Metode adaptivnog generiranja mreže zadane CAD modelom

Gradečak, Davorin

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Davorin Gradečak

METODE ADAPTIVNOG GENERIRANJA MREŽE ZADANE CAD MODELOM

Diplomski rad

Voditelj rada:
prof. dr. Luka Grubišić

Zagreb, srpanj 2021.

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

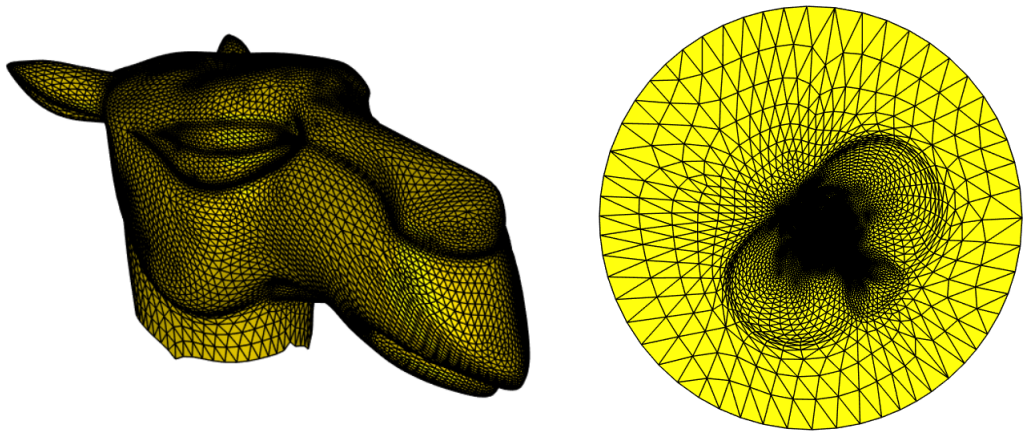
Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

Sadržaj

Uvod

U ovom radu bavimo se algoritmima za korekciju CAD modela tanokstijene strukture u postupku automatskog generiranja žičanog modela modeliranog tijela. Često se događa da su CAD modeli određeni triangulacijama, kao što su stereolitografska (STL) triangulacija. U takvoj triangulaciji ploha je određena kolekcijom trokuta koji su određeni njihovim vrhovima te jediničnim vektorom normale. Zbog grešaka u geometrijskom modeliranju kod takvog opisa ploha se često javljaju tanki isječci, odnosno trokuti s iznimno malim kutevima. Izbjegavanje pojave takvih elemenata trijangulacije uzrokuje poteškoće u konvergenciji algoritmima za automatsko generiranje mreže. Općenito, elementi loše kvalitete utječu na primjenu numeričkih metoda kao što su rješavanje simulacijske zadaće korištenjem metode konačnih elemenata. Degenerirani elementi negativno utječu na kondiciju matrice sustava (u slučaju da linearne sustave rješavamo direktnim metodama kao što su LU faktorizacija), odnosno otežavaju konvergenciju iterativnim metodama kao što su metoda konjugiranih gradijenata. Kvalitetu strukture mreže možemo poboljšati korištenjem metode ponovnog generiranja mreže koja je tema ovog rada. Na slici 0.1 vidimo ilustraciju glavne



Slika 0.1: Žičani model glave deve i njegova projekcija na jediničnu kružnicu $\mathcal{S}'$

zadaće ovog rada. Polazeći od geometrijskog CAD modela devine glave kao plohe u $\mathbb{R}^3$

želimo konstruirati (u našem slučaju po dijelovima linearno i injektivno) preslikavanje u domenu u $\mathbb{R}^2$. Nakon toga sliku plohe u $\mathbb{R}^3$ želimo trijangulirati korištenjem standardnog robusnog softvera za generiranje triangulacija u ravnini. Korigirani žičani model plohe sada možemo dobiti kao prasliku trijangulacije u $\mathbb{R}^2$.

U nastavku ćemo sažeto izložiti neke aspekte teorije konačnih elemenata koji su potrebni za razumijevanje zadaća koje ćemo promatrati u ovom radu. Također, precizirat ćemo kriterije kvalitete mreže koji će se koristiti te izložiti strukturu rada.

Osnovna notacija i definicije

Neka je $\mathcal{V}$ realan unitaran prostor sa skalarnim produktom $(\cdot, \cdot)$. Funkciju $q : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ koja je

1. Simetrična – $q(v_1, v_2) = q(v_2, v_1)$, $v_1, v_2 \in \mathcal{V}$
2. Linearna u prvom i u drugom argumentu – $q(\alpha v_1 + \beta v_2, v_3) = \alpha q(v_1, v_3) + \beta q(v_2, v_3)$, $v_1, v_2, v_3 \in \mathcal{V}$, $q(v_3, \alpha v_1 + \beta v_2) = \alpha q(v_3, v_1) + \beta q(v_3, v_2)$, $v_1, v_2, v_3 \in \mathcal{V}$
3. Nenegativna – $q(v) \geq 0$, $v \in \mathcal{V}$

nazivamo realna kvadratna forma. Ako je dan i vektor $f \in \mathcal{V}$, tada definiramo apstraktnu Drichletovu energiju kvadratne forme q i vanjske sile f izrazom

$$E(v) = \frac{1}{2} q(v, v) - (f, v)$$

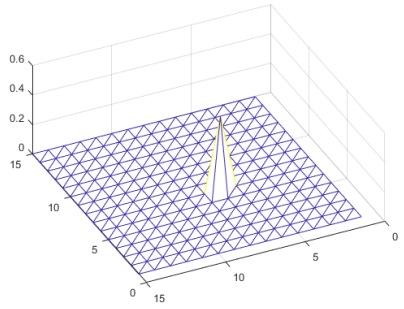
U ovoj definiciji prostor $\mathcal{V}$ može biti konačno ili beskonačno dimenzionalan. Tipično, analizirat ćemo kvadratnu formu na nekom beskonačno dimenzionalnom prostoru funkcija promatrajući njeno ponašanje na "dobro odabranom" konačno dimenzionalnom potprostoru. U metodi konačnih elemenata minimiziramo Drichletovu energiju po prostoru po dijelovima polinomijalnih funkcija.

Definirajmo sada što znači po djelovima polinomijalna funkcija stupnja $k \in \mathbb{N}$. U slučaju kada se nalazimo u jednoj dimenziji teselacija intervala $\mathcal{S} = [a, b]$ je dana njegovom subdivizijom $a = x_0 < x_1 < \cdots < x_n = b$. Restrikcije po dijelovima polinomijalne funkcije na element izabrane teselacije $[x_i, x_{i+1}]$ mora biti polinom najviše stupnja k, a sama funkcija je globalno neprekidna na $\mathcal{S} = [a, b]$. U slučaju dvodimenzionalnih problema, promatramo područje $\mathcal{S} \subset \mathbb{R}^2$ koje mora biti ograničeno i imati gladak ili poligonalni rub. Neka je $\mathcal{T}$ konačan skup poligona, ili poopćenih poligona (poligoni kojima je najviše jedna stranica glatka krivulja a ostale su dužine) koji su u parovima disjunktni odnosno u parovima dijele najviše zajedničku stranicu ili vrh i za koje vrijedi $\cup_{T \in \mathcal{T}} T = \mathcal{S}$. Za funkciju definiranu na $\mathcal{S}$ kažemo da je po dijelovima polinomijalna stupnja k, obzirom na teselaciju $\mathcal{T}$, ako je restrikcija funkcije na svaki T polinom stupnja k i funkcija je neprekidna na $\mathcal{S}$.

Prostor po dijelovima polinomijalnih funkcija stupnja 1 je konačnodimenzionalan te je zadan kao linearna ljuska skupa funkcija oblika, odnosno baznih funkcija. Funkcije oblika su po dijelovima linearne i poprimaju vrijednost 1 u jednom čvoru mreže, a u svim ostalima su 0. Jednostavan primjer funkcija oblika u jednoj dimenziji je sljedeći:

$$\varphi_k(x) = \begin{cases} \frac{x-x_{k-1}}{x_k-x_{k-1}}, & x \in [x_{k-1}, x_k] \\ \frac{x_{k+1}-x}{x_{k+1}-x_k}, & x \in [x_k, x_{k+1}] \\ 0 & \text{inače.} \end{cases}$$

U dvije dimenzije funkcija oblika je također jedan u jednom čvoru i nula u svim ostalim. No za razliku od jednodimenzionalne situacije, ona će biti nenula u svim onim elementima teselacije T koji diraju čvor u kojem je bazna funkcija 1. Primjer u dvije dimenzije vidimo na sljedećoj slici.



Slika 0.2: Bazna funkcija prostora po dijelovima linearnih funkcija na teselaciji kvadrata u istokračne pravokutne trokute.

Neka je sada $\{\varphi_i : i = 1, \cdots, N\}$ skup baznih funkcija vezanih uz triangulaciju $\mathcal{T}$. Minimum Dirichletove energije na njihovoj linearnoj ljusci, tj. na prostotu $\mathcal{V} = \text{span}\{\varphi_i : i = 1, \cdots, N\}$ je funkcija

$$u = \sum_{i=1}^{N} \alpha_i \varphi_i \tag{1}$$

pri čemu koeficijente α_i, $i = 1, \cdots, N$ dobivamo kao rješenje linearne jednadžbe

$$\begin{bmatrix} q(\varphi_1, \varphi_1) & \cdots & q(\varphi_1, \varphi_N) \\ \vdots & \ddots & \vdots \\ q(\varphi_N, \varphi_1) & \cdots & q(\varphi_N, \varphi_N) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{bmatrix} = \begin{bmatrix} (\varphi_1, f) \\ \vdots \\ (\varphi_N, f) \end{bmatrix}.$$

Ovdje smo dodatno pretpostavili da je forma q pozitivno definitna, odnosno da postoji $\delta > 0$ takav da je $q(v, v) \geq \delta(v, v)$. Definirajmo sada matricu $K_{ij} = q(\varphi_i, \varphi_j)$, $i, j = 1, \cdots, N$. Ovu matricu nazivamo matrica krutosti forme q.

Metoda Lagrangeovih multiplikatora

Problem traženja uvjetnog ekstrema funkcije u ovom radu rješavamo metodom Lagrangeova multiplikatora. Kako bismo našli minimum funkcije $f(x) \in C^1(\mathbb{R}^n, \mathbb{R})$ koja zadovoljava uvjet $g(x) = 0$, $g \in C^1(\mathbb{R}^n, \mathbb{R}^m)$, definiramo Lagrangeovu funkciju

$$\mathcal{L}(x, \lambda_1, \dots \lambda_m) = f(x) - \sum_{j=1}^{m} \lambda_j g_j(x)$$

Teorem 0.0.1. *Neka je $f(x) \in C^1(\mathbb{R}^n, \mathbb{R})$ te $g(x) \in C^1(\mathbb{R}^n, \mathbb{R}^m)$. Ako funkcija f ima lokalni ekstrem u x^* na skupu $g^{-1}(0)$, tada postoji λ^* takav da je (x^*, λ^*) stacionarna točka Langrangeove funkcije $\mathcal{L}$.*

Da bismo odredili stacionarne točke, potrebno je riješiti sustav:

$$\nabla_{x,\lambda_1,\dots,\lambda_m} \mathcal{L}(x, \lambda_1, \dots, \lambda_m) = 0,$$

odnosno tomu ekvivalentno:

$$\begin{cases} \nabla_x f(x) - \sum_{j=1}^{m} \lambda_j \nabla_x g_j(x) = 0 \\ g_1(x) = \cdots = g_m(x) = 0 \end{cases} \tag{2}$$

U ovom radu ćemo promatrati funkcije f koje su kvadratične, a ograničenja će biti zadana linearnim funkcijama. Posljedično, traženje stacionarnih točaka i još važnije ekstrema se za kvadratične funkcije svodi na rješavanje linearnih sustava [15].

Floydov algoritam

Neka je $G = (V, E)$ težinski graf, te neka nema negativan ciklus (ciklus kojem je suma težina manja od 0). Floydov, odnosno Floyd-Warshallov algoritam nalazi najkraći put između svakog para vrhova na tom grafu. Neka je $|V| = N$ te neka su vrhovi grafa numerirani s $1, 2, \dots, N$. Promotrimo funkciju `najkraciPut(i,j,k)` koja za dane vrhove grafa i i j vraća najkraći put iz i do j samo koristeći skup vrhova $\{1, \dots, k\}$. Koristeći ovu funkciju, cilj nam je izračunati `najkraciPut(i,j,N)` za svaki $i, j \in \{1, 2, \dots, N\}$.

Za svaki par vrhova, `najkraciPut(i,j,k)` može prolaziti kroz vrh k iz i koristeći proceduru `najkraciPut(i,k,k-1)` te iz k do j koristeći `najkraciPut(k,j,k-1)`, ili ne prolazi kroz vrh k, u kojem slučaju je taj najkraći put jednak `najkraciPut(i,j,k-1)`. Sad se nazire rekurzivna formula za `najkraciPut(i,j,k)`, uz oznaku da je težina brida između i i j jednaka $w(i, j)$:

$$\begin{aligned} najkraciPut(i, j, 0) &= w(i, j) \\ najkraciPut(i, j, k) &= min(najkraciPut(i, j, k-1), \\ &\qquad najkraciPut(i, k, k-1) + najkraciPut(k, j, k-1)) \end{aligned}$$

Za Floydov algoritam, inicijaliziramo matricu incidencije D:

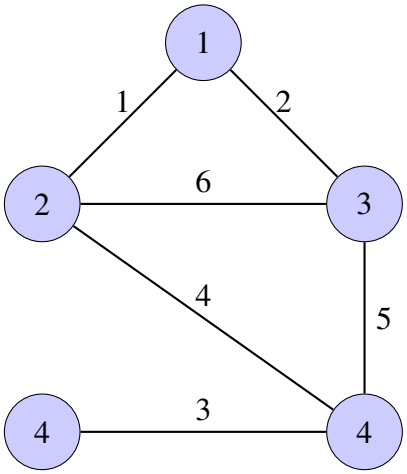
$$D_{ij} = \begin{cases} w_{ij} & \text{ako } \{i,j\} \in E, \\ 0 & \text{ako } i = j \\ \infty & \text{inače} \end{cases}$$

Slijedi pseudokod za sam algoritam, koji vraća matricu D kojoj vrijednost na elementu $D(i, j)$ označava duljinu najkraćeg puta iz i u j na zadanom grafu[1]:

for $k = 1, \ldots, N$ **do**
  for $i = 1, \ldots, N$ **do**
    for $j = 1, \ldots, N$ **do**
      if $D(i, j) > D(i, k) + D(k, j)$ **then**
        $D(i, j) = D(i, k) + D(k, j)$
      end if
    end for
  end for
end for

Mi ćemo u ovom radu Floydov algoritam koristiti kako bi pronašli par točaka na rubu triangulacije koje su najudaljenije.

Primjer 0.0.2. *Slijedi primjer izvedbe Floydovog algoritma na grafu $G(V, E)$ sa slike: Svaka matrica D_i, $i = 1, \ldots 5$ će predstavljati jednu iteraciju po najgornjoj petlji (po k)*



Slika 0.3: Težinski graf G na kojem računamo najkraće puteve

algoritma. Podebljani elementi su elementi koji su u toj iteraciji promijenili vrijednost.

$$D = D_0 = \begin{pmatrix} 0 & 1 & 2 & \infty & \infty \\ 1 & 0 & 6 & \infty & 4 \\ 2 & 6 & 0 & \infty & 5 \\ \infty & \infty & \infty & 0 & 3 \\ \infty & 4 & 5 & 3 & 0 \end{pmatrix} \qquad D_1 = \begin{pmatrix} 0 & 1 & 2 & \infty & \infty \\ 1 & 0 & \mathbf{3} & \infty & 4 \\ 2 & \mathbf{3} & 0 & \infty & 5 \\ \infty & \infty & \infty & 0 & 3 \\ \infty & 4 & 5 & 3 & 0 \end{pmatrix}$$

$$D_2 = \begin{pmatrix} 0 & 1 & 2 & \infty & \mathbf{5} \\ 1 & 0 & 3 & \infty & 4 \\ 2 & 3 & 0 & \infty & 5 \\ \infty & \infty & \infty & 0 & 3 \\ \mathbf{5} & 4 & 5 & 3 & 0 \end{pmatrix} \qquad D_3 = \begin{pmatrix} 0 & 1 & 2 & \infty & 5 \\ 1 & 0 & 3 & \infty & 4 \\ 2 & 3 & 0 & \infty & 5 \\ \infty & \infty & \infty & 0 & 3 \\ 5 & 4 & 5 & 3 & 0 \end{pmatrix}$$

$$D_4 = \begin{pmatrix} 0 & 1 & 2 & \infty & 5 \\ 1 & 0 & 3 & \infty & 4 \\ 2 & 3 & 0 & \infty & 5 \\ \infty & \infty & \infty & 0 & 3 \\ 5 & 4 & 5 & 3 & 0 \end{pmatrix} \qquad D_5 = \begin{pmatrix} 0 & 1 & 2 & \mathbf{8} & 5 \\ 1 & 0 & 3 & \mathbf{7} & 4 \\ 2 & 3 & 0 & \mathbf{8} & 5 \\ \mathbf{8} & \mathbf{7} & \mathbf{8} & 0 & 3 \\ 5 & 4 & 5 & 3 & 0 \end{pmatrix}$$

Implicitno restartani Arnoldijev algoritam

U ovom radu koristimo implictno restartani Arnoldijev algoritma kako bi riješili generalizirani problem svojstvene vrijednosti (GEP) pomoću programskog paketa **ARPACK**. [11]

Neka je zadana matrica $A \in \mathbb{C}^{n\times n}$. Želimo naći skalar $\lambda \in \mathbb{C}^n$, $x \neq 0$, tako da vrijedi

$$Ax = \lambda x.$$

Takav λ zovemo *svojstvena vrijednost* matrice A, a x zovemo *svojstveni vektor* pridružen svojstvenoj vrijednosti λ. Ovaj problem nazivamo standardni problem svojstvenih vrijednosti. Generalizirani problem svojstvenih vrijednosti je:

$$Ax = \lambda Bx$$

ako je B regularna, možemo ga svesti na standardni problem na sljedeći način: $B^{-1}Ax = \lambda x$.

Arnoldijev algoritam

Definicija 0.0.3. *Neka je $A \in \mathbb{C}^{n\times n}$ i $b \in \mathbb{C}^n$. Krilovljeva matrica reda i je definirana s $K_i = K_i(A, b) = [b, Ab, \ldots, A^{i-1}b]$, a i-ti Krilovljev potprostor $\mathcal{K}_i$ je definiran kao ljuska od K_i, $\mathcal{K}_i = \mathcal{K}_i(A, b) = span\{b, Ab, \ldots, A^{i-1}b\}$.*

Dakle ako je $Ax = b$, onda je $x \in \mathcal{K}_n(A, b)$. Očekujemo da će već za $k << n$ u prostoru $\mathcal{K}_k(A, b)$ postojati dobre aproksimacije za x.

Definicija 0.0.4. *Kažemo da je $n \times n$ matrica H u Hessenbergovoj formi ako je $H_{ij} = 0$ za $i > j + 1$. H je strogo Hessenbergova ako je $H_{j+1,j} \neq 0$ za sve $j = 1, \ldots, n - 1$.*

Arnoldijev algoritam koristi modificiranu Gram-Schmidt metodu kako bi dobio niz ortonormiranih vektora takvih da ti vektori čine linearnu ljusku Krilovljevog potprostora $\mathcal{K}_n$. On također formira gornje Hessenbergovu matricu H_k, čije se svojstvene vrijednosti koriste za aproksimaciju podskupa svojstvenih vrijednosti početne matrice. Matrica H_k je ortogonalna projekcija matrice A na zadani Krilovljev potprostor te se njene svojstvene vrijednosti nazivaju *Ritzovim vrijednositma*. Eksplicitno, za matricu A, vektor b, i broj koraka k dajemo pseudokod za algoritam:

$v_1 = b/\|b\|$
for stupac j=1 to k **do**
  $r = Av_j$
  for i=1 to j **do**
    $h_{ij} = v_i^* r$
    $r = r - v_i h_{ij}$
  end for
  $h_{j+1,j} = \|r\|$
  if $h_{j+1,j} = 0$ **then**
    $l = j$
    $V = [v_1, \ldots, v_l]$
    $H = (h_{ij})_{(l+1)\times l}$
    STOP
  end if
  $v_{i+1} = \frac{r}{h_{j+1,j}}$
end for
$l = k$
$V = [v_1, \ldots, v_k]$
$H = (h_{i,j})_{k+1,k}$

U matričnom obliku to možemo zapisati kao:

$$AV_k = V_k H_k + f_k e_k^T$$

Implicitno restartani Arnoldijev algoritam

U implicitno restartanom Arnoldijevom algoritmu radimo više iteracija nego što je potrebno. Nakon k iteracija dobijemo k aproksimacija za svojstvene vrijednosti. Izdvojimo m svojstvenih vrijednosti koje su nam od interesa (u našem slučaju, to su one najbliže nekoj traženoj vrijednosti). Ostale svojstvene vrijednosti koristimo kako bi napravili pomake u

matrici H, te na taj način smanjili doprinos svojstvenih vrijednosti koje nam nisu od interesa u početnom vektoru. Arnoldijev algoritam ne pokrećemo ponovno od prvog koraka, već od m-tog, tj. izvršavamo ga implicitno.

Slijedi algoritam za Implicitno restartani Arnoldijev algoritam, s ulazom (A, V_m, H_m, f_m) dobivenim nakon m koraka Arnoldijevog algoritma te tolerancijom *tol* [16]:

for $i = 1, \ldots$ do konvergencije **do**
Izračunaj $\sigma(H_m)$ i odgovarajuće svojstvene vektore
if $\|f_m\| < tol$ **then**
break
end if
Odaberi $p = m - k$ pomaka $\left(\mu_1^{(m)}, \ldots, \mu_p^{(m)}\right)$ prema $\sigma(H_m)$
$q^T = e_m^T$
for $j = 1, 2, \ldots, p$ **do**
QR faktorizacija: $(Q_j, R_j) = qr(H_m - \mu_j^m I)$
$H_m = Q_j^* H_m Q_j$
$V_m = V_m Q_j$
$q^* = q^* Q_j$
end for
$f_k = v_{k+1} H_m(k+1, k) + f_m Q(m, k)$
$V_k = V_m(1 : n, 1 : k)$
$H_k = H_m(1 : k, 1 : k)$
S početkom u $AV_k = V_k H_k + f_k e_k^T$ Arnoldijevom faktorizacijom, primijeni još p koraka da bi se dobila nova Arnoldijeva faktorizacija od m koraka: $AV_m = V_m H_m + f_m e_m^T$
end for

Mjera kvalitete mreže i struktura ovog rada

Efikasan način za izgradnju kvalitetnih mreža za opisane CAD modele je da od inicijalnog modela izgradimo parametrizaciju koja omogućava ponovno generiranje mreže od spojenih isječaka.

Parametrizacija plohe će u našem slučaju biti funkcija koja preslika točke u domeni koja je podskup od $\mathbb{R}^2$ na neki podskup od $\mathbb{R}^3$. Različite parametrizacije očuvaju odnosno narušavaju različita svojstva početne mreže. Stoga, kako bi ih mogli uspoređivati definiramo sljedeće mjere kvalitete:

Kvalitetu oblika trokutnih mreža možemo evaluirati računanjem omjera slike za svaki trokut na sljedeći način: za trokut T je

$$\kappa_T = \alpha \frac{\text{unutarnji radius}}{\text{vanjski radius}} = 4 \frac{\sin\hat{\alpha} \sin\hat{\beta} \sin\hat{\gamma}}{\sin\hat{\alpha} + \sin\hat{\beta} + \sin\hat{\gamma}}$$

gdje su $\hat{\alpha}, \hat{\beta}, \hat{\gamma}$ tri unutarnja kuta trokuta T. S tom definicijom, optimalni jednakostranični trokuti imaju $\kappa = 1$ i ravni degenerirani trokuti imaju $\kappa = 0$. Promatramo prosječnu $\bar{\kappa}_T$ te minimalnu kvalitetu $\kappa_{\min}$. Neka je $\text{spr}(X)$ spektralni radijus matrice X, tada je $\|A\| = \sqrt{\text{spr}(A^*A)}$ spektralna norma matrice A. Može se pokazati da za matricu krutosti K pridruženu trijangulaciji $\mathcal{T}$ vrijedi ocjena $\|K\|\|K^{-1}\| \approx Ch^{-2}$. Ovdje je h vanjski radijus najmanjeg elementa teselacije $\mathcal{T}$ a konstanta C ovisi o najvećoj konstanti κ_T^{-1}. Modernu i puno detaljniju analizu možete pročitati u radu [10].

Istaknimo da je $\|K\|\|K^{-1}\|$ uvjetovanost linearnog sustava $Kx = f$. Uvjetovanost sustava mjeri osjetljivost rješenja na perturbacije ulaznih podataka. Za neki $\hat{x}$ vrijedi

$$\frac{\|x - \hat{x}\|}{\|x\|} \leq \|K\|\|K^{-1}\| \frac{\|K\hat{x} - f\|}{\|f\|} .$$

Sada je jasno da su matrice krutosti, pa samim time i problemi minimizacije Dirichletove energije, za one mreže koje imaju elemente s izrazito malim kutevima potencijalno loše uvjetovani. Korigiranjem mreže nastojimo prekondicionirati (smanjiti lošu uvjetovanost) problem, što je i motivacija za proučavanje algoritama koje ćemo prikazati – posebno sa osvrtom na njihovu realizaciju u **python** okruženju – u ovom radu.

Neka je l_i^h relativna, obzirom na dano polje željenih veličina elemenata, mjera veličine brida i. Mjera l_i^h je jedan kada je veličina brida točno onolika kolika je definirana poljem željenih dimenzija dikstretizacije (npr definiranih metričkim tenzorom parametrizacije plohe), za točnu definiciju ove metrike pogledajte [7]. Definiramo *indeks efikasnosti* τ [7] mreže kao eksponencijalnu vrijednost očekivane razlike između jedinice i l_i^h:

$$\tau[\%] = 100 \exp\left(\frac{1}{n_e} \sum_{i=1}^{n_e} d_i \right) .$$

Ovdje je $d_i = l_i^h - 1$ ako $l_i^h < 1$, $d_i = \frac{1}{l_i^h} - 1$ ako $l_i^h > 1$ i n_e je broj bridova u mreži. Optimalna mreža je mreža za koju je svaka veličina svakog brida l_i^h jednaka 1 tako da je indeks efikasnosti $\tau = 100\%$. Još mjerimo i brzinu izvršavanja svake parametrizacije za danu mrežu.

U prvom poglavlju uvodimo osnovne definicije iz diferencijalne geometrije te uvodimo baricentrične koordinate kao način parametrizacija koordinata točaka u unutrašnjosti danog trokuta.

U drugom poglavlju definiramo harmonijska preslikavanja, te uvodimo pojam Dirichletove energije koje ona minimiziraju. U trećem poglavlju uvodimo diskretno harmonijsko preslikavanje te 3 različite parametrizacije plohe s ograničenjima na rubu (takozvanim fiksnim rubom). Prikazujemo njihovu implementaciju u pythonu koristeći **libigl**, **numpy** i **scipy** biblioteke. U četvrtom poglavlju prezentiramo preslikavanja otvorenog ruba: konformno preslikavanje određeno u smislu najmanjih kvadrata te njegovu alternativu, spektralno

konformno preslikavanje. U petom poglavlju prezentiramo probleme vezane uz diskretna linearna preslikavanja, to jest problem osiguravanja injektivnosti samih preslikavanja. U šestom poglavlju za obrađene parametrizacije prikazujemo primjere te uspoređujemo diskretizacije s obzirom na gore definirane mjere uspjeha. Uzimamo 3 testna primjera s dovoljno velikim brojem vrhova te sa različitim brojem rubova kako bismo ilustrirali razliku u karakteristikama između parametrizacija. Konačno, u sedmom poglavlju nudimo smjernice za odabir parametrizacije ovisno o željenim karakteristikama rezultata te komentiramo moguće primjene i daljnji napredak u ovom području.

Poglavlje 1

Reprezentacija diskretnih ploha

U ovom poglavlju prezentiramo bitne rezultate iz diferencijalne geometrije vezane za parametrizaciju ploha.

Definicija 1.0.1. *Pretpostavimo da ploha $\mathcal{S} \subset \mathbb{R}^3$ ima parametarsku reprezentaciju:*

$$\mathbf{x}(u, v) = (x(u, v), y(u, v), z(u, v))$$

*za točke (u, v) u domeni $\mathbb{R}^2$. Kažemo da je takva reprezentacija **regularna** ako (i) su skalarne funkcije x, y, z glatke, tj. klase C^∞, i (ii), vektori:*

$$x_u = \frac{\partial x}{\partial u}, \quad x_v = \frac{\partial x}{\partial v}$$

linearno nezavisni u svakoj točki (njihov vektorski produkt $x_1 \times x_2$ nije nula).

Ako uzmemo u obzir ortonormiranu bazu $\left\{\overrightarrow{i}, \overrightarrow{j}, \overrightarrow{k}\right\}$ vektorskog prostora $\mathbb{R}^3$ tad se vektorska funkcija $\overrightarrow{x} = \overrightarrow{x}(u, v)$ zadaje s tri skalarne funkcije: $x(u, v)$, $y(u, v)$, i $z(u, v)$ izrazom:

$$\overrightarrow{x}(u, v) = x(u, v) \cdot \overrightarrow{i} + y(u, v) \cdot \overrightarrow{j} + z(u, v) \cdot \overrightarrow{k}$$

tu jednadžbu nazivamo **vektorskom jednadžbom plohe** $\mathcal{S} \subset \mathbb{R}^3$.

Varijable u i v zovemo **parametrima**, a uv-ravninu zovemo **parametarskom ravninom**.

Ako fiksiramo parametar v, tako da je $v = v_0 = konst.$, onda iz vektorske jednadžbe plohe dobijemo jednadžbu krivulje:

$$C_u \quad \ldots \quad \overrightarrow{x}(u, v_0) = x(u, v_0) \cdot \overrightarrow{i} + y(u, v_0) \cdot \overrightarrow{j} + z(u, v_0) \cdot \overrightarrow{k} = (x(u, v_0), y(u, v_0), z(u, v_0))$$

koju nazivamo u-**krivuljom** te analogno definiramo v-**krivulju**.

Ako variramo vrijednost u_0, dobijemo familiju u-krivulja koja zajedno s familijom v-krivulja čini **parametarsku ili koordinatnu mrežu**.

Pretpostavimo sada da se ploha $\mathcal{S}$ nalazi u parametarskoj ravnini, te neka je f preslikavanje iz $\mathcal{S}$ na drugu plohu $\mathcal{S}'$. Tada definiramo parametrizaciju $\mathbf{x}^* = f \circ \mathbf{x}$ od $\mathcal{S}'$ takvu da su koordinate $f(\mathbf{p}) \in \mathcal{S}'$ iste kao i odgovarajućoj praslici točke $\mathbf{p} \in \mathcal{S}$.

Definicija 1.0.2. *Kažemo da je preslikavanje f **dopustivo** ako je parametrizacija $\mathbf{x}^*$ regularna.*

Definicija 1.0.3. *Dopustivo preslikavanje iz $\mathcal{S}$ u $\mathcal{S}'$ je **izometrično** ako je duljina bilo kojeg luka na $\mathcal{S}^*$ jednaka kao i njegova praslika na $\mathcal{S}$. Takvo preslikavanje se zove **izometrija**.*

Dvije površine su izometrične ako postoji izometrija između njih.

Definicija 1.0.4. *Dopustivo preslikavanje iz $\mathcal{S}$ u $\mathcal{S}'$ je **konformno** (čuva kuteve) ako je kut presjeka svakog para lukova koji se sijeku na $\mathcal{S}^*$ jednak odgovarajućim praslikama na $\mathcal{S}$ u odgovarajućoj točki.*

1.1 Parametrizacija diskretnih ploha

U našim primjerima počinjemo s $3D$ plohom te je želimo preslikati na ravninu:

$$\mathbf{x} \in \mathcal{S} \subset \mathbb{R}^3 \mapsto \mathbf{u}(\mathbf{x}) \in \mathcal{S}' \in \mathbb{R}^2$$

Takva neprekidna parametrizacija postoji ako površine $\mathcal{S}$ i $\mathcal{S}'$ imaju istu topologiju, tj. imaju isti genus i isti broj rubova N_B.

Definicija 1.1.1. ***Genus** povezane orijentabilne plohe je maksimalan moguć broj rezova duž jednostavnih zatvorenih krivulja bez narušavanja povezanosti plohe. [8]*

Intuitivno, genus plohe je broj "rupa" na plohi. Na primjer, sfera ima genus 0 i $N_B = 1$, a torus ima genus 1 i $N_B = 0$. Ovdje smatramo da je jedina dostupna reprezentacija površine $\mathcal{S}$ mreža $\mathcal{S}_\mathcal{T}$ sačinjena od trokuta, tj. unija skupa trokuta T_j koja se siječe samo na zajedničkim bridovima i vrhovima $\mathcal{T} = \{T_1, \ldots, T_N\}$. Promatramo trianguliranu površinu $\mathcal{S}$ koja ima N_V vrhova, N_E bridova, i N_T trokuta. Genus $G(\mathcal{S}_\mathcal{T})$ je dan Euler-Poincaré formulom:

$$G(\mathcal{S}_\mathcal{T}) = \frac{-N_V + N_E - N_T + 2 - N_B}{2}$$

Promatramo diskretne parametrizacije: Svaki vrh V_i, $i = 1, \ldots, N_V$ triangulacije ima dva skupa koordinata: $3D$ koordinate $\mathbf{x}_i = (x_i, y_i, z_i) \in \mathcal{S}$ i parametarske koordinate $\mathbf{u}_i =$

$(u_i, v_i) \in \mathcal{S}'$. Svaki trokut ima dvije reprezentacije: Jedan u $3D$ prostoru i jednu u $2D$ parametarskom prostoru. Promotrimo trokut T_j s tri vrha V_1, V_2, V_3. Parametrizacija je bijektivna akko se trokuti ne preklapaju u parametarskom prostoru.

Sam trokut može bit parametriziran koristeći npr. baricentrične koordinate:

$$x(\xi) = (1 - \xi - \eta)x_1 + \xi x_2 + \eta x_3 \tag{1.1}$$

Slično, možemo parametrizirati trokut u $\mathcal{S}'$:

$$\mathbf{u}(\xi) = (1 - \xi - \eta)\mathbf{u}_1 + \xi \mathbf{u}_2 + \eta \mathbf{u}_3 \tag{1.2}$$

Kako bi našli preslikavanje $x(\mathbf{u})$, invertiramo:

$$\mathbf{u} - \mathbf{u}_1 = \underbrace{\begin{bmatrix} u_2 - u_1 & u_3 - u_1 \\ v_2 - v_1 & v_3 - v_1 \end{bmatrix}}_{\mathbf{u}_{,\xi}} \underbrace{\begin{pmatrix} \xi \\ \eta \end{pmatrix}}_{\xi}$$

što daje:

$$\xi(\mathbf{u}) = (\mathbf{u}_{,\xi})^{-1}(\mathbf{u} - \mathbf{u}_1) = (\xi_{,\mathbf{u}})(\mathbf{u} - \mathbf{u}_1) \tag{1.3}$$

Dakle, diskretno preslikavanje $\mathbf{x}(\mathbf{u})$ možemo izračunati u 3 koraka:

1. Nađemo jedinstveni trokut T_j parametarskog prostora $\mathcal{S}'$ koji sadrži točku $\mathbf{u}$
2. Izračunamo lokalne koordinate $\xi = (\xi, \eta)$ točke u unutar trokuta T_j koristeći jednadžbu 1.3
3. Iskoristimo jednadžbu 1.1 da izračunamo preslikavanje $\mathbf{x}(\mathbf{u}) = x(\xi(u))$

Procedure generiranja mreže često traže i derivaciju $\mathbf{x}_{,\mathbf{u}}$. Imamo:

$$\mathbf{x}_{,\mathbf{u}} = \mathbf{x}_{,\xi}\xi_{,\mathbf{u}} = \begin{bmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \\ z_2 - z_1 & z_3 - z_1 \end{bmatrix} \frac{\begin{bmatrix} v_3 - v_1 & -(u_3 - u_1) \\ -(v_2 - v_1) & u_2 - u_1 \end{bmatrix}}{(u_2 - u_1)(v_3 - v_1) - (v_2 - v_1)(u_3 - u_1)}$$

Metrički tenzor (ili prva fundamentalna forma)

$$M = \mathbf{x}_{,\mathbf{u}}{}^T \mathbf{x}_{,\mathbf{u}}$$

tada može računati duljine, kuteve i površine. Promotrimo krivulju C na parametarskom prostoru $\mathcal{S}'$. Njena duljina je:

$$l_C = \int_C dl = \int_C \sqrt{\mathrm{d}x^2} = \int_C \sqrt{(\mathbf{x}_{,\mathbf{u}}\mathrm{d}u)^2} = \int_C \sqrt{\mathrm{d}u^T M(u)\,\mathrm{d}u}$$

U praksi za generiranje mreže: neka je C brid mreže parametarskog prostora između $\mathbf{u_1}$ i $\mathbf{u_2}$. Ako označimo s $e = \mathbf{u_2} - \mathbf{u_1}$, njegova parametrizacija je:

$$C = \{\mathbf{u} \in \mathcal{S} \,|\, \mathbf{u} = \mathbf{u}_1 + te, t \in [0, 1]\}.$$

Poglavlje 2

Harmonijska preslikavanja

2.1 Konformna i harmonijska preslikavanja

Konformna preslikavanja imaju mnoga dobra svojstva, te su povezana s teorijom kompleksnih funkcija. Promotrimo slučaj preslikavanja iz područja $\mathcal{S}$ na ravninu. Takvo preslikavanje možemo gledati kao funkciju kompleksne varijable, $w = f(z)$. Lokalno, konformno preslikavanje je funkcija f koja je analitička oko točke z takva da je $f'(z) \neq 0$. Konformno preslikavanje stoga zadovoljava Cauchy-Riemannove jednadžbe, koje za $z = x + iy$, $w = u + iv$ glase:

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} \quad \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x}$$

Ako sada diferenciramo prvu jednadžbu po x, a drugu po y, dobijemo dvije Laplace-ove jednadžbe:

$$\Delta u = 0, \quad \Delta v = 0.$$

Ovdje smo sa

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

označili Laplaceov operator. To nas dovodi do definicije harmonijskog preslkavanja:

Definicija 2.1.1. *Funkciju* $\mathbf{u} \in C^2(\mathcal{S})$ *koja rješava Laplaceovu jednadžbu* $-\Delta\mathbf{u} = 0$ *nazivamo* ***harmonijska funkcija (harmonijsko preslikavanje)****.*

Stoga je konformno preslikavanje također harmonijsko, te imamo sljedeći niz implikacija: [5]

$$\textbf{izometrijsko} \Longrightarrow \textbf{konformno} \Longrightarrow \textbf{harmonijsko}$$

Velika prednost harmonijskih preslikavanja od konformnih je lakoća izračuna aproksimacija. Nakon što odaberemo odgovarajuće preslikavanje ruba (što je ekvivalentno odabiru Dirichletovih rubnih uvjeta za u i v), svaka funkcija u i v je rješenje linearne eliptičke parcijalne diferencijalne jednadžbe koja se može aproksimirati različitim metodama, kao što je metoda konačnih elemenata ili konačne razlike. Obje te metode dovode do linearnog sustava jednadžbi. Harmonijska preslikavanja su također bijektivna za konveksna područja, što nas dovodi do sljedećeg teorema:

Teorem 2.1.2 (Rado-Kneser-Choquet)**.** *Ako je $f : \mathcal{S} \to \mathbb{R}^2$ harmonijsko preslikavanje i preslika rub $\partial\mathcal{S}$ homeomorfno na rub $\partial\mathcal{S}'$ neke konveksne površine $\mathcal{S}' \subset \mathbb{R}^2$, tada je f injektivna.*

Dokaz. Skica. Glavna ideja dokaza je prvo pokazati da je f lokalno injektivna. Globalna injektivnost se tada dobije iz lokalne pozivanjem na princip argumenta za harmonijske funkcije. Za lokalnu injektivnost je dovoljno pokazati da Jakobijan J_f nije nula za svaku točku u $\mathcal{S}$. Cijeli dokaz teorema može se naći u [4], str. 30., dok je princip argumenta za harmonijske funkcije prikazan na 9. stranici. □

S druge strane, harmonijska preslikavanja nisu uvijek konformna te ne očuvaju kuteve. Na primjer, lako možemo provjeriti iz Cauchy-Riemannovih i Laplaceovih jednadžbi da je bilinearno preslikavanje $f : [0,1]^2 \longrightarrow \mathbb{R}^2$ definirano s

$$u = x(1+y) \quad v = y$$

harmonijsko ali ne konformno.

Još jedna mana harmonijskih preslikavanja je "jednostranost". Inverz harmonijskog preslikavanja nije nužno harmonijsko. To možemo vidjeti u prošlom primjeru: inverzno preslikavanje $x = u/(1+v)$, $y = v$ nije harmonijsko jer funkcija $x(u,v)$ ne zadovoljava Laplaceovu jednadžbu. Usprkos tomu, harmonijska preslikavanja minimiziraju deformaciju u smislu da minimiziraju Dirichletovu energiju[2]:

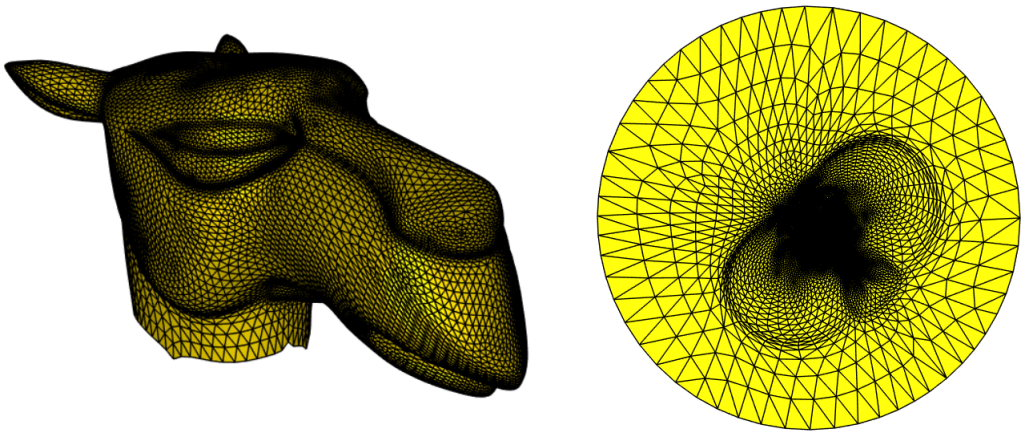
Definicija 2.1.3. *Za dani otvoren skup $\mathcal{S} \subset \mathbb{R}^n$ i funkciju $\mathbf{u} : \mathcal{S} \mapsto \mathbb{R}$,* ***Dirichletova energija*** *je realni broj:*

$$E_D(\mathbf{u}) = \int_{\mathcal{S}} \frac{1}{2} |\nabla \mathbf{u}|^2 \mathrm{d}s \tag{2.1}$$

gdje je $\nabla\mathbf{u} : \mathcal{S} \mapsto \mathbb{R}^n$ gradijent funkcije $\mathbf{u}$, a $|\cdot|$ označava Euklidsku ili Frobeniusovu normu na prostoru matrica.

U sljedećem poglavlju, pokazat ćemo kako izračunati parametrizacije u diskretnom slučaju za danu ulaznu trokutastu mrežu.

2.2 Konstrukcija harmonijskog preslikavanja



Slika 2.1: Parametrizacija glave deve $\mathcal{S}$ na jediničnu kružnicu $\mathcal{S}'$

Harmonijsko preslikavanje minimizira distorziju u smislu da minimizira Dirichlet-ovu energiju preslikavanja $\mathbf{u}(\mathbf{x})$ uz Dirichlet-ove rubne uvjete $\mathbf{u} = \mathbf{u}_D$ na jednom od N-zatvorenih rubova na plohi. Taj rub označavamo s $\partial\mathcal{S}^0$.

Kao što je ilustrirano na slici 2.1, odlučili smo preslikati plohu $\mathcal{S}$ u $3D$ prostoru na jediničnu kružnicu $\mathcal{S}'$. Stoga trebamo plohe genusa 0 koje imaju bar jedan rub koji će biti preslikan na jediničnu kružnicu.

Jaka formulacija koja odgovara ovom problemu minimizacije je dana s:

$$\nabla^2 u = 0 \quad \nabla^2 v = 0 \quad \text{na } \mathcal{S} \tag{2.2}$$

s odgovarajućim Dirichlet-ovim uvjetima na rubu $\partial\mathcal{S}^0$ od $\mathcal{S}$:

$$\mathbf{u} = \mathbf{u}_D(\mathbf{x}), \quad \text{na } \partial\mathcal{S}^0.$$

gdje su u i v komponentne funkcije $\mathbf{u}$. To znači da ako možemo numerički izračunati rješenje tih dviju parcijalnih diferencijalnih jednadžbi 2.2, rješenja $u(x)$ i $v(x)$ definiraju diskretno harmonijsko preslikavanje $\mathbf{u}(\mathbf{x})$.

Poglavlje 3

Aproksimacija harmonijskog preslikavanja fiskiranog ruba

Ono što je zajedničko svim metodama parametrizacije plohe je aproksimacija glatke plohe $\mathcal{S}$ po dijelovima linearnom plohom $\mathcal{S}_\mathcal{T}$ u obliku trokutaste mreže, tj. unije skupa $\mathcal{T} = \{T_1, \ldots, T_M\}$ trokuta takvih da je presjek svaka dva para trokuta $T_i \neq T_j$ prazan skup, zajednički vrh, ili zajednički brid. S V ćemo označiti skup vrhova. Ako $\mathcal{S}_\mathcal{T}$ ima rub, on će biti poligonalan te ćemo s V_B označiti skup vrhova koji leže na rubu a s V_I skup unutarnjih vrhova. S E označavamo skup bridova u $\mathcal{T}$.

Izračunati parametrizaciju znači izračunati funkciju koja preslikava parametarsku uv ravninu u danu plohu $\mathcal{S} \subset \mathbb{R}^3$. Budući da radimo s trokutastom mrežom $\mathcal{S}_\mathcal{T}$, cilj je pronaći odgovarajuću domenu $\mathcal{S}' \subset \mathbb{R}^2$ te po dijelovima linearno preslikavanje $\mathbf{u} : \mathcal{S}_\mathcal{T} \mapsto \mathcal{S}'$ koje je linearno na svakom trokutu T_i u $\mathcal{S}_\mathcal{T}$ te neprekidno. Takvo preslikavanje je jedinstveno određeno slikama vrhova u $\mathcal{S}'$.

3.1 Diskretna aproksimacija harmonijskog preslikavanja

Problem računanja parametrizacije plohe možemo modelirati kao problem minimalizacije Dirichletove energije

$$\min_{\substack{\mathbf{u} \\ \mathbf{u}(\mathcal{I})=\mathcal{I}'}} \int_{\mathcal{S}_\mathcal{T}} \|\nabla u\|^2 + \|\nabla v\|^2 \, dA \ .$$

po prostoru svih funkcija $\mathbf{u}$, $\mathbf{u} = (u, v) : \mathcal{S}_\mathcal{T} \mapsto \mathcal{S}'$ koje preslikavaju skup točaka $\mathcal{I} \subset \partial\mathcal{S}_\mathcal{T}$ u skup točaka $\mathcal{I}' \subset \partial\mathcal{S}'$. Ovdje su u i v komponente parametrizacije, a $\mathcal{I}$ i $\mathcal{I}'$ su prikladno izabrani skupovi točaka u kojima implementiramo ograničenja na rubu. Ovaj problem ćemo efektivno računati tako da ćemo se ograničiti na konačno dimenzionalan prostor funkcija na kojem se Dirichletova energija može realizirati u matričnom obliku. Problem

minimizacije uz ograničenja ćemo zapisati korištenjem Lagrangeovih multiplikatora, što će nam omogućiti da problem traženja stacionarnih točaka svedemo na problem rješavanja nelinearnog sustava jednadžbi (2).

Neka je u po dijelovima linearna skalarna funkcija i neka su φ_i funkcije oblika pridružene čvoru i. Vrijednost funkcije u u vrhu i ćemo označavati s u_i. Tada funkciju u možemo reprezentirati, kao što je prikazano u jednadžbi (1), kao sumu

$$u(x) = \sum_{i=1}^{n} u_i \varphi_i(x) \, .$$

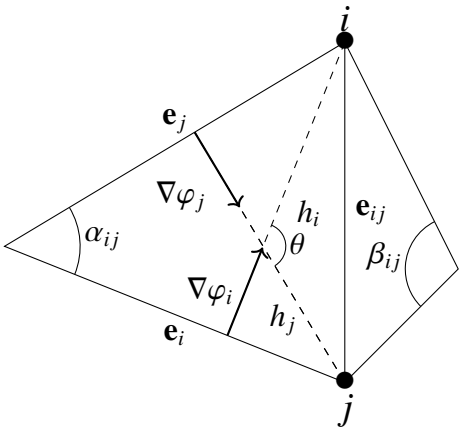
Ako uvrstimo takvu reprezentaciju funkcije u u definiciju Dirichletove energije, dobijemo diskretnu energiju koja je kvadratna funkcija u vrijednostima u_i:

$$\begin{aligned}
\int_{\mathcal{S}_\mathcal{T}} \|\nabla u\|^2 \, dA &= \int_{\mathcal{S}_\mathcal{T}} \left\| \nabla \left(\sum_{i=1}^{n} u_i \varphi_i(x) \right) \right\|^2 dA \\
&= \int_{\mathcal{S}_\mathcal{T}} \left(\sum_{i=1}^{n} u_i \nabla \varphi_i(x) \right) \cdot \left(\sum_{i=1}^{n} u_i \nabla \varphi_i(x) \right) dA \\
&= \int_{\mathcal{S}_\mathcal{T}} \left(\sum_{i=1}^{n} u_i \nabla \varphi_i(x) \right) \cdot \left(\sum_{i=1}^{n} u_i \nabla \varphi_i(x) \right) dA \\
&= \int_{\mathcal{S}_\mathcal{T}} \sum_{i=1}^{n} \sum_{j=1}^{n} \nabla \varphi_i(x) \cdot \nabla \varphi_j(x) u_i u_j \, dA \\
&= \sum_{i=1}^{n} \sum_{j=1}^{n} u_i u_j \int_{\mathcal{S}_\mathcal{T}} \nabla \varphi_i(x) \cdot \nabla \varphi_j(x) \, dA \\
&= \mathbf{u}^T L \mathbf{u} \, .
\end{aligned}$$

Ovdje je $L_{ij} = \int_S \nabla \varphi_i(x) \cdot \nabla \varphi_j(x) \, dA$, a $\mathbf{u}$ označava vektor-stupac koji sadrži kao i-tu komponentu vrijednosti funkcije u u vrhu i. Neka je sada $\mathbf{v}$ analogan zapis funkcije v. Minimizacija Dirichletove energije po prostoru po djelovima linearnih funkcija se svodi na minimizaciju zbroja $\mathbf{u}^T L \mathbf{u} + \mathbf{v}^T L \mathbf{v}$.

S ovako definiranim baznim funkcijama φ_i, vrijednosti u matrici sustava L_{ij} su jedinstveno određene geometrijom mreže. Te vrijednosti su poznate kao *kotangens težine*. Funkcija *kotangens* zbog njenih trigonometrijskih formula, a *težine* jer matrica L definira Laplacijan težinskog grafa dane mreže.

Izračunajmo još vrijednosti L_{ij}. Prvo primijetimo da je vrijednost $\nabla \varphi_i$ konstantna na svakom trokutu, te nije 0 samo na trokutima incidentnim vrhu i. Za takav trokut T_α, $\nabla \varphi_i$ je



Slika 3.1: Kutevi i vrhovi unutar trokuta T_α i T_β

okomit nasuprotnom bridu e_i s vrijednosti recipročnom visini h, gdje je h visina trokuta s nožištem u e_i:

$$\|\nabla\varphi_i\| = \frac{1}{h} = \frac{\|\mathbf{e}_i\|}{2A}$$

gdje je $\mathbf{e}_i$ vektorski zapis brida e_i, te A površina trokuta.

Promotrimo sad dva susjedna vrha i i j, povezanih bridom $\mathbf{e}_{ij}$. Tada je $\nabla\varphi_i$ okomit na $\mathbf{e}_i$ te analogno $\nabla\varphi_j$ okomit na $\mathbf{e}_j$. Neka je kut između ta dva vektora θ. Tada vrijedi:

$$\nabla\varphi_i(x) \cdot \nabla\varphi_j(x) = \|\nabla\varphi_i(x)\| \left\|\nabla\varphi_j(x)\right\| \cos\theta = \frac{\|\mathbf{e}_i\|}{2A}\frac{\|\mathbf{e}_i\|}{2A}\cos\theta$$

Sada primijetimo da je kut između $\mathbf{e}_i$ i $\mathbf{e}_j$, nazovimo ga α_{ij} (slika 3.1) jednak $(\pi - \theta)$, ali i da vrijedi:

$$\cos\theta = -\cos(\pi - \theta) = -\cos(\alpha_{ij})$$

Sad možemo gornju jednadžbu zapisati u obliku:

$$\frac{\|\mathbf{e}_i\|}{2A}\frac{\|\mathbf{e}_i\|}{2A}\cos\alpha_{ij}$$

Ako primijenimo definiciju sinusa za pravokutne trokute:

$$\sin\alpha_{ij} = \frac{h_j}{\|\mathbf{e}_i\|} = \frac{h_i}{\left\|\mathbf{e}_j\right\|}$$

Primijenimo na jednadžbu:

$$\frac{\frac{h_j}{\sin\alpha_{ij}}}{2A}\frac{\left\|\mathbf{e}_j\right\|}{2A}\cos\alpha_{ij}$$

Pošto $\left\|\mathbf{e}_j\right\| h_j = 2A$, dobijemo:

$$\nabla\varphi_i(x) \cdot \nabla\varphi_j(x) = -\frac{\operatorname{ctg}\alpha_{ij}}{2A}$$

Analogno za trokut T_β:

$$\nabla\varphi_i(x) \cdot \nabla\varphi_j(x) = -\frac{\operatorname{ctg}\beta_{ij}}{2B}$$

Kako su φ_i i φ_j različite od nule samo unutar ta dva trokuta, dobijemo:

$$\int_S \nabla\varphi_i(x)\cdot\nabla\varphi_j(x)dA = A\left(\nabla\varphi_i(x)\cdot\nabla\varphi_j(x)\right)\Big|_{T_\alpha} + B\left(\nabla\varphi_i(x)\cdot\nabla\varphi_j(x)\right)\Big|_{T_\beta} = -\frac{1}{2}(\operatorname{ctg}\alpha_{ij}+\operatorname{ctg}\beta_{ij})$$

Preslikavanje ruba mreže na jediničnu kružnicu

Ograničenja u minimizacijskom problemu su dana s $\mathbf{u}(\mathcal{I}) = \mathcal{I}'$ gdje je $\mathcal{I} \subset \partial\mathcal{S}_\mathcal{T}$ i $\mathcal{I}' \subset \partial\mathcal{S}'$. Želimo definirati preslikavanje koje će vrhove iz $\mathcal{I}$ preslikati na $\partial K(0,1) = \partial\mathcal{S}'$.

Rub triangulacije je po dijelovima linearna zatvorena krivulja. Neka je L duljina te krivulje. Definiramo $\mathcal{I} := \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ gdje su $\mathbf{x}_1, \ldots, \mathbf{x}_n$ vrhovi triangulacije na toj krivulji t.d. vrijedi $\{\mathbf{x}_i, \mathbf{x}_{i+1}\} \in E$ za $i = 1, \ldots n-1$ te je također $\{\mathbf{x}_n, \mathbf{x}_1\} \in E$.

Udaljenost vrha $\mathbf{x}_i$ od vrha $\mathbf{x}_1$ duž ruba $\partial\mathcal{S}_\mathcal{T}$ možemo računati kao:

$$l(\mathbf{x}_i) = \begin{cases} 0, & \text{i=1} \\ \|\mathbf{x}_i - \mathbf{x}_{i-1}\| + l(\mathbf{x}_{i-1}), & \text{inače} \end{cases}$$

Konačno, definiramo preslikavanje koje će preslikati točke ruba na jediničnu kružnicu:

$$\mathbf{u}(\mathbf{x}_i) = \left(\cos\left(\frac{2\pi l(\mathbf{x}_i)}{L}\right), \sin\left(\frac{2\pi l(\mathbf{x}_i)}{L}\right)\right)$$

Napomena 3.1.1. *U svim preslikavanjima u ovom poglavju, potrebno je fiksirati rub plohe na neki konveksan poligon. To činimo na način kako je ovdje opisano te stoga ta preslikavanja zovemo* ***preslikavanjima fiksiranog ruba****.*

Implementacija

Slijedi isječak kôda za izračun diskretnog harmonijskog preslikavanja. Svaka mreža je definirana listom vrhova i naličja (trokuta) te ju učitamo:

```
V, F  = igl.read_triangle_mesh(os.path.join(root_folder, "camelhead.off"
    ))
```

Preslikamo rub na krug na prije opisan način:

```
b = igl.boundary_loop(F)
bc = igl.map_vertices_to_circle(V,b)
```

Rješavanje problema minimizacije

Za matricu $U \in \mathbb{R}^{n\times 2}$ promatramo problem minimizacije:

$$\min_U \frac{1}{2} \operatorname{tr}\left(U^T L U\right)$$

gdje je $L \in \mathbb{R}^{n\times n}$ rijetko popunjena matrica s vrijednostima:

$$L_{ij} = \begin{cases} w_{ij} & \text{ako } i \neq j \text{ i } \exists\{i,j\} \in E, \\ -\sum_{\ell \neq i} L_{i\ell} & \text{ako } i = j \\ 0 & \text{inače} \end{cases}$$

Primijetimo sada da točke u uv-parametarskom prostoru možemo reprezenzirati matricom $U \in \mathbb{R}^{n\times 2}$, gdje je u i-tom retku prvi stupac u koordinata i-te točke, a drugi stupac v koordinata. Dirichletova energija sada može biti zapisana kao trag kvadratne forme. To efektivno znači da primjenjujemo Q na svaki stupac od U zasebno i sumiramo rezultat:

$$\begin{aligned} \operatorname{tr}\left(U^T Q U\right) &= \operatorname{tr}\left(\begin{bmatrix}\mathbf{u}^T \\ \mathbf{v}^T\end{bmatrix} Q \begin{bmatrix}\mathbf{u} & \mathbf{v}\end{bmatrix}\right) \\ &= \operatorname{tr}\left(\begin{bmatrix}\mathbf{u}^T Q\mathbf{u} & \mathbf{u}^T Q\mathbf{v} \\ \mathbf{v}^T Q\mathbf{u} & \mathbf{v}^T Q\mathbf{v}\end{bmatrix}\right) \\ &= \mathbf{u}^T Q\mathbf{u} + \mathbf{v}^T Q\mathbf{v} \end{aligned}$$

Zapis energije u obliku traga ima prednost u tome što imamo isti Q za svaki stupac. Za minimizaciju kvadratne energije, to znači da se rješavanje može paralelizirati (Single instruction, multiple data). Također, dio izračuna kao što je faktorizacija matrice Q možemo napraviti prije samog rješavanja, i to samo jednom.

U **libigl**, minimizaciju energije

$$\min_{\mathbf{z}} \frac{1}{2}\mathbf{z}^T Q\mathbf{z} + \mathbf{z}^T B + C,\ C \in \mathbb{R}$$

na mreži s ograničenjima

$$A_{eq}\mathbf{z} = B_{eq}$$

i fiksiranim vrijednostima $\mathbf{z}_{bc}$ na indeksima $\mathbf{z}_b$ rješava funkcija `min_quad_with_fixed` i to metodom Lagrangeovih multiplikatora [9]. U našem slučaju, imamo samo fiksirane vrijednosti te nam je ova specifikacija preopćenita. Stoga postavljamo B, A_{eq} i B_{eq} na nulu.

```
B=np.zeros((V.shape[0],2))
Aeq = sp.sparse.csc_matrix(np.zeros((1,V.shape[0])))
Beq = np.zeros((1,2))
igl.min_quad_with_fixed(Q, B, b, bc, Aeq, Beq, False)
```

Min quad with fixed

`Min_quad_with_fixed` minimizira kvadratnu formu metodom Lagrangeovih multiplikatora. Ta metoda dodaje varijablu za svako linearno ograničenje (općenito, $m \times 1$ vektor varijabla λ) te rješava problem pronalaska sedlaste točke. U matričnom obliku to izgleda ovako:

$$\mathcal{L}(\mathbf{z}, \lambda) = \frac{1}{2}\mathbf{z}^T Q\mathbf{z} + \mathbf{z}^T B + C + \lambda^T(A_{eq}\mathbf{z} - B_{eq}).$$

Stacionarnu točku ove Lagrangeove funkcije za optimizaciju kvadratične funkcije uz linearna ograničenja računamo rješavanjem sustava linearnih jednadžbi [15]

$$\begin{bmatrix} Q & A_{eq}^T \\ A_{eq} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \lambda \end{bmatrix} = \begin{bmatrix} -C \\ B_{eq} \end{bmatrix}.$$

Ako uvrstimo $A_{eq} = 0$ i $B_{eq} = 0$ problem se svodi na rješavanje linearnog sustava $Q\mathbf{z} = 0$. Također, budući da su težine w_{ij} unutar matrice Q zbroj kotangensa u nasuprotnim vrhovima oba trokuta koji sadrže brid $\{i, j\}$, očito vrijedi $w_{ij} = w_{ji}$. Stoga je Q simetrična matrica.

Rješavanje sustava $Q\mathbf{z} = 0$

Tražimo rješenje sustava $Q\mathbf{z} = 0$, gdje je Q $n \times n$ diskretni Laplacijan, te $\mathbf{z}$ vektor u ili v koordinata točke. Dimenzija $\mathbf{z}$ odgovara broju vrhova te je njihova vrijednost nepoznata, ali neke vrijednosti predstavljaju vrijednosti na vrhovima ruba. Te vrijednosti znamo te ih možemo prebaciti na desnu stranu. Konceptualno, to je veoma lako ako prije pivotiramo tako da unutarnji vrhovi $\mathbf{z}$ dođu na početak, a vrhovi ruba na kraj vektora $\mathbf{z}$. Neka je $\mathbf{z}_i$ vektor koji sadrži unutarnje vrhove, a $\mathbf{z}_b$ vektor koji sadrži vrhove ruba:

$$\begin{pmatrix} Q_{i,i} & Q_{i,b} \\ Q_{b,i} & Q_{b,b} \end{pmatrix} \begin{pmatrix} \mathbf{z}_i \\ \mathbf{z}_b \end{pmatrix} = \begin{pmatrix} 0_i \\ \mathbf{z}_{bc} \end{pmatrix}$$

Donji blok jednadžbi nam je sad nebitan te ga možemo izbaciti:

$$\begin{pmatrix} Q_{i,i} & Q_{i,b} \end{pmatrix} \begin{pmatrix} \mathbf{z}_i \\ \mathbf{z}_b \end{pmatrix} = 0_i$$

Konačno, možemo prebaciti poznate vrijednosti na desnu stranu i dobiti sustav koji rješavamo:

$$Q_{i,i}\mathbf{z}_i = -Q_{i,b}\mathbf{z}_b$$

Matricu $Q_{i,i}$ `min_quad_with_fixed` faktorizira *LU* faktorizacijom. Elemente $Q_{i,i}$ laplacijana za harmonijsko preslikavanje smo već izračunali:

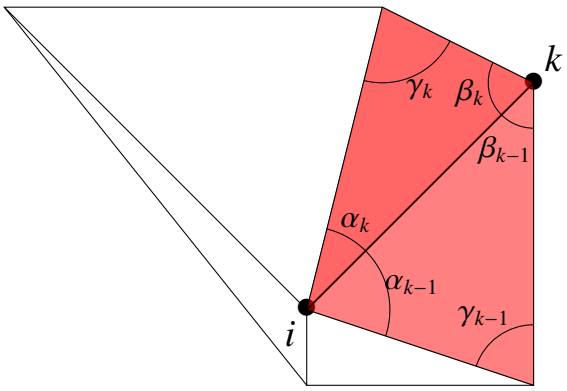
$$Q_{ij} = \begin{cases} -\sum_k Q_{ik}, & i = j \\ -\frac{1}{2}(\operatorname{ctg}\gamma_{j-1} + \operatorname{ctg}\gamma_j), & j \text{ incidentan s } i \\ 0, & \text{inače} \end{cases} \tag{3.1}$$

budući da se kotangens laplacijan često javlja u raznim područjima računalne grafike, **libigl** ima već definiranu funkciju za njegov izračun:

```
1 L = igl.cotmatrix(V,F)
```

Primjer harmonijskog preslikavanja smo vidjeli na slici 2.1.

3.2 Preslikavanje konveksnih kombinacija vrhova



Slika 3.2: Kutevi i vrhovi koje je potrebno evaluirati za izračun konveksnog preslikavanja

U ovom poglavlju ćemo se fokusirati na svojstva konveksnih kombinacija diskretnih (po dijelovima linearnih) harmonijskih preslikavanja.

Zbog RKC teorema 2.1.2, razumno je očekivati da će, ako je $\mathcal{S}'$ konveksan, diskretno harmonijsko preslikavanje $\mathbf{u} : \mathcal{S}_\mathcal{T} \mapsto \mathcal{S}'$ biti bijektivno kao i u neprekidnom slučaju, tj. za svaki orijentirani trokut $T = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3]$ iz $\mathcal{S}_\mathcal{T}$, slika trokuta $\mathbf{u}(T) = [\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3]$ neće biti degenerirani trokut i imat će istu orijentaciju.

Pokazuje se da ova tvrdnja vrijedi ako su sume kotangensa $\operatorname{ctg}\gamma_{k-1} + \operatorname{ctg}\gamma_k$ pozitivne. Kako bismo ovu tvrdnju ilustrirali, promatramo preslikavanje

$$\mathbf{u}_i = \sum_{k=1}^{d_i} \lambda_{ik}\mathbf{u}_k \tag{3.2}$$

s koeficijentima λ_{ik} koji su zadani formulom

$$\lambda_{ik} = \frac{w_{ik}}{\sum_{j=1}^{d_i} w_{ij}} . \tag{3.3}$$

Koeficijenti w_{ij} se nazivaju težine konveksnog preslikavanja. Slijedi da ako su sve težine w_{ij} pozitivne, tada su i težine λ_{ij} pozitivne, te je slika svakog unutarnjeg vrha $\mathbf{u}_i$ konveksna kombinacija svojih susjeda $\mathbf{u}_k$ te stoga mora biti u njihovoj konveksnoj ljusci. Stoga zovemo bilo koje po dijelovima linearno preslikavanje ovakvog tipa *preslikavanje konveksnih kombinacija*. Poseban slučaj u kojem su težine λ_{ij} uniformne, tj. za svaki unutarnji vrh $\mathbf{x}_i$ su jednake $1/d_i$ gdje je d_i stupanj vrha $\mathbf{x}_i$, zove se Tutte-ovo *baricentričko preslikavanje* [19]. Svaka slika vrha $\mathbf{u}_i$ je baricentar svojih susjeda. Tutte je svoju tvrdnju pokazao za 3-povezane planarne grafove, no taj se teorem može primijeniti i na trokutaste mreže, te se dokaz može jednostavno proširiti da se dopusti proizvoljan broj pozitivnih težina λ_{ij} u jednadžbi (3.2) koja zadovoljava:

$$\sum_{k=1}^{d_i} \lambda_{ik} = 1$$

Jednostavniji dokaz tog rezultata se nalazi i u [6].

Teorem 3.2.1. *Ako je $f : \mathcal{S}_\mathcal{T} \mapsto \mathcal{S}'$ preslikavanje konveksnih kombinacija vrhova koje preslikava $\partial\mathcal{S}_\mathcal{T}$ homeomorfno na konveksni poligon $\partial\mathcal{S}'$, tada je f injekcija.*

Dokaz teorema prati ideju dokaza RKC teorema, u smislu da prvo pokažemo da je preslikavanje f lokalno injektivno. Potpun dokaz ovog teorema može se naći u [6]. U diskretnom slučaju to znači da pokazujemo da je f injektivna na svakom četverokutu u $\mathcal{S}_\mathcal{T}$. Pod četverokut u triangulaciji $\mathcal{T}$ mislimo na uniju dva trokuta T_1 i T_2 koji dijele zajedničku stranicu, odnosno brid: $Q = T_1 \cup T_2$.

Teorem 3.2.2. *Neka su $\mathcal{T}$ i f kao u prethodnom teoremu. Tada je $f|_Q$ injektivna na svakom četverokutu u $\mathcal{T}$.*

Dokaz. Lokalnu injektivnost pokažemo pomoću dvije leme:

1. $f(v)$ se nalazi u unutrašnjosti od $\mathcal{S}'$ za svaki unutarnji vrh v od $\mathcal{T}$
2. Ako je $f|_{T_1}$ injektivna, tada je i $f|_{T_1 \cup T_2}$ injektivna.

Injektivnost na četverokutu dobijemo "povezivanjem" proizvoljnog trokuta T_1 sa trokutom T_k koji ima vrh na rubu $\partial\mathcal{S}'$:

Zbog povezanosti $\mathcal{S}_\mathcal{T}$, svaka 2 trokuta iz $\mathcal{T}$ su povezani putem trokuta, tj. nizom trokuta $T_1, T_2, \ldots, T_k$, takvog da svaka 2 trokuta T_i i T_{i+1} imaju zajednički brid za svaki i. Stoga je svaki trokut T_1 iz $\mathcal{T}$ povezan s bilo kojim trokutom T_k koji sadrži rubni brid od $\mathcal{T}$. Zbog prve leme, trokut $f(T_k)$ nije degeneriran. Primjenom druge leme zaključujemo da $f(T_{k-1})$ također nije degeneriran. Analogno vrijedi za $f(T_{k-2})$. Ponavljanjem argumenata iz druge leme dobivamo da $f(T_1)$ nije degeneriran. Stoga je f injektivna na svakom trokutu u $\mathcal{T}$. Iz druge leme sad slijedi da je f injektivna na svakom četverokutu u $\mathcal{T}$. □

Sad preostaje pokazati da globalna injektivnost slijedi iz lokalne. Za to su nam potrebne još 2 leme:

1. Ako je presjek dva trokuta T_1 i T_2 u $\mathcal{T}$ zajednički vrh v, tada je presjek $f(T_1) \cap f(T_2)$ zajednički vrh $f(v)$.

2. Ako je presjek $T_1 \cap T_2 = \emptyset$, tada je i $f(T_1) \cap f(T_2) = \emptyset$.

Dokaz. Dokaz teorema 3.2.1.

Neka su $\mathbf{x}_1 \neq \mathbf{x}_2$ točke u $\mathcal{S}_\mathcal{T}$. Ako pripadaju istom trokutu, tada $f(\mathbf{x}_1) \neq f(\mathbf{x}_2)$. Inače $\mathbf{x}_1 \in T_1$ i $\mathbf{x}_2 \in T_2$ gdje je $T_1 \neq T_2$. No zbog lokalne injektivnosti i zadnjih dviju lema, f je injektivna na $T_1 \cup T_2$ pa je $f(\mathbf{x}_1) \neq f(\mathbf{x}_2)$. □

Prisjetimo se težina u slučaju diskretnog harmonijskog preslikavanja:

$$w_{ik} = \operatorname{ctg} \gamma_{k-1} + \operatorname{ctg} \gamma_k$$

Pošto vrijedi

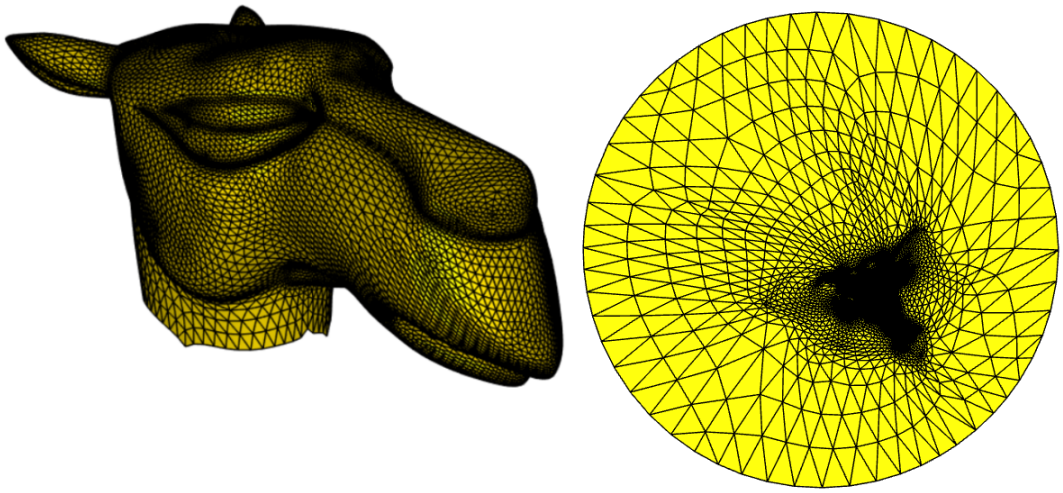
$$\operatorname{ctg} \gamma_{k-1} + \operatorname{ctg} \gamma_k = \frac{\sin(\gamma_{k-1} + \gamma_k)}{\sin \gamma_{k-1} \sin \gamma_k},$$

lako vidimo da one dovode do: $w_{ik} > 0 \iff \gamma_{k-1} + \gamma_k < \pi$. Stoga slijedi:

Propozicija 3.2.3. *Diskretno harmonijsko preslikavanje $f : \mathcal{S}_\mathcal{T} \mapsto \mathcal{S}'$ je injektivno ako preslika $\partial\mathcal{S}_\mathcal{T}$ homeomorfno u konveksan poligon $\partial\mathcal{S}'$ te ako je suma svakog para suprotnih kuteva četverokuta u $\mathcal{S}_\mathcal{T}$ manja od π.*

Općenito, uvjet za suprotne kuteve je ispunjen kad su trokuti "dobro-oblikovani", i vrijedi pogotovo kad su svi kutevi u svim trokutima unutar $\mathcal{S}_\mathcal{T}$ manji od $\pi/2$.

S druge strane, konstruirani su primjeri [3] koji pokazuju da diskretno harmonijsko preslikavanje možda nije bijektivno kad taj uvjet ne vrijedi: neki trokuti se mogu "preklopiti", tj. imati krivu orijentaciju.



Slika 3.3: Preslikavanje koordinata srednje vrijednosti

Koordinate srednje vrijednosti

Alternativna konstrukcija preslikavanja konveksnih kombinacija je bazirana na *koordinatama srednje vrijednosti* i motivirana na sljedeći način: Ideja je u uočavanju da harmonijske funkcije zadovoljavaju teorem srednje vrijednosti. Na svakoj točki u njenoj (planarnoj) domeni, vrijednost harmonijske funkcije je jednaka prosjeku vrijednosti oko bilo kojeg kruga oko te točke. To sugerira pronalazak po dijelovima linearnog preslikavanja $f : \mathcal{S}_{\mathcal{T}} \mapsto \mathcal{S}'$ za planarnu trokutastu mrežu $\mathcal{S}_{\mathcal{T}}$, koja zadovoljava teorem srednje vrijednosti na svakom unutarnjem vrhu mreže $\mathbf{x}_i$. Neka je Γ_i krug s središtem u $\mathbf{x}_i$, s radijusom $r_i > 0$ dovoljno malim da Γ_i presijeca trokute u $\mathcal{T}$ koji su incidentni s $\mathbf{x}_i$. Tada vrijedi:

$$\mathbf{u}_i = \frac{1}{2\pi r_i} \int_{\Gamma_i} \mathbf{u}(\mathbf{x}) \mathrm{d}s$$

Ispada da se ta jednadžba može zapisati u obliku 3.2, neovisno o odabiru $r_i > 0$. Potrebno je još odrediti težine w_{ik}. U specijalnom slučaju kad je mnogokut $\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_k$ konveksan, postoji rješenje [20] u kojem koordinate možemo zapisati u obliku racionalnih polinoma:

$$\begin{aligned} w_{ik} &= \frac{A(\mathbf{x}_{k-1}, \mathbf{x}_k, \mathbf{x}_{k+1})}{A(\mathbf{x}_{k-1}, \mathbf{x}_k, \mathbf{x}_i) A(\mathbf{x}_k, \mathbf{x}_{k+1}, \mathbf{x}_i)} \\ &= \frac{\operatorname{ctg} \gamma_{k-1} \operatorname{ctg} \beta_k}{\|\mathbf{x}_k - \mathbf{x}_i\|^2} \end{aligned}$$

gdje je $A(a, b, c)$ površina trokuta $[a, b, c]$ s predznakom. No s tim težinama koordinate λ_{ik} mogu biti negativne, te će biti točno kad $\gamma_{k-1} + \beta_k > \pi$. Ispada da sljedeće težine imaju svojstva koja tražimo:

Propozicija 3.2.4. *Neka je* $0 < \alpha_i < \pi$ *kut uz* $\mathbf{x}_i$ *u trokutu* $[\mathbf{x}_i, \mathbf{x}_k, \mathbf{x}_{k+1}]$. *Tada su težine*

$$w_{ik} = \frac{\operatorname{tg}(\alpha_k/2) + \operatorname{tg}(\alpha_{k-1}/2)}{\|\mathbf{x}_k - \mathbf{x}_i\|} \tag{3.4}$$

uz 3.3 koordinate za $\mathbf{x}_i$ *s obzirom na* $\mathbf{x}_1, \ldots, \mathbf{x}_n$.

Dokaz. Kako je $0 < \alpha_i < \pi$, $\operatorname{tg}(\alpha_i/2)$ je dobro definiran i pozitivan, pa je stoga i λ_{ik} dobro definiran i pozitivan za $k = 1, \ldots, n$ i po definiciji vrijedi da je $\sum_k \lambda_{ik} = 1$. Preostaje dokazati da je suma $\mathbf{x}_k$ stvarno jednaka $\mathbf{x}_i$. Iz 3.2 slijedi da je sljedeća jednadžba ekvivalentna jednadžbi 3.4

$$\sum_{k=1}^{n} w_{ik}(\mathbf{x}_k - \mathbf{x}_i) = 0 \tag{3.5}$$

Koristimo polarne koordinate oko $\mathbf{x}_i$, tj.

$$\mathbf{x}_k = \mathbf{x}_i + r_k(\cos\theta_k, \sin\theta_k).$$

Tada imamo

$$\frac{\mathbf{x}_k - \mathbf{x}_i}{\|\mathbf{x}_k - \mathbf{x}_i\|} = (\cos\theta_k, \sin\theta_k)$$

i vrijedi

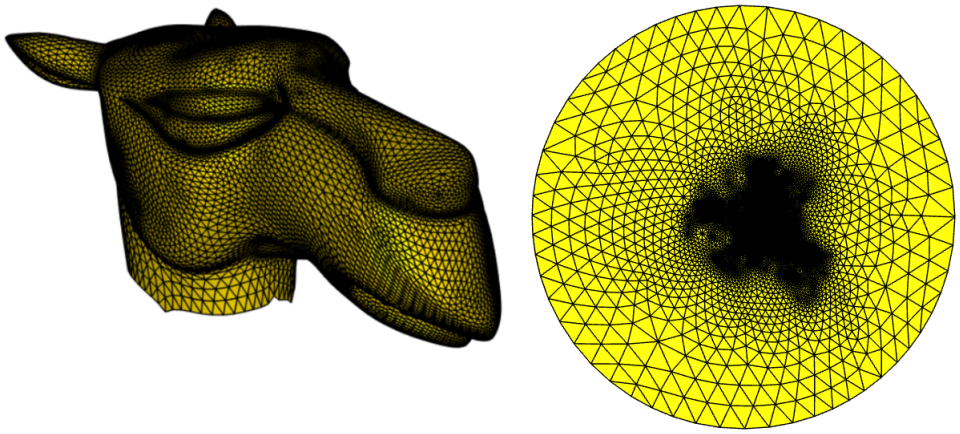
$$\alpha_k = \theta_{k+1} - \theta_k$$

iz jednadžbe 3.5 dobivamo:

$$\sum_{k=1}^{n} \left(\operatorname{tg}(\alpha_{k-1}/2) + \operatorname{tg}(\alpha_k/2)\right)(\cos\theta_k, \sin\theta_k) = 0$$

ili ekvivalentno:

$$\sum_{k=1}^{n} \operatorname{tg}(\alpha_k/2)\left((\cos\theta_k, \sin\theta_k) + (\cos\theta_{k+1}, \sin\theta_{k+1})\right) = 0$$



Slika 3.4: Baricentričko preslikavanje na jediničnu kružnicu

Kako bismo pokazali zadnju jednakost, primijetimo da vrijedi:

$$\begin{aligned} 0 &= \int_0^{2\pi} (\cos\theta, \sin\theta)\,\mathrm{d}\theta \\ &= \sum_{k=1}^{n} \int_{\theta_k}^{\theta_{k+1}} (\cos\theta, \sin\theta)\,\mathrm{d}\theta \\ &= \sum_{k=1}^{n} \int_{\theta_k}^{\theta_{k+1}} \frac{\sin(\theta_{k+1}-\theta)}{\sin\alpha_k}(\cos\theta_k, \sin\theta_k) + \frac{\sin(\theta-\theta_k)}{\sin\alpha_k}(\cos\theta_{k+1}, \sin\theta_{k+1})\,\mathrm{d}\theta \end{aligned}$$

gdje zadnju jednakost dobijemo iz formule za razliku sinusa. Traženu jednakost dobijemo primjenom sljedeće dvije jednakosti:

$$\int_{\theta_k}^{\theta_{k+1}} \sin(\theta_{k+1}-\theta)\,\mathrm{d}\theta = \int_{\theta_k}^{\theta_{k+1}} \sin(\theta-\theta_k)\,\mathrm{d}\theta = 1-\cos\alpha_k,$$

i

$$\operatorname{tg}(\alpha_k/2) = \frac{1-\cos\alpha_k}{\sin\alpha_k}.$$

□

Implementacija

Pogledajmo kako izgleda matrica L sustava kod baricentričkog preslikavanja. U tom slučaju je vrijednost $\lambda_{ij} = 1/d_i$, odnosno $w_{ik} = 1$.

Prisjetimo se kako izgleda Laplacijan:

$$L_{ij} = \begin{cases} w_{ij} & \text{ako } i \neq j \text{ i } \{i,j\} \in E, \\ -\sum_{\ell \neq i} L_{i\ell} & \text{ako } i = j \\ 0 & \text{inače} \end{cases}$$

Primijetimo da pošto je $w_{ij} = 1$, to znači da je tada matrica L simetrična. Matricu L možemo dobiti kao razliku dijagonalne matrice u kojem je vrijednost i-tog dijagonalnog elementa stupanj i-tog vrha, i matrice incidencije grafa. Slijedi isječak kôda koji upravo tako računa matricu L za preslikavanje sa slike 3.4:

```
# Izracunaj matricu incidencije za danu mrezu
a = igl.adjacency_matrix(F)
# Izracunaj dijagonalnu matricu stupnja grafa
a_sum = np.sum(a, axis=1)
a_diag = np.diag(np.array(a_sum).ravel())
# Izgradi laplacijan grafa
u = a - a_diag
```

Konveksno preslikavanje srednje vrijednosti

Za konveksno preslikavanje srednje vrijednosti, težine (a time i elemente matrice sustava) računamo po 3.4. Primijetimo da rezultirajuća matrica L više nije simetrična pošto općenito $w_{ij} \neq w_{ji}$. Slijedi kôd za izgradnju Laplacijana:

```
# Inicijaliziraj topologiju mreze
ev, fe, ef = igl.edge_topology(V,F)

# Inicijaliziraj matricu sustava
weights = np.zeros((V.shape[0],V.shape[0]))

# Za danu listu trokuta i  dani vrh, nadi bridove nasuprot vhu v
def find_edges(triangles, vertex, ev):
    target_edges = []
    for triangle in triangles:
        for edge in triangle:
            if vertex not in ev[edge]:
                target_edges = np.append(target_edges, edge)
    return target_edges

# Izgradi matricu sustava
for i in range(ev.shape[0]):
    v_i, v_j = ev[i][0], ev[i][1]
    if ((v_i not in b) or (v_j not in b)):
        distance = distances[v_i, v_j]
```

```
        all_edges = fe[ef[i, :], :]
        relevant_angles = angles[ef[i, :], ]
        relevant_edges_i = find_edges(all_edges, v_i, ev)
        alphas_i=relevant_angles[np.isin(all_edges, relevant_edges_i)]
        weights[v_i,v_j] = (np.tan(0.5 * alphas_i[0]) + np.tan(0.5 *
    alphas_i[1]) )
            / distance
        relevant_edges_j = find_edges(all_edges, v_j, ev)
        alphas_j=relevant_angles[np.isin(all_edges, relevant_edges_j)]
        weights[v_j,v_i] = (np.tan(0.5 * alphas_j[0]) + np.tan(0.5 *
    alphas_j[1]))
            / distance

w_sum = np.sum(weights, axis=1)
w_sumsum = np.array(w_sum).ravel()
w_diag = np.diag(w_sumsum)
u = weights - w_diag
```

Te nakon toga riješimo sustave:

```
v_all = np.arange(V.shape[0])
v_in = np.setdiff1d(v_all, b)

.
.
.

l_ii = u[v_in, :]
l_ii = l_ii[:, v_in]
l_ib = u[v_in, :]
l_ib = l_ib[:, b]

# Rijesi sustav
lu = sla.splu(-l_ii)
xs = lu.solve(np.transpose(l_ib.dot(bc[:,0])))
ys = lu.solve(np.transpose(l_ib.dot(bc[:,1])))
uv= np.column_stack((xs,ys))
# Zapisi vrhove u ispravnom redoslijedu
Final_uv= np.zeros((V.shape[0],2))
Final_uv[v_in, :] = uv
Final_uv[b, :] = bc
```

Poglavlje 4

Preslikavanja slobodnog ruba

Ovdje izlažemo preslikavanja *slobodnog ruba*. Za razliku od preslikavanja fiksiranog ruba, preslikavanja izložena u ovom poglavlju nemaju fiksiran cijeli rub. Uočimo da neka ograničenja i dalje moraju postojati, inače ćemo dobiti degenerirano rješenje $u = 0$, $v = 0$.

4.1 Konformno preslikavanje određeno u smislu najmanjih kvadrata

Konformno preslikavanje određeno u smislu najmanjih kvadrata minimizira deformacije kuteva [12]. Kao što smo već rekli, ovdje nije potrebno fiksirati cijeli rub, no kako bi izbjegli degenerirano rješenje ipak je potrebno fiksirati bar 2 točke na samom rubu. Za po dijelovima linearno preslikavanje, konformno preslikavanje određeno u smislu najmanjih kvadrata može se dobiti minimiziranjem energije:

$$\min_{\substack{\mathbf{u}=(u,v)\\ \mathbf{u}(\mathcal{I})=\mathcal{I}'}} E_{LSCM}(\mathbf{u}) = \min_{\substack{\mathbf{u}=(u,v)\\ \mathbf{u}(\mathcal{I})=\mathcal{I}'}} \int_{\mathcal{S}_\mathcal{T}} \frac{1}{2}|\nabla u^\perp - \nabla v|^2 \mathrm{d}s \tag{4.1}$$

gdje $^\perp$ označava rotaciju od 90° u pozitivnom smjeru u $\mathcal{S}$. Za $3D$ plohu definiranu vektorom normale $\mathbf{n}$, rotacija gradijenta u pozitivnom smjeru se može zapisati kao:

$$\nabla u^\perp = \mathbf{n} \times \nabla u$$

Jednadžba (4.1) se može pojednostaviti na sljedeći način:

$$\begin{aligned} E_{LSCM}(\mathbf{u}) &= \int_{\mathcal{S}} \frac{1}{2}(\nabla u^\perp \cdot \nabla u^\perp + \nabla v \cdot \nabla v - 2\nabla u^\perp \cdot \nabla v)\mathrm{d}s \\ &= \int_{\mathcal{S}} \frac{1}{2}(\nabla u \cdot \nabla u + \nabla v \cdot \nabla v - 2(\mathbf{n} \times \nabla u) \cdot \nabla v)\mathrm{d}s \end{aligned}$$

Zamijenimo vektorski i skalarni produkt i dobimo:

$$\begin{aligned} E_{LSCM}(\mathbf{u}) &= \int_{\mathcal{S}} \frac{1}{2}\left((\nabla u)^2 + (\nabla v)^2\right) - \mathbf{n}\cdot(\nabla u \times \nabla v)\mathrm{d}s \\ &= E_D(\mathbf{u}) - \mathcal{A}(\mathbf{u}) \end{aligned}$$

gdje je $\mathcal{A}(\mathbf{u})$ funkcional površine. Iz prethodne jednadžbe slijedi da je konformna energija definirana kao Dirichletov funkcional $E_D(\mathbf{u})$ minus funkcional površine $\mathcal{A}(\mathbf{u})$, gdje je $\mathcal{A}(\mathbf{u}) = \frac{1}{2}\sum_{\{i,j\}\in\partial\mathcal{S}} |\,\mathbf{u}_i\,\mathbf{u}_j\,|$. Ovdje imamo jednostavan kvadratni izraz: sumiramo po bridovima ruba determinantu matrice s koordinatama vrhova kao stupcima. Tu kvadratnu formu možemo zapisati kao $\mathbf{u}^T A\mathbf{u}$ s vektoriziranim $u-$ i $v-$ koordinatama preslikavanja u $\mathbf{u} \in \mathbb{R}^{2n}$, gdje je $A \in \mathbb{R}^{2n\times 2n}$ rijetko popunjena matrica koja poprima nenul vrijednosti samo na vrhovima na rubu $\partial\mathcal{S}$. Kada k tome dodamo Dirichletovu energiju, dani izraz možemo napisati na sljedeći način:

$$\mathbf{u}^T \underbrace{\left(\begin{pmatrix} L & 0 \\ 0 & L \end{pmatrix} - A\right)}_{L_c} \mathbf{u} \tag{4.2}$$

gdje je $L \in \mathbb{R}^{n\times n}$ kotangens Laplacijan koji smo definirali u diskretnom harmonijskom preslikavanju.

Rubni uvjeti

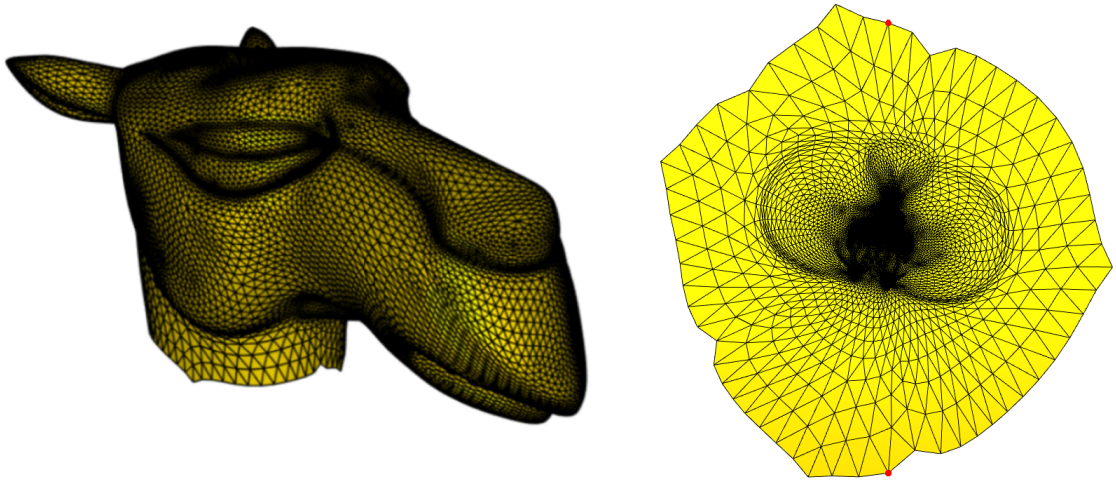
Kako bi izbjegli degenerirano rješenje, moramo fiksirati (bar) 2 vrha. Promatramo mrežu $\mathcal{S}_\mathcal{T}$ kao težinski graf gdje su težine bridova udaljenosti između vrhova. Tada na $\partial\mathcal{S}_\mathcal{T}$ kao induciranom podgrafu od $\mathcal{S}_\mathcal{T}$ tražimo 2 točke koje su najudaljenije. Neka je $l : V_B \times V_B \mapsto \mathbb{R}$ funkcija koja vraća udaljenost dva vrha triangulacije na $\partial\mathcal{S}_\mathcal{T}$. Odabiremo 2 vrha $\mathbf{x}_i$ i $\mathbf{x}_j$ takav da je $l(\mathbf{x}_i, \mathbf{x}_j) = \max_{k,l} l(\mathbf{x}_k, \mathbf{x}_l)$ te njih preslikamo u vrhove $(0,0)$ i $(0,1)$ u parametarskom prostoru. Odnosno $\mathbf{u}(\mathbf{x}_i) = (0,0)$, a $\mathbf{u}(\mathbf{x}_j) = (0,1)$.

Implementacija

Slijedi kôd za izračun vrhova ruba na $\partial\mathcal{S}_\mathcal{T}$ čija je udaljenost najveća. Funkcija `shortest_path` koristi Floydov algoritam koji smo opisali u uvodu.

```
boundary_distances = igl.all_pairs_distances(V[bnd],V[bnd], False)
neighbours = igl.adjacency_matrix(F)
neighbours = neighbours[bnd]
neighbours = neighbours[:,bnd]
rez = boundary_distances * neighbours
```

```
rez = rez.copy(order='C')
rez2 = sp.sparse.csgraph.shortest_path(rez, directed=False)
ind = np.unravel_index(np.argmax(rez2, axis=None), rez2.shape)
```



Slika 4.1: Parametrizacija glave deve $\mathcal{S}$ pomoću konformnog preslikavanja određenog u smislu najmanjih kvadrata

Formula za izračun matrice Q za konformno preslikavanje određeno u smislu najmanjih kvadrata je dana u 4.2. Fiksirali smo najudaljenije točke ruba na točke $(0,0)$ i $(0,1)$ u parametarskom prostoru. Primjer dobivene parametrizacije se može vidjeti na slici 4.1.

```
b[0] = bnd[ind[0]]
b[1] = bnd[ind[1]]
bc = np.array([[0.0, 0.0], [1.0, 0.0]])

A = vector_area_matrix(F)

L = igl.cotmatrix(V,F)

L_flat = sp.sparse.block_diag((L,L))

b_flat = np.zeros((b.size*bc.shape[1],1))
bc_flat =  np.zeros((bc.size,1))

for column in range(np.shape(bc)[1]):
    b_flat[column*b.size : column*b.size+ b.shape[0]] = column*V.shape
    [0] + b.reshape(b.size,1)
    bc_flat[column*bc.shape[0]:(column+1)*bc.shape[0]] = bc[:,column].
    reshape(2,1)

```

```
18 ## Minimiziraj LSCM energiju
19
20 Q = csc_matrix(2*A) - L_flat
```

4.2 Spektralna parametrizacija

Kod konformnog preslikavanja određenog u smislu najmanjih kvadrata, izbor vrhova može drastično utjecati na rezultate. Iako fiksiranje dva vrha daje dobre rezultate, može doći do globalne distorzije te ponekad značajne degradacije u konformnosti kod tih fiksiranih vrhova. Taj problem se pogorša ako odlučimo fiksirati više od 2 vrha. Možemo dobiti parametrizaciju s manjom distorzijom spektralnom analizom matrice L_c.

Fiedlerov vektor matrice L_c

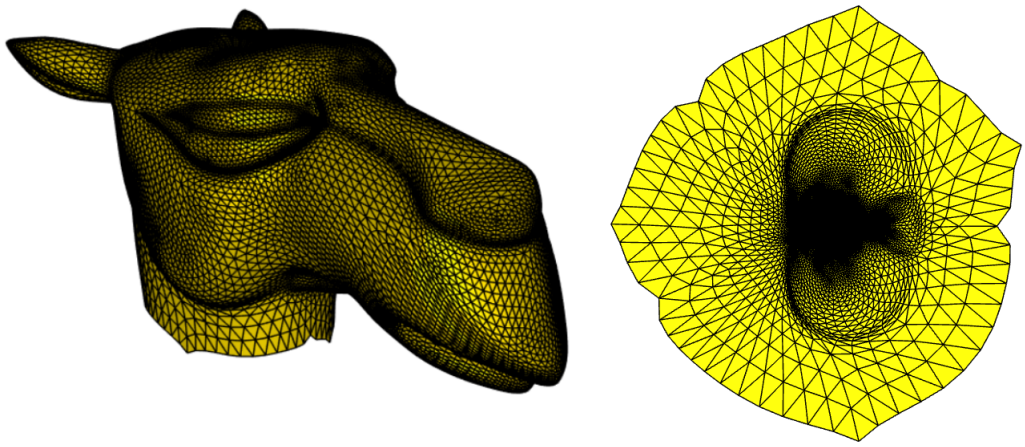
Želimo izbjeći fiksiranje vrhova u sustavu $L_c\mathbf{u} = 0$ kroz spektralnu teoriju. Bitan rezultat iz teorije grafova je karakterizacija svojstvenih vektora rijetko popunjene simetrične pozitivne semidefinitne Laplacijan matrice: svojstveni vektor $\mathbf{u}^*$ pridružen prvoj ne-nul svojstvenoj vrijednosti $n \times n$ Laplacijan matrici L (tj. vektor koji zadovoljava $L\mathbf{u}^* = \lambda\mathbf{u}^*$, gdje je λ najmanja nenul svojstvena vrijednost) ranga $n-k$ je rješenje problema minimizacije kvadratne forme s ograničenjima:

$$\begin{aligned} \mathbf{u}^* &= \arg\min_{\mathbf{u}\in\mathbb{R}^{2n}} \mathbf{u}^T L\mathbf{u} \\ \mathbf{u}^T E &= 0 \\ \mathbf{u}^T\mathbf{u} &= 1 \end{aligned}$$

gdje je E $n \times k$ matrica čiji stupci generiraju jezgru od L. Taj svojstveni vektor $\mathbf{u}^*$ zovemo *Fiedlerov vektor* od L. U našem kontekstu parametrizacije mreže, Fiedlerov vektor od L_c ima jednostavnu intuitivnu interpretaciju: To je najbliže rješenje linearnog sustava konformnog preslikavanja određenog u smislu najmanjih kvadrata pod uvjetom da je baricentar rješenja u 0 ($\mathbf{u}^{*T}E = 0$) i njen moment inercije (tj. suma kvadriranih udaljenosti od baricentra) mora biti jediničan ($\mathbf{u}^{*T}\mathbf{u}^* = 1$). Taj spektralni pristup daje parametrizaciju bez fiksiranja vrhova: ograničenja su efektivno podijeljena jednako kroz mrežu.

No ovo iskorištavanje Fiedlerovog vektora samo po sebi nije dobra alternativa. Fiedlerov vektor minimizira takozvani *Rayleigh kvocijent*:

$$\mathbf{u}^* = \arg\min_{\mathbf{u}} \left| \frac{\mathbf{u}^T L\mathbf{u}}{\mathbf{u}^T\mathbf{u}} \right|$$



Slika 4.2: Spektralno preslikavanje glave deve

time dobivamo balans između diskretne konformnosti (kako bi najbolje minimizirali brojnik) te $\mathcal{L}^2$ udaljenosti od ishodišta (kako bi maksimizirali nazivnik). Norma $\mathcal{L}^2$, koja je bitna za energiju između svakog vrha i ishodišta, nije prigodna u ovom kontekstu: minimiziranje Rayleighjevog kvocijenta plaćamo gubitkom lokalne konformnosti.

Spektralno konformno preslikavanje

Umjesto da koristimo svojstveni vektor Laplacijan matrice direktno, definiramo diskretnu spektralnu konformnu parametrizaciju $\mathbf{u}^*$ kao generalizirani svojstveni vektor koji zadovoljava:

$$L_c\mathbf{u} = \lambda B\mathbf{u} \tag{4.3}$$

gdje B je $2V \times 2V$ dijagonalna matrica s jedinicom na svakom dijagonalnom elementu koji odgovara rubnom vrhu i 0 na ostalim elementima. Kako je L_c pozitivno semidefinitna, te su obje matrice simetrične, generalizirane svojstvene vrijednosti su realne. Taj novi problem svojstvene vrijednosti odgovara sljedećoj minimizaciji s ograničenjima:

$$\begin{aligned} \mathbf{u}^* &= \min_{\mathbf{u}\in\mathbb{R}^{2n}} \mathbf{u}^T L_c \mathbf{u} \\ \mathbf{u}^T BE &= 0 \\ \mathbf{u}^T B\mathbf{u} &= 1 \end{aligned}$$

gdje je E $2V \times 2$ matrica takva da je $E_{i1} = 1, i = 1, \ldots, V$, te $E_{i2} = 1, i = V+1, \ldots, 2V$ (ostali elementi od E su 0). Ekvivalentno, 4.3 odgovara modificiranom Rayleighjevom

kvocijentu $\mathbf{u}^T L_C \mathbf{u} / \mathbf{u}^T B \mathbf{u}$, gdje nazivnik sad ovisi samo o rubnim vrhovima isječka. Stoga će optimalni svojstveni vektor balansirati konformnost za kvadratnu formu definiranu matricom B na rubu: spektralna parametrizacija maksimizira kvadriranu udaljenost rubnih vrhova do njihova baricentra na jediničnoj kružnici definiranoj konformnom energijom.

Implementacija

S numeričkog stajališta, postoje vrlo efikasni solveri koji pronalaze Fiedlerov vektor $\mathbf{u}^*$ generaliziranog problema svojstvene vrijednosti koji mi trebamo riješiti. U našoj implementaciji koristimo `eigs` funkciju iz biblioteke **scipy** koja je zapravo wrapper za **ARPACK** funkcije za pronalazak svojstvenih vrijednosti i svojstvenih vektora. **ARPACK** pak implementira već opisani implicitno restartani Arnoldijev algoritam. [14]

Slijedi implementacija spektralnog konformnog preslikavanja, čiji se rezultat može vidjeti na slici 4.2:

```
A = vector_area_matrix(F)
L = igl.cotmatrix(V, F)
L_diag = sp.sparse.block_diag(L,L)

Q = L_diag - csc_matrix(2 * A)

b = igl.boundary_loop(F)
M = np.zeros(V.shape[0])
M[b] = 1
M = sp.sparse.csc_matrix(np.diag(M))

B = sp.sparse.block_diag(M,M)

lamb, v = sla.eigs(Q, 3, csc_matrix(B), which='LM', sigma = 0 )

Spec_uv = np.zeros((V.shape[0],2))
Spec_uv[:,0] = v[:V.shape[0],2]
Spec_uv[:,1] = v[V.shape[0]:,2]
```

Poglavlje 5

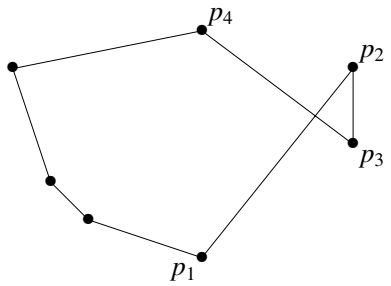
Problemi s diskretnim linearnim preslikavanjima

Iznimno bitan aspekt kod izračunavanja parametrizacije je dobiti bijektivno preslikavanje, gdje svaka točka na parametriziranoj domeni odgovara točno jednoj točki na mreži. Pogledajmo pet različitih diskretnih linearnih preslikavanja koje smo do sad prezentirali:

1. harmonijsko preslikavanje konačnih elemenata: rješenje problema minimizacije kvadratne forme s ograničenjima s težinama danim u 3.1
2. Baricentričko konveksno preslikavanje: rješenje problema minimizacije kvadratne forme s ograničenjima gdje su elementi Laplacijana dani s 3.1 a težine $w_{ij} = 1$
3. Konveksno preslikavanje srednjih vrijednosti: rješenje linearnih sustava $L\mathbf{u} = 0$ i $L\mathbf{v} = 0$ gdje su elementi Laplacijana dani s 3.1 a težine w_{ij} s 3.4
4. Konformno preslikavanje konačnih elemenata: rješenje problema 4.2 s dva (proizvoljno) fiksirana vrha
5. Spektralno konformno preslikavanje: rješenje generaliziranog problema svojstvenih vrijednosti 4.3

Harmonijska i konveksna preslikavanja preslikaju rub na fiksirane konveksne rubove (npr. jedinična kružnica), dok su konformna preslikavanja parametrizacije otvorenog ruba. Preslikavanja konveksnih kombinacija su uvijek bijektivna, no ne-bijektivne parametrizacije se mogu javiti kod računanja diskretnih harmonijskih ili konformnih preslikavanja kao rješenja problema minimizacije. Ovdje opisujemo te probleme te prolazimo kroz postprocessing algoritme koji garantiraju bijektivno preslikavanje.

Prvi problem se tiče preklapanja trokuta koji se može javiti kod izračuna linearnih konformnih preslikavanja s otvorenim rubovima. Kako bi detektirali taj problem, koristimo



Slika 5.1: presjek bridova u parametrizaciji $\partial\mathcal{S}^0$

ideju sličnu ideji iz [17] koja provjerava ima li preklapanja rubova. Oni se detektiraju prolaskom po bridovima ruba $\partial\mathcal{S}^0$ i računanjem postoji li presjek između bridova (vidi sliku 5.1). Presjek dva bridna segmenta $[p_1, p_2]$ i $[p_3, p_4]$ parametarskih jednadžba:

$$p_a = p_1 + t_a(p_2 - p_1), \qquad p_b = p_1 + t_b(p_4 - p_3)$$

se može izračunati pronalaskom točke gdje $p_a = p_b$. To daje sljedeći linearni sustav jednadžbi:

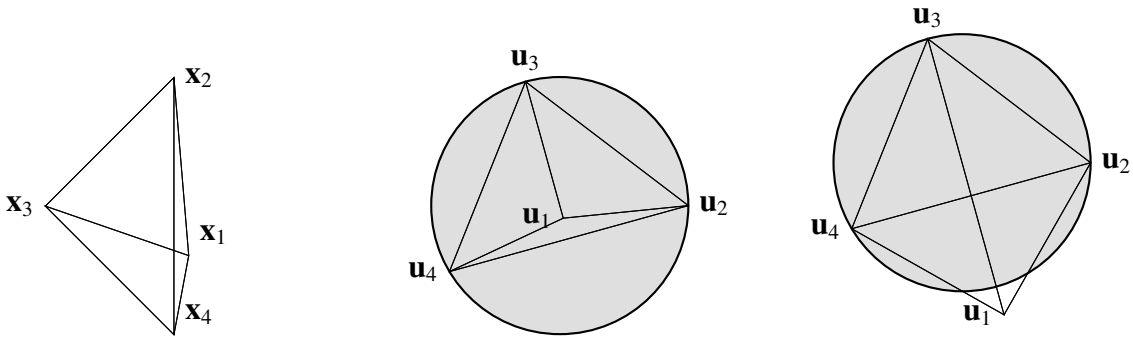
$$\begin{pmatrix} x_2 - x_1 & x_3 - x_4 \\ y_2 - y_1 & y_3 - y_4 \end{pmatrix} \begin{pmatrix} t_a \\ t_b \end{pmatrix} = \begin{pmatrix} x_3 - x_1 \\ y_3 - y_1 \end{pmatrix}$$

Ako je $t_a \in [0, 1]$ i $t_b \in [0, 1]$, nalazimo se u slučaju presjeka segmenta i dogodi se preklapanje trokuta. U tom slučaju, prebacimo se na drugi tip konformne parametrizacije ili ponovno izračunamo konformnu parametrizaciju odabirom druga dva fiksirana vrha.

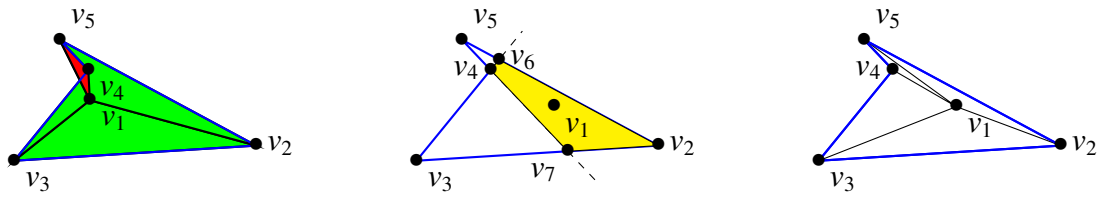
5.1 Okretanje trokuta

Drugi problem se tiče okretanja trokuta. Taj problem se može javiti kod harmonijskih ili konformnih preslikavanja. Jedan način kako možemo prisiliti da diskretno preslikavanje bude bijektivno je implementirati *algoritam za lokalnu provjeru rupa* koji lokalno promijeni unutarnje rupe u mreži u kojima se dogodi okretanje.

Već smo spomenuli da diskretno harmonijsko preslikavanje nije uvijek bijektivno, za razliku od neprekidnih. To možemo vidjeti u sljedećem primjeru: Promatramo jednostavnu triangulaciju definirana sa tri trokuta na slici 5.2: $\mathcal{S}_\mathcal{T} = \{(1, 2, 3), (1, 3, 4), (1, 4, 2)\}$ te neka je $\mathbf{x}_1 = (r, 0, 1)$ za neki realni broj $r > 0$, $\mathbf{x}_2 = (1, 1, 0)$, $\mathbf{x}_3 = (0, 0, 0)$, i $\mathbf{x}_4 = (1, -1, 0)$. Tri rubna vrha $\mathbf{x}_1$, $\mathbf{x}_2$, i $\mathbf{x}_3$ su preslikani na rub jedinične kružnice te bi vrh $\mathbf{x}_1$ trebao biti preslikan unutar trokuta $(\mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4)$ kako bi osigurali bijektivno preslikavanje. No u slučaju kad su mreže iskrivljene, numeričke metode za rješavanje Laplaceove jednadžbe mogu dati rješenja harmonijske parametrizacije koja nisu bijektivna.



Slika 5.2: Triangulacija za koju diskretno harmonijsko preslikavanje nije bijektivno. Za prvu parametrizaciju je $r = 1.5$, a u drugoj $r = 4.5$ te točka uopće nije u jediničnoj kružnici

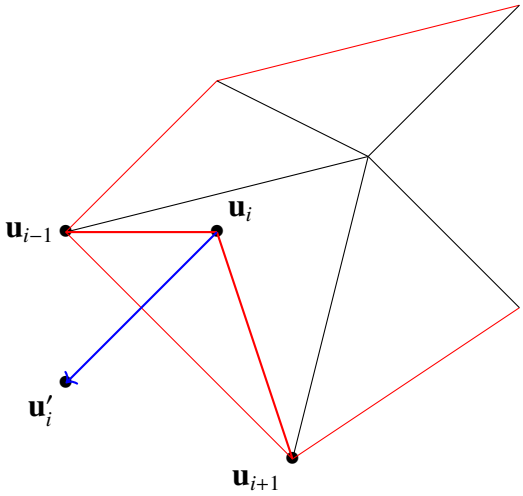


Slika 5.3: Vrh v_1 s 4 susjedna trokuta (v_1, v_2, v_3), (v_1, v_3, v_4), (v_1, v_4, v_5), (v_1, v_5, v_2) koji definiraju mnogokut (v_2, v_3, v_4, v_5). Trokut u crvenom nije dobro orijentiran i preklapa se s drugim trokutima. Pomičemo vrh v_1 unutar jezgre mnogokuta. Na kraju su svi trokuti dobro orijentirani.

Jedna mogućnost kako bi mogli osigurati bijektivnost svodi se na to da svaku točku u parametarskom prostoru stavimo u centar gravitacije svojih susjeda, tj. izračunati *baricentričko preslikavanje*. Umjesto da provodimo baricentričko preslikavanje globalno, postoji lokalan način za provođenje diskretnih bijektivnih preslikavanja - lokalni algoritam za provjeru rupa:

1. Izračunaj harmonijsko preslikavanje
2. Za svaki unutrašnji vrh v_i parametarske ravnine, provjeri je li svaki susjedni trokut točno orijentiran
3. Ako neki element nije, pomakni vrh u centar gravitacije za jezgru poligona P oko vrha

Okretanje trokuta se još može javiti kod otvorenog ruba. Stoga je potrebno provjeriti sve trokute uz rub te vratiti ispravnu orijentaciju pomakom parametarske točke $\mathbf{u}_i$ u točku $\mathbf{u}_i'$. Ta nova točka će se nalaziti na projekcijskom pravcu točke $\mathbf{u}_i$ na segmentu $[\mathbf{u}_{i-1}, \mathbf{u}_{i+1}]$.



Slika 5.4: Popravak okretanja trokuta kod rubova. Točka $\mathbf{u}_i$ je pomaknuta u $\mathbf{u}'_i$ projekcijskog pravca s $\mathbf{u}_i$ na $[\mathbf{u}_{i-1}, \mathbf{u}_{i+1}]$

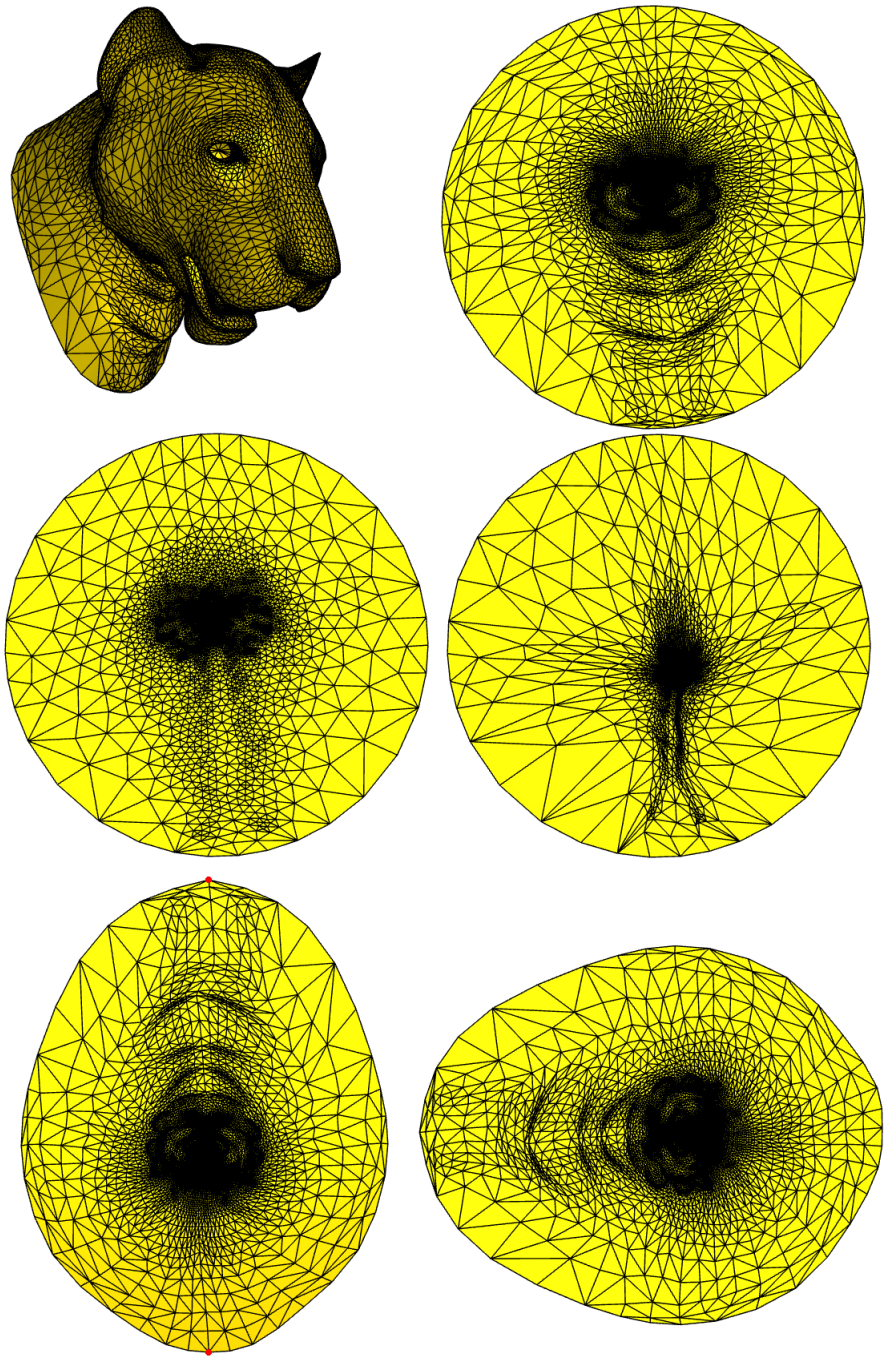
Poglavlje 6

Primjeri ponovnog generiranja mreže

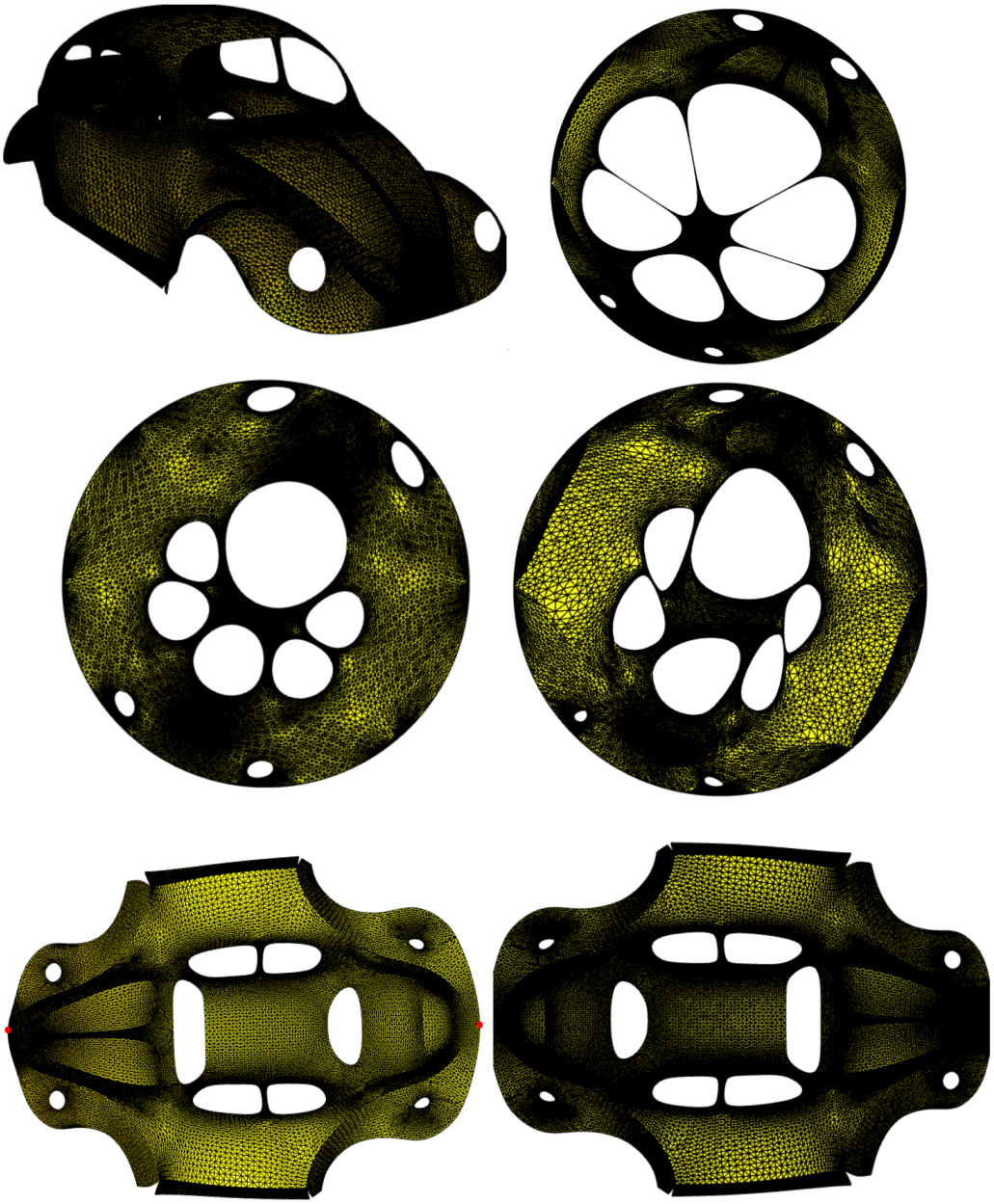
Sve parametrizacije izračunate su na virtualnoj instanci na Google Cloud Platform od 12 vCPU i 35GB RAM-a. Ukupno parametriziramo 3 plohe:

1. glavu deve koja se sastoji od 11381 vrhova te 22704 trokuta, za koju smo već vidjeli opisane parametrizacije
2. glavu lava koja se sastoji od 8356 vrhova te 16674 trokuta
3. karoseriju Volkswagen Beetle-a (”bube”) koja se sastoji od 19887 vrhova te 38656 trokuta, te uz to još ima 10 rubova, odnosno ”rupa”

Rezultati su dani u tablici 6.1, s parametrizacijama na slikama 6.1 i 6.2. U svim slučajevima harmonijsko preslikavanje je daleko najbrže, no daje parametrizaciju najgore kvalitete. Baricentričko preslikavanje daje mrežu dobre kvalitete, što i ima smisla jer odabirom baricentra dobivamo trokute blizu jednakostraničnima. U slučaju mreže s dodatnim rubovima, baricentričko preslikavanje ipak ne daje dobre rezultate (kvaliteta trokuta drastično pada upravo kod dodatnih rubova). U tom slučaju, spektralno konformno preslikavanje daje zadovoljavajuće rezultate, no ono je također sporije. Također se iz slike 6.2 jasno vidi zašto su parametrizacije slobodnog ruba u ovom slučaju mnogo kvalitetnije.



Slika 6.1: 3*D* mreža lavlje glave s odgovarajućim parametrizacijama: harmonijsko, baricentričko, konveksno, konformno i spektralno



Slika 6.2: 3*D* model ”bube” s odgovarajućim parametrizacijama

Parametrizacija glave deve				
Parametrizacija / mjera	κ_{min}	$\overline{\kappa}$	t(s)	τ
Harmonijsko	0.000861	0.676618	0.0617	37.108113
Baricentričko	0.375128	0.984694	3.40	37.120576
Konveksno	0.000179	0.692953	12.4	37.042598
Konformno	0.000829	0.676174	6.37	36.943840
Spektralno	0.000649	0.672987	7.82	37.685880
Parametrizacija lavlje glave				
Harmonijsko	0.018260	0.660184	0.109	37.230555
Baricentričko	0.215865	0.925995	1.53	37.142174
Konveksno	0.001439	0.670824	7.50	37.034724
Konformno	0.026051	0.669172	3.46	36.976596
Spektralno	0.026514	0.669074	5.04	37.613886
Parametrizacija bube				
Harmonijsko	$3.597 \cdot 10^{-9}$	0.410302	0.130	37.299283
Baricentričko	$1.740 \cdot 10^{-5}$	0.712079	11.60	37.239490
Konveksno	$1.717 \cdot 10^{-5}$	0.590478	28.9	37.252592
Konformno	$1.384 \cdot 10^{-9}$	0.244915	19.6	57.964728
Spektralno	0.000580	0.584120	23.70	37.752037

Tablica 6.1: Mjere kvalitete parametrizacija za različite modele

Poglavlje 7

Zaključak

U slučaju da nam je za parametrizaciju najviše bitna brzina izvršavanja, te smo za to spremni žrtvovati kvalitetu trokuta, harmonijsko preslikavanje daje najbolje rezultate. Ako želimo dobiti što bolju parametrizaciju s što manje degeneriranih trokuta, baricentričko preslikavanje je daleko najbolje. Konačno, u slučaju ploha s više rubova, konformna preslikavanja daju odlične rezultate u usporedbi s parametrizacijama fiksiranog ruba, ali to plaćamo brzinom izvođenja.

Mana koja je najviše očita iz tablice rezultata 6.1 je brzina izvođenja prezentiranih parametrizacija. Nešto od toga možemo pripisati odabiru **Pythona** kao jezika za implementaciju, no također valja istražiti mogu li se sami algoritmi ubrzati.

Usprkos tomu, parametrizacije ploha se mogu koristiti u više svrha osim ponovnog generiranja kao što su kompresija mreža, vizualizacija u medicinske svrhe (na primjer računanje parametrizacije moždane ovojnice na površinu), ili modeliranje modela iz ploha materijala (za izradu $3D$ modela potrebno je izrezati iz površine materijala ispravnu $2D$ parametrizaciju).

Bibliografija

[1] Gilles Brassard i Paul Bratley, *Algorithmics: Theory and Practice*, Prentice-Hall, Inc., USA, 1988, ISBN 0130232432.

[2] Richard Courant, *Dirichlet's Principle, Conformal Mapping, and Minimal Surfaces*, Springer New York, 1977, http://dx.doi.org/10.1007/978-1-4612-9917-2.

[3] T. Duchamp, A. Certain, A. Derose i W. Stuetzle, *Hierarchical Computation Of Pl Harmonic Embeddings*, Teh. izv., 1997.

[4] Peter Duren, *Harmonic Mappings in the Plane*, Cambridge Tracts in Mathematics, Cambridge University Press, 2004.

[5] Michael S. Floater i Kai Hormann, *Surface Parameterization: a Tutorial and Survey.*, Advances in Multiresolution for Geometric Modelling (Neil A. Dodgson, Michael S. Floater i Malcolm A. Sabin, ur.), Springer, 2005, str. 157–186, ISBN 978-3-540-26808-6, http://dblp.uni-trier.de/db/books/collections/DFS2005.html#FloaterH05.

[6] Michael Floater, *One-to-One Piecewise Linear Mappings Over Triangulations*, Mathematics of Computation **72** (2002).

[7] Pascal Jean Frey i Paul Louis George, *Mesh Generation: Application to Finite Elements*, ISTE, 2007, ISBN 1903398002.

[8] Jean Gallier i Dianna Xu, *A Guide to the Classification Theorem for Compact Surfaces*, Geometry and Computing, sv. 9, Springer, 2013, ISBN 978-3-642-34363-6, https://doi.org/10.1007/978-3-642-34364-3.

[9] Alec Jacobson, Daniele Panozzo et al., *libigl: A simple C++ geometry processing library*, 2018, https://libigl.github.io/.

[10] Lennard Kamenski, Weizhang Huang i Hongguo Xu, *Conditioning of finite element equations with arbitrary anisotropic meshes*, Math. Comput. **83** (2014), br. 289, 2187–2211, ISSN 0025-5718; 1088-6842/E (English).

[11] R. B. Lehoucq, D. C. Sorensen i C. Yang, *ARPACK Users Guide: Solution of Large Scale Eigenvalue Problems by Implicitly Restarted Arnoldi Methods.*, 1997.

[12] Bruno Lévy, Sylvain Petitjean, Nicolas Ray i Jérome Maillot, *Least Squares Conformal Maps for Automatic Texture Atlas Generation*, ACM Trans. Graph. **21** (2002), br. 3, 362–371, ISSN 0730-0301, https://doi.org/10.1145/566654.566590.

[13] Emilie Marchandise, Jean François Remacle i Christophe Geuzaine, *Optimal Parametrizations for Surface Remeshing*, Eng. with Comput. **30** (2014), br. 3, 383–402, ISSN 0177-0667, https://doi.org/10.1007/s00366-012-0309-3.

[14] Patrick Mullen, Yiying Tong, Pierre Alliez i Mathieu Desbrun, *Spectral Conformal Parameterization*, Computer Graphics Forum (2008), ISSN 1467-8659.

[15] Henri P. Gavin i Jeffrey T. Scruggs, *Constrained Optimization Using Lagrange Multipliers, CEE 201L. Uncertainty, Design, and Optimization*, https://people.duke.edu/~hpgavin/cee201/LagrangeMultipliers.pdf.

[16] Fazeli S. A. Shahzadeh, Nahid Emad i Zifan Liu, *A KEY TO CHOOSE SUBSPACE SIZE IN IMPLICITLY RESTARTED ARNOLDI METHOD*, Teh. izv., listopad 2014, https://hal.archives-ouvertes.fr/hal-01070577, Submitted to the journal of Numerical Algorithms.

[17] A. Sheffer i E. Sturler, *Parameterization of Faceted Surfaces for Meshing using Angle-Based Flattening*, Engineering with Computers **17** (2001), 326–337.

[18] Milena Sošić, *Uvod u diferenicjalnu geometriju, predavanja*, http://www.math.uniri.hr/~msosic/Uvod%20u%20dif%20geometriju/Predavanja/13_Plohe.pdf.

[19] W. T. Tutte, *How to draw a Graph*, (1962).

[20] Eugene L. Wachspress, *A rational finite element basis / Eugene L. Wachspress*, Academic Press New York, 1975, ISBN 012728950.

Sažetak

U ovom radu bavimo se parametrizacijama ploha $3D$ žičanog modela. Prezentiramo motivaciju te teorijsku pozadinu parametrizacije ploha te predstavljamo 5 različitih parametrizacijskih metoda. Te parametrizacije potom implementiramo u programskom jeziku **python** koristeći klasične biblioteke **numpy** i **scipy**, te **libigl** kao biblioteku za procesiranje geometrije. Nakon toga, mjerimo njihova svojstva na primjerima kakvi se mogu naći u stvarnom svijetu. Konačno, uspoređujemo njihovu kvalitetu i efikasnost te dajemo preporuku ovisno o željenoj brzini te kvaliteti izračuna parametrizacije.

Summary

In this master thesis we deal with surface parameterizations of a $3D$ CAD model. We present the motivation behind and theoretical background of surface parameterizations as well as 5 concrete parameterization methods. We then implement these methods in **python** using **numpy** and **scipy** libraries, in addition to **libigl**, a library for geometry processing. Afterwards, we measure the properties and quality of resulting parameterizations on examples of real-life scale. Finally, we compare their efficiency and quality and give a recommendation as to which parameterization to pick based on desired computational speed and quality.

Životopis

Rođen sam 17.12.1996. godine u Varaždinu. Nakon osnovne škole 2011. upisujem opću gimnaziju u Čakovcu te sudjelujem u ekipnim i individualnim natjecanjima iz matematike. 2015. godine započinjem studiranjem na preddiplomskom sveučilišnom studiju Matematike na Matematičkom odsjeku Prirodoslovno-matematičkog fakulteta u Zagrebu. Nakon stjecanja diplome sveučilišnog prvostupnika, 2018. godine upisujem diplomski sveučilišni studij Matematike, smjer Računarstvo i matematika.