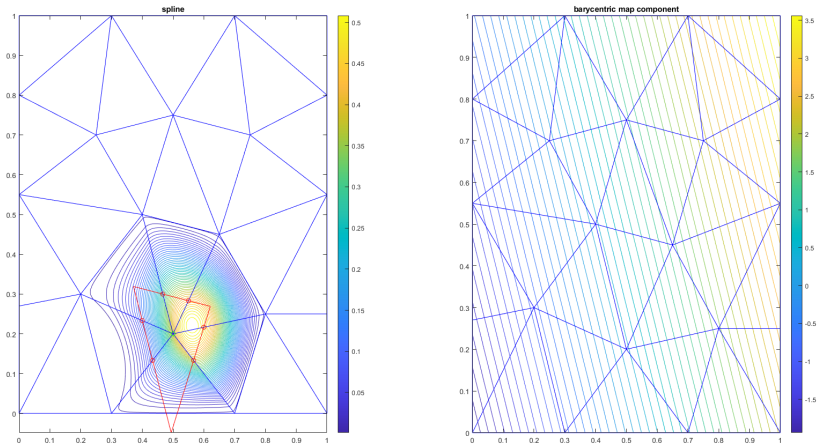
Rdeča barva - generirane točke, modra barva - generirane točke, ki so del konveksne ogrinjače. Levo - naiven trikotnik, desno - optimalen trikotnik. Primer, ko samo dve stranici sovpadata s stranicama konv. ogrinjače.

V nadaljevanju poglejmo zlepke $B_{i,r}$, kjer uporabimo algoritem za optimalen trikotnik:

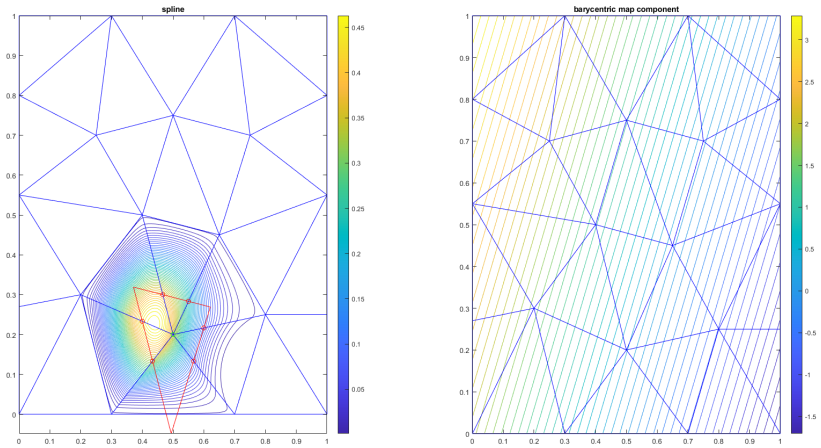




$$i = 7, r = 1$$

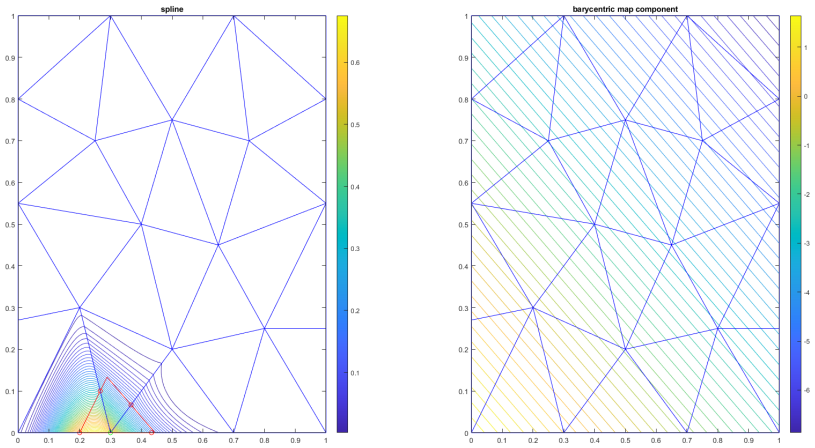


$i = 7, r = 2$

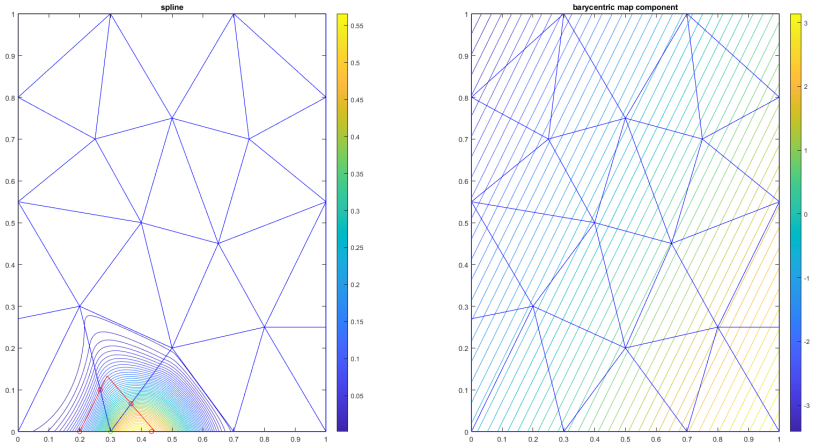


$i = 7, r = 3$

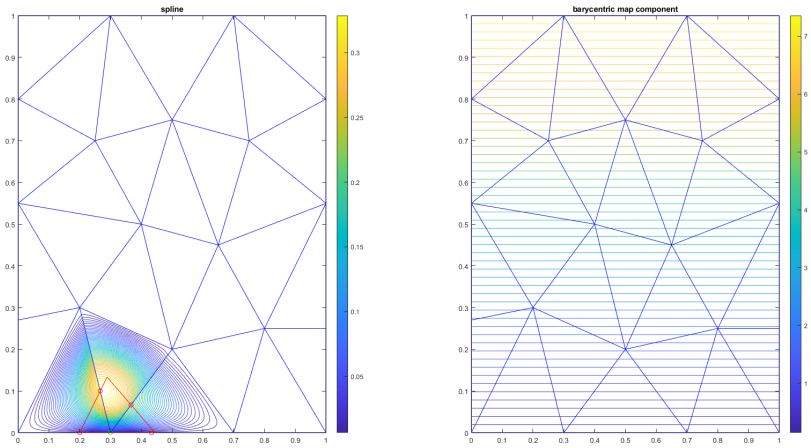
Kot pričakovano je zlepek neničelen (pozitiven) le na trikotnikih, katerih eno izmed oglišč je i. Algoritem za določanje reda gladkosti pove, da je zlepek C^0 na povezavah teh trikotnih, na ostalih pa je gladek. Parameter r nam pove, v kateri smeri smeri od verteksa i ima zlepek maksimum. To je približno smer, v katero kaže gradient funkcije $\mathrm{bar}_r\langle q_i\rangle(x, y)$.


spline
barycentric map component

$i = 2, r = 1$


spline
barycentric map component

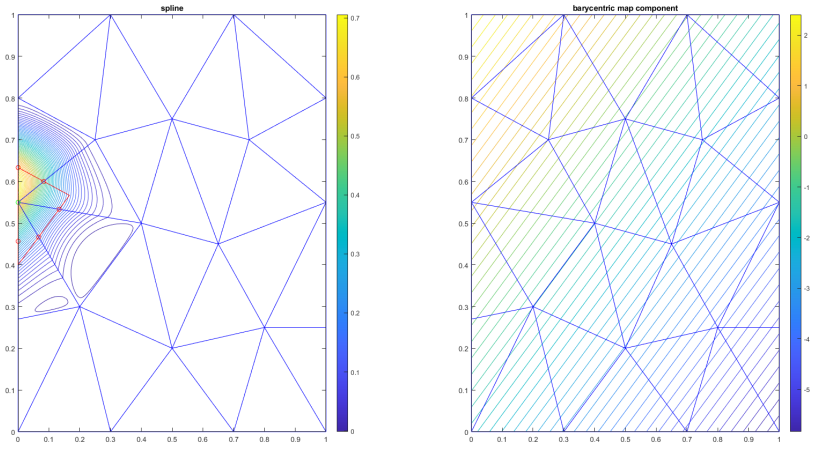
$i = 2, r = 2$



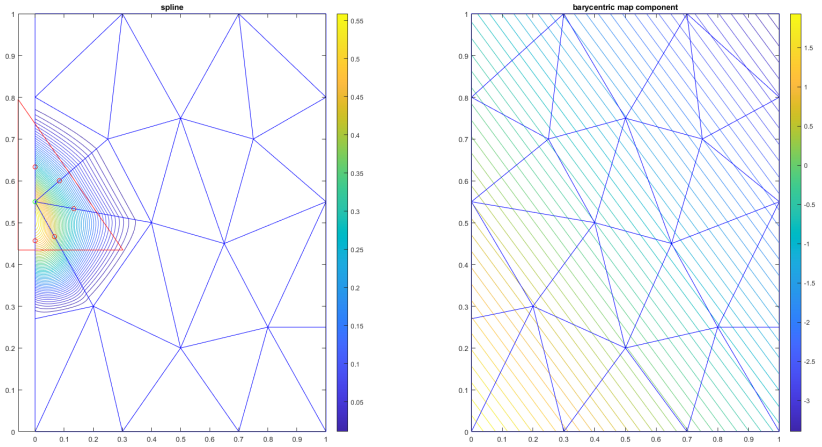
$$i = 2, r = 3$$

Če je trikotnik na robu je situacija podobna. Tukaj je algoritem za iskanje optimalnega trikotnika včasih imel težave s singularnostjo matrik (algoritem na določenih korakih rešuje linearne sisteme). Nisem šel v podrobnosti, vendar je rezultat vseeno smiselen trikotnik zato predvidevam, da so tisti skoraj singularni kandidati za optimalen trikotnik imeli tako veliko ploščino, da na koncu niso bili upoštevani.

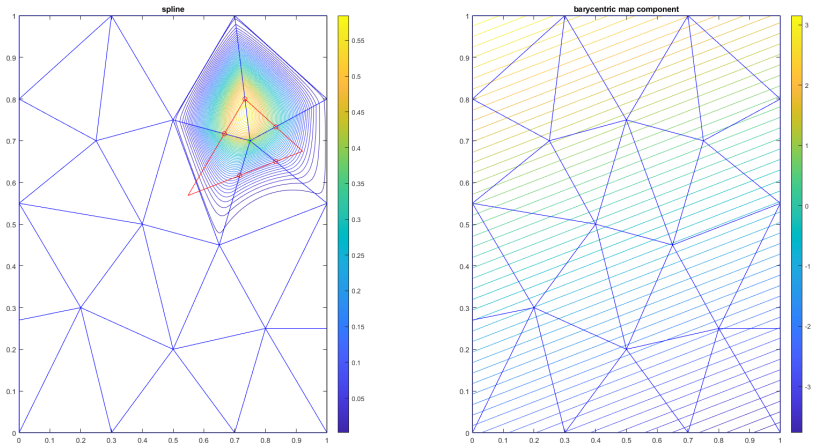
Zadnje slike pokažejo še, da so rezultati kvalitativno podobni, če za določanje trikotnika uporabimo naivni algoritem:



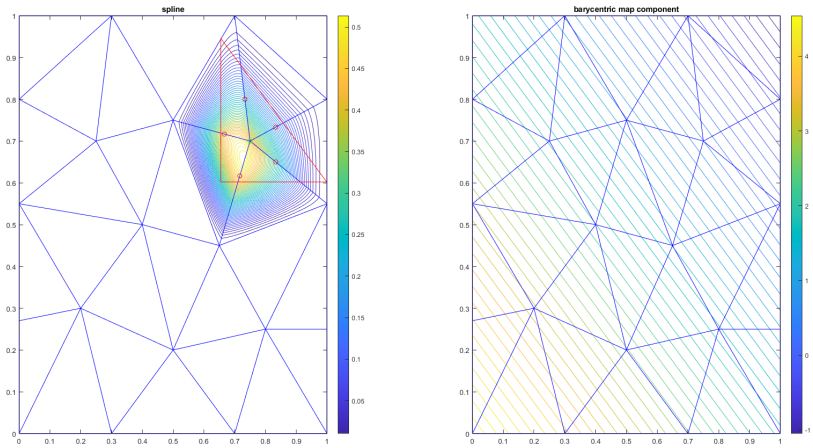
$i = 10, r = 1$, optimalen trikotnik



$i = 10, r = 1$, naiven trikotnik



$i = 17, r = 1$, optimalen trikotnik



$i = 17, r = 1$, naiven trikotnik