

ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA

---

SCUOLA DI SCIENZE  
Corso di Laurea Magistrale in Matematica

**UN MODELLO VARIAZIONALE  
NON CONVESSO PER IL  
DENOISING DI SUPERFICI**

**Relatore:**  
Chiar.ma Prof.ssa  
SERENA MORIGI  
**Correlatore:**  
Dott.  
MARTIN HÚSKA

**Presentata da:**  
GIUSEPPE ANTONIO  
RECUPERO

Sesta Sessione  
Anno Accademico 2019/2020



*Tutto matematica è.  
Questo diceva io a studenti quando in liceo insegnavo.  
Se cosa con numeri esprimere non può io, quella cosa non esiste.  
Se lo stesso a esistere si ostina, chiedete al Prof di filosofia e me non seccate.*

Professor Knock,  
Dylan Dog n.125, “Tre per Zero”, Febbraio 1997



# Introduzione

La rappresentazione di un oggetto o una superficie come una mesh triangolata ha vaste applicazioni in ambito industriale, di reverse engineering, design, realtà virtuale, videoludico. La mesh può essere generata attraverso modellizzazione geometrica oppure ottenuta da misurazioni 3D di oggetti reali, ad esempio tramite scanner o sonar.

Queste misurazioni sono inevitabilmente soggette ad errori, legati sia alla sensibilità degli strumenti, sia ai processi di digitalizzazione dei dati, che comportano una perturbazione della localizzazione dei vertici della triangolazione. Dunque, prima che la mesh sia pronta per essere elaborata, è necessario svolgere un processo di rimozione degli errori, anche detto *denoising* o *fairing*.

Questo procedimento può agire direttamente sui vertici, correggendone la posizione, oppure seguire uno sviluppo a due fasi, modificando prima l'orientamento delle normali per poi, a partire da queste, ridefinire l'intera mesh.

La difficoltà principale del denoising sta nel rimuovere il rumore e, al contempo, preservare correttamente le caratteristiche geometriche della superficie, cioè la forma, le dimensioni, gli spigoli, gli angoli, le aree curve e piatte.

Nel corso degli anni, il problema di denoising è stato affrontato tramite due differenti approcci, nati in origine nel campo dell'analisi delle immagini. Il primo ricostruisce la mesh attraverso processi iterativi guidati da equazioni differenziali a derivate parziali (EDP). Alcuni esempi di questo tipo sono dati dal Laplace fairing, dal Mean Curvature Flow e dagli algoritmi di filtraggio delle normali. Il secondo approccio è di tipo variazionale: il processo di denoising si traduce nella risoluzione di un problema di minimo legato ad un'opportuno funzionale che dipende dai vertici e/o dalle normali alla superficie, composto da un termine di fedeltà ai dati iniziali e da un termine regolarizzante o di penalità che rafforza la qualità della forma della superficie da ricostruire. Mentre il termine di fedeltà è solitamente definito tramite una norma  $\ell_2$ , diverse proposte sono state sviluppate e analizzate per quanto riguarda il termine regolarizzante.

A grandi linee, il rumore produce un aumento della curvatura, ovvero della variazione del campo normale alla mesh. Il termine di penalità ha l'obiettivo di ridurla ed è quindi definito come una funzione  $\rho(t)$  non negativa, nulla nell'origine, applicata a  $\nabla N$ , cioè al gradiente della normale.

Una prima proposta per definire la funzione  $\rho$  è data dalla norma  $\ell_0$ , non continua, che misura la sparsità del vettore  $\nabla N$  contando i coefficienti non nulli. I modelli costruiti in questo modo, come [5], hanno il difetto di alterare le regioni lisce della superficie, appiattendole o generando falsi spigoli. Inoltre, se il livello di rumore è alto, le mesh ricostruite hanno una triangolazione poco uniforme, con molte facce sovrapposte o triangoli degeneri.

Un'alternativa è data dai modelli di variazione totale, anche detta *total variation* (TV), come ad esempio [7]. L'uso della norma  $\ell_1$ , continua e convessa, fa sì che il problema di minimo abbia un'unica soluzione. In tal caso, il processo di denoising genera effetto staircase e produce tagli negli angoli e negli spigoli della superficie.

In questa tesi, il termine di penalità, che insieme al termine di fedeltà in norma  $\ell_2$  forma la funzione costo da ottimizzare, è stato definito tramite una funzione non-convessa  $\phi(t; a)$  il cui grafico approssima la norma  $\ell_0$ .

A causa della non convessità e non derivabilità di  $\phi(t; a)$ , la funzione costo non ha un unico minimo e non è differenziabile. Il problema di ottimizzazione viene dunque risolto applicando l'Alternating Direction Method of Multipliers (ADMM), che permette di ricondurre un complesso problema di ottimizzazione in una sequenza iterativa di problemi di minimizzazione più semplici.

Il lavoro di tesi si è proposto di analizzare il problema del mesh denoising nell'ambito del geometry processing, di definire una nuova formulazione variazionale, proporre e realizzare un metodo di ottimizzazione per la risoluzione del denoising di superfici rappresentate da mesh triangolate.

In particolare, nel Capitolo 1 presentiamo una breve analisi dei principali modelli di denoising di superfici, sviluppati sia tramite EDP sia con metodi variazionali, evidenziandone pregi e difetti.

Nel Capitolo 2 studiamo dal punto di vista teorico alcuni metodi di ottimizzazione che ci serviranno per la risoluzione del problema di minimo legato al modello. In particolare, analizziamo la discesa gradiente, il metodo di Newton e il metodo BFGS per funzioni differenziabili e i metodi ALM e ADMM per funzioni non differenziabili.

Nel Capitolo 3 presentiamo il nostro modello variazionale, approfondendo i motivi della scelta della funzione  $\phi(t; a)$  come termine di penalità, e mostriamo la discretizzazione e la risoluzione tramite ADMM.

Infine, nel Capitolo 4, osserviamo le proprietà di convergenza dell'algoritmo e confrontiamo i risultati di denoising con quelli ottenuti da altri modelli di ricostruzione, dal punto di vista sia qualitativo sia quantitativo.

# Indice

<b>Introduzione</b>	<b>iii</b>
<b>1 Modelli matematici per il denoising di superfici</b>	<b>1</b>
1.1 Mesh denoising: EDP . . . . .	2
1.1.1 Laplace fairing . . . . .	2
1.1.2 Mean Curvature flow . . . . .	3
1.1.3 Prefiltraggio delle normali . . . . .	4
1.2 Mesh denoising: metodi variazionali . . . . .	5
<b>2 Metodi di ottimizzazione</b>	<b>7</b>
2.1 Metodi Line Search per funzioni differenziabili . . . . .	7
2.1.1 Calcolo dello step . . . . .	8
2.1.2 La discesa gradiente . . . . .	9
2.1.3 Il metodo di Newton . . . . .	10
2.1.4 Il metodo BFGS . . . . .	11
2.1.5 Proprietà di convergenza . . . . .	14
2.1.6 Convergenza: discesa gradiente . . . . .	15
2.1.7 Convergenza: Metodo di Newton . . . . .	18
2.1.8 Convergenza: BFGS . . . . .	19
2.2 Metodi per funzioni non differenziabili . . . . .	21
2.2.1 Il metodo ADMM . . . . .	22
2.2.2 Un'applicazione al denoising di immagini: il modello TV- $\ell_2$ . . . . .	23
<b>3 Il modello variazionale</b>	<b>27</b>
3.1 Formulazione . . . . .	27
3.1.1 Il termine di penalità . . . . .	28
3.2 La discretizzazione . . . . .	31
3.3 La minimizzazione del problema (3.1) . . . . .	32
3.3.1 La consistenza delle normali . . . . .	34
3.3.2 Il problema per (3.16) . . . . .	35
3.3.3 Il problema per (3.17) . . . . .	37
3.3.4 Il problema per (3.18) . . . . .	37
3.4 L'algoritmo . . . . .	45

<b>4</b>	<b>Validazione del modello</b>	<b>47</b>
4.1	Modello matematico del degrado della superficie . . . . .	48
4.1.1	Figure di merito per valutare l'errore . . . . .	50
4.2	Convergenza empirica . . . . .	51
4.3	Dipendenza dal parametro $\lambda$ . . . . .	53
4.4	L'influenza del parametro $a$ . . . . .	54
4.5	Confronto con altri modelli . . . . .	56
	<b>Conclusione</b>	<b>69</b>

# Capitolo 1

## Modelli matematici per il denoising di superfici

Si consideri una superficie 3D, ottenuta misurando o scannerizzando un qualsiasi oggetto reale. La limitata sensibilità degli strumenti di misura e il processo di digitalizzazione dei dati producono degli errori nella rappresentazione della superficie.

Sia i modelli di fairing, sia quelli di denoising hanno l'obiettivo di correggere questi errori. Tuttavia hanno un comportamento differente sulle aree di discontinuità della superficie, come spigoli, angoli creste o altri dettagli particolari. Il fairing punta infatti a ridurle o eliminarle, rendendo la superficie il più possibile liscia e regolare. Invece il denoising ha l'obiettivo di preservarli accuratamente, in modo da ottenere un'approssimazione il più possibile precisa della superficie originale.

Una superficie in  $\mathbb{R}^3$  può essere rappresentata in modo esplicito o implicito.

Nelle rappresentazioni esplicite la superficie è definita tramite i parametri  $u, v \in \mathbb{R}$  come l'insieme di punti  $(x(u, v), y(u, v), z(u, v))$ , con  $x, y, z$  funzioni più o meno regolari. I modelli che puntano a tali ricostruzioni, ad esempio le spline [1], permettono di ottenere superfici molto regolari, ma diventano inefficaci per trattare oggetti dalla topologia complessa, a causa della difficoltà nel definire tali parametrizzazioni.

Questo problema non si presenta nei modelli che sfruttano le rappresentazioni implicite, in cui la superficie è definita come un insieme di livello di un'opportuna funzione scalare su  $\mathbb{R}^3$ .

Una rappresentazione esplicita della superficie è data da una sua approssimazione discreta, nella forma di una mesh triangolata definita dai suoi vertici, edge e facce.

Assumiamo dunque che la superficie sia una varietà bidimensionale  $\mathcal{M}$  in  $\mathbb{R}^3$  e indichiamo con  $(\Omega, S)$  una carta di  $\mathcal{M}$ , con  $\Omega$  un aperto in  $\mathbb{R}^2$  e  $S$  la corrispondente parametrizzazione. Chiamiamo  $M$  la mesh triangolare che approssima  $\mathcal{M}$ , definita da un insieme di vertici  $V = \{v_1, \dots, v_{n_V}\}$ , un insieme di triangoli  $T = \{\tau_1, \dots, \tau_{n_T}\}$  e un insieme di edge  $E = \{e_1, \dots, e_{n_E}\}$ . Inoltre denotiamo con  $e_{ij}$  l'edge che unisce i vertici  $v_i$  e  $v_j$ . Il vantaggio di questa definizione consiste nella semplicità di costruzione e/o modifica della mesh, anche per superfici con topologia complessa.

Il rumore generato dal processo di scansione della superficie è interpretabile come una perturbazione della posizione dei vertici. Dunque, ponendo  $V^*$  l'insieme dei vertici

originali, assumiamo che i vertici in input  $V^0$  siano esprimibili nella forma

$$V^0 = V^* + \vec{E} , \quad (1.1)$$

dove  $\vec{E}$  indica il rumore. Solitamente, la scansione è effettuata da un dispositivo di rilevamento che ruota intorno all'oggetto misurando le distanze da un'insieme di suoi punti e combinando le informazioni ottenute nelle varie posizioni. È dunque ragionevole supporre che su ogni vertice l'errore abbia direzione quasi parallela alla normale alla superficie. Una tale assunzione vale anche nel caso in cui l'oggetto in esame sia degradato, ovvero abbia perso parte del materiale. Anche in tal caso la perturbazione è diretta lungo la normale.

Definiamo quindi il rumore  $\vec{e}_i$  sul vertice  $v_i$  come

$$\vec{e}_i = c_i(\gamma\vec{n}_i + \delta\vec{d}_i) , \quad (1.2)$$

dove  $\vec{n}_i$  è la normale esterna al vertice  $v_i$ .  $\vec{d}_i$  invece introduce aleatorietà anche nella direzione. In particolare, generato un vettore random  $\vec{r}_i$ ,  $\vec{d}_i$  è la sua proiezione lungo il piano ortogonale a  $\vec{n}_i$ , normalizzata.  $c_i$  è uno scalare ottenuto da una distribuzione normale con media 0 e deviazione standard  $L^{(m)}$ , cioè la lunghezza media degli edge della mesh. Dunque il rumore è una combinazione lineare dei due vettori tra loro ortogonali  $\vec{n}_i$  e  $\vec{d}_i$ , avente come coefficienti i due parametri  $\gamma, \delta \in [0, 1)$ .

I metodi di denoising rimuovono il rumore correggendo la posizione dei vertici. Due approcci sono stati sviluppati per risolvere il problema.

Il primo consiste nel costruire modelli in cui l'evoluzione dei vertici è guidata da equazioni differenziali a derivate parziali (EDP). In generale, questi modelli hanno un forte effetto regolarizzante, ma sono imprecisi nel preservare la forma o i bordi della superficie

Il secondo approccio è di tipo variazionale. Il problema si traduce nella minimizzazione di un'opportuna funzione, dipendente dai vertici e/o dalle normali della mesh.

## 1.1 Mesh denoising: EDP

### 1.1.1 Laplace fairing

Il modello più semplice è il **Laplace fairing**, di tipo diffusivo, definito tramite l'equazione del calore:

$$\frac{\partial S}{\partial t} = \nabla^2 S ,$$

dove  $\nabla^2$  è il laplaciano. La risoluzione avviene tramite differenze finite, con un processo iterativo del tipo

$$S^{k+1} = S^k + \lambda \nabla^2 S^k .$$

Questa equazione ha un forte effetto regolarizzante sulla superficie, ma produce anche un'indesiderata contrazione. Diverse varianti sono state proposte per evitare questo

fenomeno. Taubin [4], ad esempio, introduce un nuovo parametro  $\mu < -\lambda$  e aggiunge uno step di dilatazione:

$$\begin{aligned}\tilde{S}^k &= S^k + \lambda \nabla^2(S^k), \\ S^{k+1} &= \tilde{S}^k + \mu \nabla^2(\tilde{S}^k).\end{aligned}$$

Le possibili discretizzazioni del laplaciano sono definite sull'insieme dei vertici  $V$ . La più semplice è data dal cosiddetto *umbrella operator*

$$U(v_i) = \frac{1}{\sum_{j \in N_1(i)} \omega_j} \sum_{j \in N_1(i)} \omega_j v_j - v_i,$$

dove  $N_1(i)$  è l'insieme dei vertici adiacenti al vertice  $v_i$  e i pesi  $\omega_j$  possono essere scelti come costanti o anch'essi dipendenti dai vertici  $v_j$ .

Una variante, descritta in [3], modifica questa definizione in modo da tenere conto anche della lunghezza degli edge:

$$L(v_i) = \frac{2}{E} \sum_{j \in N_1(i)} \frac{v_j - v_i}{|e_{ij}|},$$

con  $E = \sum_{j \in N_1(i)} |e_{ij}|$ , dove  $|e_{ij}|$  è la lunghezza dell'edge  $e_{ij}$ .

In generale, si osserva che il Laplace fairing ottiene ottimi risultati sulle aree irregolari, mentre ha comportamenti indesiderati sulle superfici piate, poiché ogni vertice  $v_i$ , invece di restare immutato, tende ad essere attratto verso il baricentro dei vertici ad esso adiacenti. Questo accade perché il laplaciano ha componenti sia normali sia tangenziali e la sua approssimazione, anche su superfici piate, non è quasi mai nulla.

Questo fenomeno non avviene se si usa un modello per cui l'evoluzione dei vettori avviene solo lungo la direzione normale alla superficie. È il caso del curvature flow [2].

### 1.1.2 Mean Curvature flow

L'equazione differenziale che guida l'evoluzione dei punti nel modello di curvature flow è

$$\frac{\partial v_i}{\partial t} = -\bar{\mathbf{k}}_i n_i,$$

dove  $\bar{\mathbf{k}} = (\mathbf{k}_1 + \mathbf{k}_2)/2$  è la curvatura media della superficie e  $n_i$  è la normale al vertice  $v_i$ .

La curvatura normale  $\bar{\mathbf{k}}_i n_i$  è calcolata come  $\nabla A/2A$ , dove  $A$  è l'area della regione formata dai triangoli cui appartiene il vertice  $x_i$ , da cui si ricava la seguente discretizzazione:

$$-\bar{\mathbf{k}}_i n_i = \frac{1}{4A} \sum_{j \in N_1(i)} (\cot \alpha_j + \cot \beta_j)(v_j - v_i),$$

con  $\alpha_j$  e  $\beta_j$  angoli opposti all'edge  $e_{ij}$  nei triangoli che lo contengono.

In questo modello, la regolarizzazione dipende solo dalle proprietà geometriche intrinseche della superficie e nelle aree piate i vertici restano fissi, senza subire slittamenti. Inoltre il modello preserva la curvatura, nelle aree dove questa è costante.

### 1.1.3 Prefiltraggio delle normali

Altri modelli sono costituiti da due step distinti.

Il primo step ha la funzione di modificare le normali e rimuovere il rumore. È solitamente definito in modo che l'aggiornamento della normale  $n_i$  dipenda da una combinazione lineare delle normali alle facce adiacenti. Segnaliamo alcuni esempi. In [15]:

$$n'_i = \text{normalizza} \left( \sum_{j \in \mathcal{D}(i)} h_j n_j \right) \quad \text{con} \quad h_j = \begin{cases} (n_i \cdot n_j - T)^2 & \text{se } n_i \cdot n_j \geq T \\ 0 & \text{altrimenti.} \end{cases}$$

dove  $T$  è un parametro positivo costante, utile a stabilire il peso di ogni normale  $n_j$  in base alla sua posizione rispetto a  $n_i$ . In [18] invece

$$n'_i = \text{normalizza} \left( \sum_{j \in \mathcal{D}(i)} \omega_{ij} n_j \right) \quad \text{con} \quad \omega_{ij} = \xi_{ij} W_c(\|c_i - c_j\|) W_s(\|n_i - n_j\|) .$$

dove  $W_c$  e  $W_s$  sono funzioni decrescenti,  $c_i$  indica il centroide della faccia  $i$ -esima. I parametri positivi  $\xi_{ij}$  sono solitamente definiti come le aree della faccia  $j$ -esima.

Il secondo step interviene invece sui vertici, ricostruendoli in modo che ad ogni faccia sia associata la corrispondente normale, calcolata al passo precedente. In entrambi gli esempi di cui sopra ([15],[18]), è usato lo stesso aggiornamento:

$$v'_i = v_i + \frac{1}{|\mathcal{D}(v_i)|} \sum_{j \in \mathcal{D}(v_i)} n'_j (n'_j \cdot (c_j - v_i)) .$$

Un terzo esempio di processo di denoising a due step è sviluppato in [31]. In questo caso, il primo step coinvolge la curvatura normale  $\vec{H}_i = \bar{\mathbf{k}}_i \vec{n}_i$ , ed è definito a partire dall'equazione differenziale

$$\begin{cases} \frac{\partial \vec{H}_i}{\partial t} = L_w \vec{H}_i + \lambda (\vec{H}_i^0 - \vec{H}_i) \\ \vec{H}(0) = \vec{H}^0 \end{cases} \quad (1.3)$$

con  $\vec{H}^0$  curvatura normale iniziale e  $L_w$  l'operatore differenziale di Laplace-Beltrami non locale così definito:

$$L_w \vec{H}_i = \sum_{j \in N_1(i)} (\vec{H}_j - \vec{H}_i) W_{ij} w_{ij} .$$

I termini  $W_{ij}$  sono scelti in modo da pesare la distanza tra le normali di due facce adiacenti. In particolare, per ogni  $i = 1, \dots, n_V$  e per ogni  $j \in N_1(i)$ ,

$$W_{ij} = \frac{\bar{W}_{ij}}{\sum_{j \in N_1(i)} \bar{W}_{ij}} \quad \text{con} \quad \bar{W}_{ij} = e^{-\|\vec{H}_i - \vec{H}_j\|_2^2 / \sigma}$$

Il secondo step ricostruisce la posizione dei vertici risolvendo il problema ai minimi quadrati

$$V^* = \underset{V}{\operatorname{argmin}} \|LV - \vec{H}^*\|_2^2 + \lambda \|V^0 - V\|_2^2 ,$$

con  $\vec{H}^*$  soluzione di (1.3) e  $L$  l'operatore che ad una mesh di vertici  $V$  associa il corrispondente campo di curvatura media normale.

In generale, questi modelli spesso non preservano correttamente gli spigoli e gli angoli della superficie. Inoltre sono poco precisi nel caso di triangolazioni non uniformi.

## 1.2 Mesh denoising: metodi variazionali

Un altro approccio al mesh denoising è dato dai metodi variazionali. Il problema viene tradotto nella minimizzazione di una funzione dipendente dai vertici o dalle normali della mesh. Solitamente, questa funzione presenta almeno due termini: uno di fedeltà ai dati iniziali e uno regolarizzante. Le diverse definizioni possibili danno origine ai vari modelli.

Un primo esempio è la minimizzazione  $\ell_0$  presentata in [5], definita dal problema

$$V^* = \operatorname{argmin}_V \{ \|V - V^0\|_2^2 + \|D(V)\|_0 \} ,$$

con  $V^0$  vertici della mesh in input e  $D$  un opportuno operatore differenziale. La pseudo-norma  $\ell_0$  conta il numero di coordinate non nulle del vettore cui è applicata, misurando quindi la sparsità del termine  $D(V)$ . La sua natura combinatoria rende difficile trattare teoricamente la risoluzione del problema di minimo. Il modello riesce a rimuovere il rumore mantenendo la forma dell'oggetto, ma produce un effetto a gradini, anche detto *staircase*, sulle regioni curve lisce.

Questo difetto viene leggermente migliorato diminuendo l'influenza dei dati iniziali sul risultato finale, definendo il termine di fedeltà tramite una norma  $\ell_1$  invece che  $\ell_2$ , come avviene ad esempio in [6].

Altri modelli utilizzano invece lo schema a due passi, risolvendo un problema di minimo sulle normali per poi aggiornare la posizione dei vertici. Spesso il termine che agisce come regolarizzatore del campo delle normali fa uso della variazione totale, anche detta *total variation* (TV), ovvero è definito tramite una norma  $\ell_1$ .

Ad esempio, in [7], l'operatore  $TV$  è definito come una combinazione lineare delle componenti di  $(\nabla \mathbf{N})$ , in questo modo:

$$(TV)(\mathbf{N}) := \sum_e l_e |(\nabla \mathbf{N})|_e = \sum_e l_e |[\mathbf{N}]_e| ,$$

dove  $[\mathbf{N}]_e$  discretizza il gradiente come la differenza tra le normali alle facce cui l'edge  $e$  appartiene. Il conseguente problema di minimo è

$$\mathbf{N}^* = \operatorname{argmin}_{\mathbf{N} \in C_N} \left\{ R_{wvtv}(\nabla \mathbf{N}) + \frac{\alpha}{2} \|\mathbf{N} - \mathbf{N}^0\|_{V_M}^2 \right\} ,$$

dove

$$R_{wvtv}(\nabla \mathbf{N}) := \sum_e w_e l_e \sqrt{\sum_{i=1}^3 |[N_i]_e|^2} .$$

Con questa definizione del termine regolarizzante, la funzione da minimizzare è convessa e dunque esiste un'unica soluzione al problema di minimo. Altre varianti usano invece un operatore di ordine maggiore, come ad esempio in [8]:

$$R_{vhow}(\nabla^2 \mathbf{N}) := \sum_e w_e l_e \sqrt{\sum_{i=1}^3 (\nabla^2 N_i|_e)^2} .$$

Questi due modelli producono buoni risultati ma, in caso di rumore particolarmente intenso, possono produrre tagli negli spigoli o causare effetto staircase, poiché tendono a minimizzare l'area coperta dalla mesh. Il risultato migliora definendo i pesi  $w_e$  in modo dinamico.

Per quanto riguarda il secondo passo, cioè l'aggiornamento dei vertici date le normali  $\mathbf{N}_\tau^*$  modificate, in gran parte dei modelli si ottiene risolvendo tramite discesa del gradiente il seguente problema di ottimizzazione:

$$V^* = \operatorname{argmin}_V \left\{ \frac{\eta}{2} \|V - V^0\|^2 + \sum_{\tau} \sum_{(v_i, v_j) \in \tau} s_{\tau} (\mathbf{N}_{\tau}^* \cdot (v_i - v_j))^2 \right\} .$$

Tuttavia questo approccio non tiene conto dell'orientamento delle normali, confondendo  $\mathbf{N}_{\tau}^*$  e  $-\mathbf{N}_{\tau}^*$  e causando errori in caso di rumore eccessivo. A questo proposito, in [8] è stata introdotta una nuova formulazione, che massimizza l'allineamento tra la normale effettiva  $\mathbf{N}_{\tau}$  e la sua ricostruzione:

$$V^* = \operatorname{argmin}_V \left\{ \frac{\eta}{2} \|V - V^0\|^2 - \sum_{\tau=(v_i, v_j, v_k)} s_{\tau} \mathbf{N}_{\tau} \cdot \left( \frac{(v_j - v_i) \times (v_k - v_i)}{\|(v_j - v_i) \times (v_k - v_i)\|} \right) \right\} .$$

La risoluzione di questi problemi di minimo fa uso dell'Augmented Lagrangian Method: introducendo una nuova variabile  $t$  e un moltiplicatore di Lagrange  $\rho$ , si ottengono due diversi problemi di minimo più semplici da risolvere.

Una combinazione di queste diverse idee è sviluppata in [9]. La funzione da minimizzare è

$$\mathbf{N}^* = \operatorname{argmin}_{\mathbf{N}} \{ \alpha E_{f_2} + \beta E_{f_1} + \gamma E_{TV} + \delta E_{aho} \} .$$

Essa combina, nell'ordine, un termine di fedeltà in norma  $\ell_2$ , un termine di fedeltà in norma  $\ell_1$ , un termine regolarizzante basato sulla total variation e un termine regolarizzante anisotropico, che corregge l'eventuale effetto scala.

La risoluzione, tramite Augmented Lagrangian Method, è seguita dall'aggiornamento dei vertici, sviluppato come in [8].

I modelli citati finora ottengono ottimi risultati nella maggioranza dei casi. Tuttavia, in caso di elevati livelli di rumore o di triangolazioni dalla forma poco uniforme, non preservano correttamente le proprietà geometriche della superficie. Le ultime ricerche, ad esempio [11], puntano a risolvere la questione attraverso un pre-filtraggio iniziale della superficie, risolvendo due problemi di minimo che tengono conto non solo della posizione dei vertici, ma anche delle proprietà geometriche delle facce, come angoli e aree. Il risultato è una triangolazione più uniforme e regolare, da cui parte poi il vero e proprio processo di denoising.

# Capitolo 2

## Metodi di ottimizzazione

Sia data una funzione  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  e un sottinsieme  $C$  di  $\mathbb{R}^n$ . I metodi di ottimizzazione hanno l'obiettivo di risolvere il problema di minimizzazione

$$\min_{x \in C} f(x) . \quad (2.1)$$

In base alle proprietà della funzione  $f$  e del vincolo  $C$ , sono definiti diversi metodi per la risoluzione del problema.

Nelle prossime sezioni presentiamo, in particolare:

- discesa gradiente, metodo di Newton e metodo BFGS, per funzioni  $f$  differenziabili non vincolate, cioè con  $C = \mathbb{R}^n$ ;
- Augmented Lagrange Method e ADMM, nel caso di funzioni  $f$  non differenziabili.

### 2.1 Metodi Line Search per funzioni differenziabili

Consideriamo il problema di minimo non vincolato

$$\min\{f(x), x \in \mathbb{R}^n\} , \quad (2.2)$$

con  $f \in C^2(\mathbb{R}^n)$ . La regolarità della funzione permette di mettere in relazione un minimo locale  $x^*$  con il gradiente e l'hessiana calcolati in quel punto.

**Proposizione 2.1.1** (Prima Condizione di Ottimalità). *Se  $x^*$  è un minimo locale e  $f$  è differenziabile in un intorno aperto di  $x^*$ , allora  $\nabla f(x^*) = 0$ .*

**Proposizione 2.1.2** (Seconda Condizione di Ottimalità). *Se  $x^*$  è un minimo locale e l'hessiana  $\nabla^2 f$  esiste ed è continua in un intorno aperto di  $x^*$ , allora  $\nabla f(x^*) = 0$  e  $\nabla^2 f(x^*)$  è semidefinita positiva.*

Mentre queste due proposizioni descrivono condizioni necessarie, la seguente tratta invece una condizione sufficiente.

**Proposizione 2.1.3.** *Se  $\nabla^2 f$  è continua in un intorno aperto di  $x^*$ , con  $\nabla^2 f(x^*)$  definita positiva, e se  $\nabla f(x^*) = 0$ , allora  $x^*$  è un minimo locale stretto di  $f$ .*

Queste proprietà, che riguardano sempre minimi locali, sono i principi basilari su cui sono stati costruiti i metodi di minimizzazione non vincolata per funzioni differenziabili. Tali metodi non assicurano dunque di poter trovare un minimo globale.

Il problema non si pone nel caso in cui un minimo locale sia anche un minimo globale, come nel caso di funzioni convesse.

**Proposizione 2.1.4.** *Se  $f$  è convessa, ogni minimo locale è anche un minimo globale. Se inoltre  $f$  è differenziabile e  $\nabla f(x^*) = 0$ , allora  $x^*$  è un minimo globale. Se  $f$  è strettamente convessa, questo minimo è unico.*

Quando non è possibile trovare un punto di minimo  $x^*$  tramite un'espressione esplicita, la strategia più comune consiste nel costruire un processo iterativo che definisca una successione di punti  $(x_k) \subseteq \mathbb{R}^n$  in modo che converga a  $x^*$ .

Il punto iniziale  $x_0 \in \mathbb{R}^n$  è solitamente scelto come prima stima del minimo in base alle informazioni già note sul problema da risolvere. I successivi aggiornamenti sono così definiti:

$$x_{k+1} = x_k + \alpha_k p_k \quad (2.3)$$

$p_k$  è un vettore in  $\mathbb{R}^n$  e indica la direzione di evoluzione dell'algoritmo. La sua costruzione sfrutta le informazioni sulla funzione  $f$  nei punti precedenti  $x_0, \dots, x_k$  in modo che, lungo questa direzione, il valore di  $f$  diminuisca. Le due metodologie più importanti per definirlo sono la discesa del gradiente e il metodo di Newton.

$\alpha_k$ , invece, è uno scalare positivo e rappresenta lo step di discesa. Se scelto costante, diventa un parametro dell'algoritmo. Altrimenti, può essere definito in maniera adattiva ad ogni passo, risolvendo il problema di minimo unidimensionale  $\min\{f(x_k + \alpha p_k), \alpha > 0\}$  attraverso un procedimento di ricerca lineare esatta o inesatta.

### 2.1.1 Calcolo dello step

Lo step di discesa ideale è lo scalare  $\alpha > 0$  che minimizza  $f(x_k + \alpha p_k)$ . Calcolarne esattamente il valore è, in generale, molto costoso. Spesso quindi, invece che risolvere il problema in maniera esatta,  $\alpha$  è scelto all'interno di un insieme di possibili valori, in modo che rispetti determinate proprietà.

La più semplice condizione da imporre su  $\alpha$  è che lo spostamento da  $x_k$  a  $x_{k+1}$  riduca effettivamente il valore di  $f$ . In termini formali

$$f(x_k + \alpha p_k) < f(x_k)$$

Si può verificare, tuttavia, che questa proprietà non è sufficiente per ottenere convergenza, sia perché è soddisfatta da tutti gli step abbastanza piccoli, sia perché non presenta nessuna richiesta sull'ampiezza della diminuzione della funzione  $f$ .

Una primo miglioramento consiste nell'imporre che la riduzione di  $f$  sia maggiore di un termine proporzionale alla lunghezza dello step  $\alpha$  e al valore della derivata direzionale  $\nabla f(x_k)^T p_k$ . Si ottiene così la Condizione di Sufficiente Decrescita, o Armijo Condition.

**Definizione 2.1.1** (Armijo Condition). Fissato  $c_1 \in (0, 1)$ , uno step  $\alpha > 0$  rispetta la Armijo Condition se

$$f(x_k + \alpha p_k) \leq f(x_k) + c_1 \alpha \nabla f(x_k)^T p_k$$

Come in precedenza, anche in questo caso la proprietà è verificata per tutti gli  $\alpha$  sufficientemente piccoli. Un possibile modo per escludere tali valori è aggiungere la Condizione di Curvatura.

**Definizione 2.1.2** (Condizione di Curvatura). Sia  $c_2 \in (c_1, 1)$ , con  $c_1$  scelto come in (2.1.1). Uno step  $\alpha > 0$  rispetta la Condizione di Curvatura se

$$\nabla f(x_k + \alpha_k p_k)^T p_k \geq c_2 \nabla f(x_k)^T p_k$$

In questa disuguaglianza, il secondo membro è proporzionale alla derivata direzionale di  $f$  nel punto iniziale  $x_k$ , mentre il primo membro è la derivata rispetto  $\alpha$  di  $f(x_k + \alpha p_k)$ . La richiesta è quindi che lo step definisca un punto  $x_k + \alpha p_k$  in cui la velocità di discesa della funzione  $f$  sia sufficientemente piccola in modulo.

**Definizione 2.1.3** (Condizioni di Wolfe). Uno step  $\alpha > 0$  verifica le Condizioni di Wolfe se rispetta contemporaneamente la Condizione di Armijo e la Condizione di Curvatura.

Si può verificare che per ogni funzione regolare inferiormente limitata, esiste uno step che soddisfa le Condizioni di Wolfe.

Un metodo alternativo per escludere step troppo piccoli, senza dover aggiungere la Condizione di Curvatura, ha a che fare con la scelta dell'insieme dei possibili step e con l'ordine con cui testarli. La Condizione di Armijo è infatti sufficiente per avere convergenza, a patto che lo step scelto sia abbastanza grande. L'idea è quella di testare gli step in ordine decrescente, secondo un algoritmo di backtracking.

---

### Algorithm 1 Backtracking Line Search

---

Fissati  $\bar{\alpha} = 1$ ,  $\rho \in (0, 1)$ ,  $c \in (0, 1)$ ;

$\alpha \leftarrow \bar{\alpha}$

**Finché**  $f(x_k + \alpha p_k) > f(x_k) + c\alpha \nabla f(x_k)^T p_k$

$\alpha \leftarrow \rho\alpha$

**Fine**

$\alpha_k \leftarrow \alpha$

---

Riepilogando, si sceglie un primo possibile step  $\bar{\alpha} = 1$ . Se questo soddisfa la Condizione di Armijo, allora viene scelto come step  $\alpha_k$ . In caso contrario, viene ridotto di un fattore  $\rho$  e testato nuovamente. L'algoritmo termina dopo un numero finito di passi, poiché la condizione è soddisfatta da tutti gli step sufficientemente piccoli.

## 2.1.2 La discesa gradiente

L'idea più semplice per definire  $p_k$  è sceglierlo come la direzione di maggior discesa della funzione  $f$ , che si può ricavare a partire dal Teorema di Taylor.

**Teorema 2.1.5** (Teorema di Taylor). Sia  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  differenziabile in modo continuo e sia  $p \in \mathbb{R}^n$ . Allora esiste  $t \in (0, 1)$  tale che

$$f(x + p) = f(x) + \nabla f(x + tp)^T p \tag{2.4}$$

Inoltre, se  $f$  è due volte differenziabile, vale

$$\nabla f(x+p) = \nabla f(x) + \int_0^1 \nabla^2 f(x+tp)p dt \quad (2.5)$$

ed esiste  $t \in (0, 1)$  tale che

$$f(x+p) = f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(x+tp)p \quad (2.6)$$

Dunque la derivata di  $f$  lungo  $p$  nel punto  $x_k$  è il valore  $\nabla f(x_k)^T p$ . Da questa espressione si ricava una caratterizzazione delle direzioni di discesa, in base alla loro posizione rispetto al gradiente.

**Definizione 2.1.4.**  $p_k$  è una direzione di discesa per la funzione  $f$  nel punto  $x_k$  se  $\nabla f(x_k)^T p_k < 0$ , cioè se  $\cos \theta_k < 0$ , con  $\theta_k$  angolo compreso tra i due vettori  $p_k$  e  $\nabla f(x_k)$ .

La direzione unitaria di maggior discesa della funzione  $f$  è dunque quella opposta al gradiente:

$$p_k = -\nabla f(x_k) \quad (2.7)$$

La discesa gradiente è un metodo del primo ordine, poiché questa costruzione non richiede nessuna informazione o ipotesi sulle derivate al secondo ordine della funzione  $f$ , con notevoli vantaggi dal punto di vista computazionale.

### 2.1.3 Il metodo di Newton

Il metodo di Newton consente di definire la direzione di discesa sfruttando non solo le informazioni sulle derivate del primo ordine della funzione  $f$ , ma anche sulle derivate del secondo ordine, in particolare sulla matrice Hessiana.

Supponiamo che  $f$  sia di classe  $C^3(\mathbb{R}^n)$ . Dallo sviluppo di Taylor al terzo ordine della funzione  $f(x+p)$  nel punto  $x$  si ricava la seguente osservazione.

**Osservazione 2.1.6.** Se  $f \in C^3(\mathbb{R}^n)$ , allora

$$f(x+p) = f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(x)p + O(\|p\|^3) \quad (2.8)$$

Ciò significa che per spostamenti  $p$  sufficientemente piccoli in norma, la funzione

$$m(p) := f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(x)p \quad (2.9)$$

è una buona approssimazione di  $f(x+p)$ . In tal caso, il vettore  $p$  che minimizza  $f(x+p)$  può essere stimato minimizzando la funzione  $m(p)$ .

Si supponga adesso che il termine quadratico  $p^T \nabla^2 f(x)p$  sia sempre positivo, cioè che la matrice hessiana  $\nabla^2 f(x)$  sia definita positiva.

**Proposizione 2.1.7.** Sia  $m(p)$  la funzione definita in (2.9). Se  $\nabla^2 f(x)$  è definita positiva, allora

$$\operatorname{argmin}_{p \in \mathbb{R}^n} m(p) = -(\nabla^2 f(x))^{-1} \nabla f(x) =: p^N \quad (2.10)$$

*Dimostrazione.*  $m(p)$  è una funzione quadratica e differenziabile. Poiché  $\nabla^2 f(x)$  è definita positiva, è anche convessa e dunque ha un unico minimo, che chiamiamo  $p^N$ . Per la prima condizione di ottimalità (2.1.1), vale  $\nabla m(p^N) = 0$  cioè

$$\nabla f(x) + \nabla^2 f(x)p^N = 0 \quad (2.11)$$

Dunque  $p^N = -(\nabla^2 f(x))^{-1} \nabla f(x)$  □

Con queste ipotesi su  $f$ , si verifica facilmente che  $p^N$  è una direzione di discesa. Infatti

$$\nabla f(x)^T p^N = -\nabla f(x)^T (\nabla^2 f(x))^{-1} \nabla f(x) < 0 \quad (2.12)$$

poiché si suppone che  $\nabla^2 f(x)$  sia definita positiva.

Il metodo di Newton sfrutta quindi le informazioni sulla matrice hessiana della funzione  $f$  per definire la direzione di discesa  $p^N$ .

Questa definizione comporta due difficoltà. La prima è il calcolo dell'hessiana, spesso complicato dal punto di vista teorico e computazionale. La seconda è la risoluzione del sistema lineare. Il metodo BFGS è una modifica del metodo di Newton che permette di evitare questi problemi calcolando solo delle approssimazioni della matrice hessiana o della sua inversa.

## 2.1.4 Il metodo BFGS

Il metodo BFGS (dal nome dei suoi creatori Broyden, Fletcher, Goldfarb e Shanno) costruisce una matrice  $B_k$  che approssima l'hessiana  $\nabla^2 f(x_k)$  riducendo il costo computazionale per il calcolo della direzione  $p_k$ .

Si consideri l'espressione (2.5) ottenuta nel Teorema di Taylor (2.1.5), con l'ipotesi  $f \in C^2(\mathbb{R}^n)$ . Aggiungendo e sottraendo il termine  $\nabla^2 f(x)p$  si ottiene

$$\begin{aligned} \nabla f(x+p) &= \nabla f(x) + \nabla^2 f(x)p + \int_0^1 [\nabla^2 f(x+tp) - \nabla^2 f(x)] p dt = \\ &= \nabla f(x) + \nabla^2 f(x)p + o(\|p\|) \end{aligned} \quad (2.13)$$

Poniamo  $x = x_k$  e  $p = x_{k+1} - x_k$ . Si ha

$$\nabla f(x_{k+1}) = \nabla f(x_k) + \nabla^2 f(x_k)(x_{k+1} - x_k) + o(\|x_{k+1} - x_k\|) \quad (2.14)$$

Per approssimazione, vale

$$\nabla^2 f(x_k)(x_{k+1} - x_k) \approx \nabla f(x_{k+1}) - \nabla f(x_k) \quad (2.15)$$

Si cerca quindi una matrice  $B_{k+1}$  che rispetti l'equazione della secante, cioè una condizione analoga all'espressione (2.15).

**Definizione 2.1.5** (Equazione della secante). Siano  $s_k = x_{k+1} - x_k$  e  $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$ . Una matrice  $B$  verifica l'equazione della secante se

$$B s_k = y_k \quad (2.16)$$

Inoltre, si impone che la matrice  $B_{k+1}$  sia simmetrica definita positiva. Affinché i vincoli siano compatibili, è necessario che  $s_k^T y_k > 0$ .

**Proposizione 2.1.8.** *Per ogni  $k$ , esiste una matrice  $B$  simmetrica definita positiva tale che  $Bs_k = y_k$  se vale una delle seguenti condizioni:*

1.  $f$  è strettamente convessa;
2.  $B_0$  è simmetrica definita positiva e lo step  $\alpha$  rispetta le Condizioni di Wolfe.

*Dimostrazione. Ipotesi 1.* Se  $f$  è strettamente convessa il prodotto scalare  $s_k^T y_k$  è positivo per definizione.

*Ipotesi 2.*  $\alpha$  rispetta le Condizioni di Wolfe, dunque anche la Condizione di Curvatura: fissato  $c_2 < 1$ , per ogni  $k$  vale

$$\nabla f(x_{k+1})^T p_k \geq c_2 \nabla f(x_k)^T p_k$$

Dimostriamo la tesi per induzione su  $k$ .  $B_0$  è simmetrica definita positiva, dunque  $p_0$  è una direzione di discesa. Infatti

$$\nabla f(x_0)^T p_0 = -\nabla f(x_0)^T B_0^{-1} \nabla f(x_0) < 0$$

Inoltre vale

$$\begin{aligned} y_0^T s_0 &= (\nabla f(x_1) - \nabla f(x_0))^T (x_1 - x_0) = (\nabla f(x_1) - \nabla f(x_0))^T \alpha_0 p_0 \geq \\ &\geq (c_2 \nabla f(x_0) - \nabla f(x_0))^T \alpha_0 p_0 = (c_2 - 1) \nabla f(x_0)^T \alpha_0 p_0 > 0 \end{aligned} \quad (2.17)$$

Esiste quindi  $B_1$  simmetrica definita positiva che rispetta l'equazione della secante.

Supponiamo adesso che una tale matrice esista anche al passo  $k$ . Con un ragionamento analogo al precedente si mostra che  $p_k$  è una direzione di discesa e che  $y_k^T s_k > 0$ . Dunque i vincoli sono compatibili anche al passo  $k + 1$  ed esiste una soluzione  $B_{k+1}$ .  $\square$

Quando i vincoli sono compatibili, non sono comunque sufficienti per determinare univocamente  $B_{k+1}$ . Infatti una matrice simmetrica definita positiva ha  $n(n+1)/2$  gradi di libertà, mentre l'equazione della secante impone  $n$  condizioni. Dunque, tra tutte le matrici che rispettano tali proprietà, si sceglie quella più vicina alla matrice del passo precedente  $B_k$ .

**Definizione 2.1.6.** Data  $B_k$  simmetrica definita positiva, si indica con  $B_{k+1}$  l'unica matrice che risolve il problema di minimo

$$\min_B \|B - B_k\| \quad (2.18)$$

vincolato alle condizioni  $B = B^T$ ,  $B$  definita positiva e  $Bs_k = y_k$ .

Diversi tipi di norma matriciale possono essere usate in (2.18). Per semplicità, si consideri la norma di Frobenius pesata  $\|A\|_W = \|W^{1/2}AW^{1/2}\|_F$ , con  $W = \bar{G}_k^{-1}$  e  $\bar{G}_k$  l'hessiana media così definita:

$$\bar{G}_k = \left[ \int_0^1 \nabla^2 f(x_k + \tau s_k) d\tau \right]$$

Questa scelta permette di trovare un'espressione esplicita per  $B_{k+1}$ , tramite la Formula DFP (Davidon-Fletcher-Powell).

**Definizione 2.1.7** (Formula DFP). Siano  $s_k = x_{k+1} - x_k$  e  $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$ . Sia  $\rho_k = 1/(y_k^T s_k)$ . La Formula DFP per la costruzione di  $B_{k+1}$  è data dall'espressione

$$B_{k+1} = (I - \rho_k y_k s_k^T) B_k (I - \rho_k s_k y_k^T) + \rho_k y_k y_k^T \quad (2.19)$$

Una volta costruite le matrici  $B_k$ , le direzioni  $p_k$  si ottengono risolvendo il sistema lineare  $B_k p_k = -\nabla f(x_k)$ . In molti casi quest'ultima operazione risulta computazionalmente costosa, in generale dell'ordine di  $O(n^3)$ . Per questo motivo è preferibile calcolare non le matrici  $B_k$  ma direttamente le loro inverse  $H_k = B_k^{-1}$ . Ciò può essere fatto in due modi.

La prima possibilità è usare la formula di Sherman-Morrison-Woodbury per passare dalla definizione di  $B_{k+1}$  in (2.19) alla seguente espressione:

$$H_{k+1} = H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \frac{s_k s_k^T}{y_k^T s_k} \quad (2.20)$$

La seconda possibilità è molto simile alla definizione del problema di minimo in 2.1.6, ma applicata alla matrice  $H_k$ . La differenza sta soltanto nell'esprimere in maniera inversa il vincolo legato all'equazione della secante.

**Definizione 2.1.8.** Data  $H_k$  simmetrica definita positiva, si indica con  $H_{k+1}$  l'unica matrice che risolve il problema di minimo

$$\min_H \|H - H_k\| \quad (2.21)$$

vincolato alle condizioni  $H = H^T$ ,  $H$  definita positiva e  $s_k = H y_k$ .

Usando la stessa norma matriciale definita in precedenza, l'unica soluzione è data dalla formula BFGS.

**Definizione 2.1.9** (Formula BFGS). Siano  $s_k = x_{k+1} - x_k$  e  $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$ . Sia  $\rho_k = 1/(y_k^T s_k)$ . La Formula BFGS per la costruzione di  $H_{k+1}$  è data dall'espressione

$$H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T \quad (2.22)$$

Le tre espressioni (2.19), (2.20) e (2.22) permettono di costruire l'approssimazione dell'hessiana o della sua inversa al passo  $k+1$  aggiornando le matrici del passo precedente  $k$  tramite le nuove informazioni su  $x_{k+1}$  e  $\nabla f(x_{k+1})$ . Tali aggiornamenti sono ben definiti se  $B_k$  e  $H_k$  sono non singolari e  $s_k^T y_k$  è non nullo. Inoltre, si può verificare che se  $B_k$  e  $H_k$  sono definite positive, anche  $B_{k+1}$  e  $H_{k+1}$  lo sono.

## 2.1.5 Proprietà di convergenza

Gli algoritmi di minimizzazione descritti si basano sul processo iterativo

$$x_{k+1} = x_k + \alpha_k p_k \quad (2.23)$$

Studiare la convergenza significa, innanzitutto, capire se effettivamente la successione  $(x_k)$  tende a raggiungere un punto di minimo per la funzione  $f$ .

**Definizione 2.1.10.** Un metodo è detto globalmente convergente se  $\|\nabla f(x_k)\| \rightarrow 0$  per  $k \rightarrow \infty$ .

I metodi globalmente convergenti, in generale, non garantiscono la convergenza a un punto di minimo, ma solamente a un punto stazionario. La globale convergenza di un metodo è strettamente legata alla scelta della direzione  $p_k$  e dello step  $\alpha_k$ .

**Teorema 2.1.9.** *Si consideri l'iterazione (2.23), con  $x_0$  punto iniziale. Sia  $p_k$  una direzione di discesa, con angolo  $\theta_k$  rispetto al vettore  $-\nabla f(x_k)$ . Sia  $\alpha_k$  uno step che rispetta le condizioni di Wolfe (2.1.3). Sia  $\mathcal{A}$  un insieme aperto contenente l'insieme di livello  $\mathcal{L} := \{x : f(x) \leq f(x_0)\}$ . Si supponga che  $f$  sia inferiormente limitata, differenziabile in  $\mathcal{A}$  e che il suo gradiente sia lipschitziano in  $\mathcal{A}$ . Allora*

$$\sum_{k \geq 0} \cos^2(\theta_k) \|\nabla f(x_k)\|^2 < \infty \quad (2.24)$$

*Dimostrazione.* Si veda il Teorema 3.2 in [20] □

**Corollario 2.1.10.** *Siano date le ipotesi del Teorema 2.1.9. Se inoltre  $\cos(\theta_k) > \delta > 0$  per ogni  $k$ , allora la successione  $(x_k)$  è globalmente convergente.*

*Dimostrazione.* Se  $k \rightarrow \infty$ , vale  $\cos^2(\theta_k) \|\nabla f(x_k)\|^2 \rightarrow 0$ . Poiché per ipotesi  $\cos(\theta_k) \geq \delta > 0$  per ogni  $k > 0$ , allora necessariamente  $\|\nabla f(x_k)\| \rightarrow 0$ , cioè la successione è globalmente convergente. □

Dunque una condizione sufficiente per avere globale convergenza è che le direzioni di discesa  $p_k$  siano lontane dall'essere ortogonali al gradiente.

Nelle precedenti sezioni abbiamo descritto tre modi per definire  $p_k$ : discesa gradiente, metodo di Newton e BFGS.

La seguente osservazione mostra che la discesa gradiente è globalmente convergente.

**Osservazione 2.1.11.** *Siano  $f, \mathcal{L}, \mathcal{A}$  tali da verificare le ipotesi del Teorema 2.1.9. Si consideri l'iterazione  $x_{k+1} = x_k + \alpha_k p_k$ , con  $\alpha_k$  che rispetti le Condizioni di Wolfe 2.1.3. Se  $p_k$  è costruita secondo il metodo del gradiente, cioè  $p_k = -\nabla f(x_k)$ , allora la successione  $(x_k)$  è globalmente convergente.*

*Dimostrazione.*  $p_k$  è una direzione di discesa per costruzione e  $\cos \theta_k = 1$  per ogni  $k$ . Dunque per il corollario 2.1.10 il metodo è globalmente convergente. □

Sia il metodo di Newton sia il metodo BFGS definiscono la direzione  $p_k$  tramite un'espressione nella forma  $p_k = -B_k^{-1}\nabla f(x_k)$ , con  $B_k$  matrice simmetrica non singolare. La seguente proposizione mostra una condizione sufficiente sulle matrici  $B_k$  per avere convergenza globale.

**Proposizione 2.1.12.** *Sia  $f$  tale da verificare le ipotesi del Teorema 2.1.9. Si consideri l'iterazione  $x_{k+1} = x_k + \alpha_k p_k$ , con  $\alpha_k$  che rispetta le Condizioni di Wolfe 2.1.3 e  $p_k = -B_k^{-1}\nabla f(x_k)$ . Se le matrici  $B_k$  sono simmetriche definite positive con un numero di condizionamento limitato, allora la successione  $(x_k)$  è globalmente convergente.*

*Dimostrazione.* Per le ipotesi su  $B_k$ , esiste  $M > 0$  tale che per ogni  $k$  vale

$$\mathcal{K}(B_k) = \|B_k^{-1}\| \|B_k\| = \frac{\lambda_{\max}(B_k)}{\lambda_{\min}(B_k)}$$

Sia  $\theta$  l'angolo compreso tra la direzione  $p_k$  e  $-\nabla f(x_k)$ . Vale

$$\begin{aligned} \cos(\theta_k) &= \frac{-\nabla f(x_k)^T p_k}{\|\nabla f(x_k)\| \|p_k\|} = \frac{\nabla f(x_k)^T B_k^{-1} \nabla f(x_k)}{\|\nabla f(x_k)\| \|B_k^{-1} \nabla f(x_k)\|} \geq \frac{\lambda_{\min}(B_k^{-1}) \|\nabla f(x_k)\|^2}{\|B_k^{-1}\| \|\nabla f(x_k)\|^2} = \\ &= \frac{\lambda_{\min}(B_k^{-1})}{\|B_k^{-1}\|} \geq \frac{\lambda_{\min}(B_k^{-1})}{\lambda_{\max}(B_k^{-1})} = \frac{\lambda_{\min}(B_k)}{\lambda_{\max}(B_k)} = \frac{1}{\mathcal{K}(B_k)} \geq \frac{1}{M} \end{aligned} \quad (2.25)$$

Dunque, per il Corollario 2.1.10, la successione  $(x_k)$  è globalmente convergente.  $\square$

## 2.1.6 Convergenza: discesa gradiente

Nell'osservazione 2.1.11, abbiamo visto che la discesa gradiente è globalmente convergente, se applicata a funzioni con gradiente lipschitziano e se gli step  $\alpha_k$  verificano le condizioni di Wolfe. Quindi  $x^*$ , punto di accumulazione della successione  $x_k$ , è un punto di minimo o un punto di sella.

In realtà, si può verificare che  $x^*$  è quasi certamente un punto di minimo. In termini più precisi, definiamo  $W^s(x^*) = \{x \text{ t.c. } \lim_k g^k(x) = x^*\}$ , con  $g(x) := x - \alpha \nabla f(x)$ . Si osserva che se  $x^*$  è un punto di sella, allora  $W^s(x^*)$  ha misura nulla.

La dimostrazione è una conseguenza dello Stable Manifold Theorem [21]

**Teorema 2.1.13** (Stable Manifold Theorem). *Sia  $E$  uno spazio di Banach,  $U \subseteq E$  un intorno di 0. Sia  $\phi : U \rightarrow E$  un diffeomorfismo locale di classe  $C^r$ , con punto fisso 0. Dato  $D_\phi(0)$  il differenziale calcolato in 0, chiamiamo  $E_s$  lo spazio generato dagli autovettori corrispondenti agli autovalori minori o uguali a 1 e  $E_u$  lo spazio generato dagli autovettori corrispondenti agli autovalori maggiori di 1. Allora:*

- $E = E_s \oplus E_u$ ;
- Esiste una varietà centrale stabile locale, cioè un disco  $W_{loc}^{cs}$  tangente a  $E_s$  in 0;
- Esiste un intorno  $B$  di 0 tale che  $\phi(W_{loc}^{cs}) \cap B \subset W_{loc}^{cs}$  e  $\bigcap_{k=0}^{\infty} \phi^{-k}(B) \subset W_{loc}^{cs}$ .

In termini pratici, il teorema afferma che se  $\phi$  è un diffeomorfismo che agisce su un intorno di un punto critico  $x^*$ , allora esiste una varietà  $W_{loc}^{cs}$  che contiene non solo il punto critico, ma anche tutti i punti che convergono a  $x^*$ . La dimensione di  $W_{loc}^{cs}$  è pari al numero di autovalori minori di 1 della matrice  $D_\phi(x^*)$ . Se inoltre il punto di partenza sta nell'intorno  $B$  di  $x^*$ , allora tutte le successive iterazioni appartengono a  $W_{loc}^{cs} \cap B$ . Nel nostro caso, come osservato nel Teorema 4.5 in [25], il diffeomorfismo è l'operatore  $g$ .

**Osservazione 2.1.14.** *L'operatore  $g(x) = x - \alpha \nabla f(x)$ , con  $\alpha < 1/L$ , è un diffeomorfismo.*

Applicando lo Stable Manifold Theorem, otteniamo il seguente corollario.

**Corollario 2.1.15.** *Se  $x^*$  è un punto di sella, allora  $W^s(x^*)$  ha misura nulla.*

*Dimostrazione.* Poiché  $g(x) = x - \alpha \nabla f(x)$ , si ha che  $D_g(x) = I - \alpha \nabla^2 f(x)$ . Osserviamo il numero di autovalori positivi di  $\nabla^2 f(x^*)$ . Se  $x^*$  è un punto di sella, almeno uno degli autovalori è non positivo, dunque  $W_{loc}^{cs}$  ha codimensione strettamente positiva ed è un insieme di misura nulla.

Supponiamo che  $x$  converga a  $x^*$ , cioè  $x \in W^s(x^*)$ . Ciò significa che  $g^k(x) \in B$  per tutti i  $k$  maggiori di un certo  $k_0$ . Dunque  $g^k(x) \in \bigcap_{h=0}^{\infty} g^{-h}(B) \subseteq W_{loc}^{cs}$ . Vale

$$W^s(x^*) \subseteq \bigcup_{l \geq 0} g^{-l} \left( \bigcap_{k=0}^{\infty} g^{-k}(B) \right) \subseteq \bigcup_{l \geq 0} g^{-l}(W_{loc}^{cs}) \quad (2.26)$$

I diffeomorfismi mappano insiemi di misura nulla in insiemi di misura nulla. Dunque  $W^s(x^*)$  ha misura nulla.  $\square$

Possiamo adesso dimostrare che la discesa gradiente converge quasi sicuramente ad un punto di minimo.

**Corollario 2.1.16.** *Sia  $C$  l'insieme dei punti di sella di una funzione  $f$  di classe  $C^2$ . Sia  $x_{k+1} = g(x_k) = x_k - \alpha \nabla f(x_k)$ , con  $\alpha < 1/L$ . Se  $x_k$  converge e  $C$  è un insieme di cardinalità numerabile, allora  $Pr(\lim_k x_k = x^*) = 1$ , con  $x^*$  punto di minimo.*

*Dimostrazione.* L'insieme dei punti iniziali  $x_0$  che convergono a un punto di sella è  $\bigcup_{x \in C} W^s(x)$ , unione numerabile di insiemi di misura nulla, dunque ha misura nulla. Perciò la probabilità che  $x^*$  sia un punto di sella è 0 e  $x^*$  è quasi certamente un punto di minimo.  $\square$

Per studiare la velocità di convergenza, rafforziamo le ipotesi di regolarità sulla funzione  $f$ . Nel teorema 2.1.9 avevamo già supposto che in un insieme di livello  $\mathcal{L} = \{x : f(x) \leq f(x_0)\}$ , il gradiente fosse lipschitziano, cioè

$$\|\nabla f(x)\| \geq L|f(x) - f(x^*)|$$

Consideriamo un'ipotesi più restrittiva.

**Definizione 2.1.11.** Una funzione  $f$  con punto stazionario  $x^*$  rispetta la disuguaglianza di Polyak-Lojasiewicz se esiste  $\mu > 0$  tale che per ogni  $x$  in un intorno  $U$  di  $x^*$  vale

$$\frac{1}{2}\|\nabla f(x)\|^2 \geq \mu(f(x) - f(x^*)) \quad (2.27)$$

Riscrivendo (2.27) come

$$\|\nabla f(x)\| \geq \sqrt{2\mu}(f(x) - f(x^*))^{1/2}$$

osserviamo che questa condizione è più forte della lipschianità, poiché richiede che il gradiente cresca più velocemente rispetto a una funzione quadratica. Implica inoltre automaticamente che  $x^*$  sia un minimo.

Con queste ipotesi, la discesa gradiente converge con velocità lineare.

**Teorema 2.1.17.** *Sia  $f$  con gradiente lipschitziano, con un punto di minimo  $x^*$ . Supponiamo che  $f$  rispetti la disuguaglianza di Polyak-Lojasiewicz. Allora l'iterazione  $x_{k+1} = x_k + \alpha_k \nabla f(x_k)$ , con  $\alpha_k$  ottenuto tramite ricerca lineare esatta, converge a  $x^*$  con velocità lineare. In particolare, vale*

$$f(x_k) - f(x^*) \leq \left(1 - \frac{\mu}{L}\right)^k (f(x_0) - f(x^*))$$

*Dimostrazione.* La lipschianità di  $f$  implica la disuguaglianza

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|x - y\|^2$$

Ricordando che  $\alpha_k$  minimizza  $f(x_k + \alpha \nabla f(x_k))$ , vale

$$\begin{aligned} f(x_{k+1}) - f(x_k) &= f(x_k - \alpha_k \nabla f(x_k)) - f(x_k) \leq f\left(x_k - \frac{1}{L} \nabla f(x_k)\right) - f(x_k) \leq \\ &\leq \left\langle \nabla f(x_k), -\frac{1}{L} \nabla f(x_k) \right\rangle + \frac{L}{2} \left\| \frac{1}{L} \nabla f(x_k) \right\|^2 \leq \\ &\leq -\frac{1}{L} \|\nabla f(x_k)\|^2 + \frac{1}{2L} \|\nabla f(x_k)\|^2 = -\frac{1}{2L} \|\nabla f(x_k)\|^2 \leq \\ &\leq -\frac{1}{2L} 2\mu(f(x_k) - f(x^*)) \end{aligned} \quad (2.28)$$

dove l'ultima disuguaglianza sfrutta l'ipotesi di Polyak-Lojasiewicz. Sottraendo a entrambi i membri  $f(x^*)$  e riordinando i termini, si ottiene infine

$$f(x_{k+1}) - f(x^*) \leq \left(1 - \frac{\mu}{L}\right) (f(x_k) - f(x^*)) \leq \left(1 - \frac{\mu}{L}\right)^k (f(x_0) - f(x^*)) \quad (2.29)$$

□

## 2.1.7 Convergenza: Metodo di Newton

Supponiamo che la successione  $x_{k+1} = x_k + \alpha_k p_k$  converga ad un punto di minimo  $x^*$ . Il seguente teorema mostra una condizione sufficiente sulle direzioni  $p_k$  affinché la convergenza sia superlineare.

**Teorema 2.1.18.** *Sia  $f \in C^2(\mathbb{R}^n)$  con un punto di minimo locale  $x^*$  in cui l'hessiana  $\nabla^2 f(x^*)$  è definita positiva. Si consideri la successione  $x_{k+1} = x_k + \alpha_k p_k$ , con  $p_k$  direzione di discesa e  $\alpha_k$  che soddisfa la Condizione di Wolfe 2.1.3 con  $c_1 \leq 1/2$ . Se  $x_k \rightarrow x^*$  e se le direzioni  $p_k$  soddisfano*

$$\lim_{k \rightarrow \infty} \frac{\|\nabla f(x_k) + \nabla^2 f(x_k) p_k\|}{\|p_k\|} = 0 \quad (2.30)$$

allora:

- lo step  $\alpha_k = 1$  è ammissibile per tutti i  $k$  maggiori di un certo  $k_0$ ;
- se  $\alpha_k = 1$  per ogni  $k > k_0$ , la successione  $(x_k)$  converge a  $x^*$  con velocità superlineare.

Nel caso in cui le direzioni  $p_k$  siano definite tramite il metodo di Newton, la frazione in (2.30) è nulla per ogni  $k$ , poiché  $p_k = -\nabla^2 f(x_k)^{-1} \nabla f(x_k)$ . Dunque, se il punto di partenza  $x_0$  è sufficientemente vicino a  $x^*$ , il metodo di Newton ha convergenza superlineare e  $\alpha_k$  può essere fissato costante pari a 1.

Inoltre, il prossimo teorema mostra che la velocità diventa quadratica se la funzione  $f$  ha una maggiore regolarità. In particolare, viene richiesto che l'hessiana sia lipschitziana in  $x^*$ , cioè che esistano un intorno  $U$  di  $x^*$  e una costante  $L > 0$  tali che

$$\|\nabla^2 f(x) - \nabla^2 f(x^*)\| \leq L \|x - x^*\|$$

per ogni  $x \in U$ .

**Teorema 2.1.19.** *Sia  $f \in C^2(\mathbb{R}^n)$  con un punto di minimo locale  $x^*$  in cui l'hessiana  $\nabla^2 f(x^*)$  è definita positiva. Si supponga che l'hessiana  $\nabla^2 f(x)$  sia Lipschitziana in un intorno di  $x^*$ . Si consideri l'iterazione  $x_{k+1} = x_k + p_k^N$ , con  $p_k^N = -\nabla^2 f(x_k)^{-1} \nabla f(x_k)$ . Allora:*

- se il punto di partenza  $x_0$  è sufficientemente vicino a  $x^*$ , allora la successione  $(x_k)$  converge a  $x^*$ ;
- la velocità di convergenza di  $(x_k)$  è quadratica;
- la successione delle norme  $(\|\nabla f(x_k)\|)$  converge a 0 con velocità quadratica.

*Dimostrazione.* Sfruttando la definizione di  $x_{k+1}$ , la condizione di ottimalità  $\nabla f(x^*) = 0$  e il Teorema di Taylor 2.1.5, vale

$$\begin{aligned} x_{k+1} - x^* &= x_k + p_k^N - x^* = x_k - x^* - \nabla^2 f(x_k)^{-1} \nabla f(x_k) = \\ &= \nabla^2 f(x_k)^{-1} (\nabla^2 f(x_k)(x_k - x^*) - (\nabla f(x_k) - \nabla f(x^*))) = \\ &= \nabla^2 f(x_k)^{-1} \left( \nabla^2 f(x_k)(x_k - x^*) - \int_0^1 \nabla^2 f(x_k + t(x^* - x_k))(x_k - x^*) dt \right) \end{aligned} \quad (2.31)$$

Dunque si ha che

$$\begin{aligned}
& \|\nabla^2 f(x_k)(x_k - x^*) - (\nabla f(x_k) - \nabla f(x^*))\| = \\
& = \left\| \int_0^1 [\nabla^2 f(x_k) - \nabla^2 f(x_k + t(x^* - x_k))] (x_k - x^*) dt \right\| \leq \\
& \leq \int_0^1 \|\nabla^2 f(x_k) - \nabla^2 f(x_k + t(x^* - x_k))\| \|x_k - x^*\| dt \leq \\
& \leq \|x_k - x^*\|^2 \int_0^1 Lt dt = \frac{1}{2}L\|x_k - x^*\|, \tag{2.32}
\end{aligned}$$

dove  $L$  è la costante di Lipschitz per l'hessiana  $\nabla^2 f(x)$  per tutti gli  $x$  vicini a  $x^*$ . Poiché  $\nabla^2 f(x^*)$  è non singolare e  $f \in C^2(\mathbb{R}^n)$ , esiste  $r > 0$  tale che per tutti gli  $x_k$  con  $\|x_k - x^*\| \leq r$  si ha che l'hessiana  $\nabla^2 f(x_k)$  è anch'essa non singolare e vale la stima

$$\|\nabla^2 f(x_k)^{-1}\| \leq 2\|\nabla^2 f(x^*)^{-1}\| \tag{2.33}$$

Per tali  $x_k$  vale quindi

$$\begin{aligned}
\|x_{k+1} - x^*\| & \leq L\|\nabla^2 f(x^*)^{-1}\|\|x_k - x^*\|^2 = \tilde{L}\|x_k - x^*\|^2 \leq \\
& \leq \tilde{L}^{k+1}\|x_0 - x^*\|^2 \tag{2.34}
\end{aligned}$$

Scegliendo  $x_0$  in modo che  $\|x_0 - x^*\| \leq \min\{r, 1/(2\tilde{L})\}$ , si ha che  $x_k$  converge a  $x^*$  con velocità quadratica.

Osserviamo adesso la convergenza di  $\|\nabla f(x_k)\|$ . Poiché  $\nabla f(x_k) + \nabla^2 f(x_k)p_k^N = 0$ , si ha che

$$\begin{aligned}
\|f(x_{k+1})\| & = \|\nabla f(x_{k+1}) - \nabla f(x_k) - \nabla^2 f(x_k)p_k^N\| = \\
& = \left\| \int_0^1 \nabla^2 f(x_k + tp_k^N)(x_{k+1} - x_k) dt - \nabla^2 f(x_k)p_k^N \right\| \leq \\
& \leq \int_0^1 \|\nabla^2 f(x_k + tp_k^N) - \nabla^2 f(x_k)\| \|p_k^N\| dt \leq \\
& \leq \frac{1}{2}L\|p_k^N\|^2 \leq \frac{1}{2}L\|\nabla^2 f(x_k)^{-1}\|^2 \|\nabla f(x_k)\|^2 \leq \\
& \leq 2L\|\nabla^2 f(x^*)^{-1}\|^2 \|\nabla f(x_k)\|^2 \tag{2.35}
\end{aligned}$$

Dunque anche la successione delle norme dei gradienti converge a 0 con velocità quadratica.  $\square$

## 2.1.8 Convergenza: BFGS

Nella sezione 2.1.4 abbiamo visto che se le matrici  $B_k$  sono simmetriche definite positive, con numero di condizionamento limitato, allora  $p_k$  è una direzione di discesa per ogni  $k$  e il metodo BFGS converge globalmente. Supponiamo che  $x^*$ , punto di accumulazione della successione  $(x_k)$ , sia un punto di minimo locale.

La velocità di convergenza è legata sia alla regolarità della funzione  $f$  sia all'accuratezza delle approssimazioni  $B_k$  rispetto alle matrici hessiane  $\nabla^2 f(x_k)$ .

Supponiamo che il punto iniziale  $x_0$  sia sufficientemente vicino al punto di minimo locale  $x^*$ . In particolare, scegliamo  $x_0$  in modo che l'insieme di livello  $\mathcal{L} = \{x \in \mathbb{R}^n \mid f(x) \leq f(x_0)\}$  sia convesso.

Nel teorema 2.1.18, abbiamo osservato che la velocità diventa superlineare se vale (2.30). Nel caso del metodo BFGS, grazie alla definizione della direzione di discesa, questa condizione diventa

$$\lim_{k \rightarrow \infty} \frac{\|(B_k - \nabla^2 f(x_k))p_k\|}{\|p_k\|} = 0 \quad (2.36)$$

Dunque la successione  $(x_k)$  può avere convergenza superlineare anche se  $B_k$  non converge a  $\nabla^2 f(x_k)$ . Infatti è sufficiente che  $B_k$  sia un'approssimazione sempre più accurata dell'hessiana lungo la direzione  $p_k$ . Il teorema seguente dimostra che la condizione (2.36) è non solo sufficiente, ma anche necessaria, se definiamo gli step  $\alpha_k$  tutti costanti pari a 1.

**Teorema 2.1.20.** *Sia  $\mathcal{L}$  un insieme aperto convesso in  $\mathbb{R}^n$  contenente  $x^*$ , punto di minimo di una funzione  $f \in C^2(\mathcal{L})$ , con hessiana  $\nabla^2 f(x^*)$  definita positiva. Sia  $x_0 \in \mathcal{L}$  tale che la successione  $x_{k+1} = x_k - B_k^{-1} \nabla f(x_k)$  appartiene a  $\mathcal{L}$  e converge a  $x^*$ . Allora  $(x_k)$  converge con velocità superlineare se e solo se vale (2.36).*

*Dimostrazione.* Si veda il Teorema 2.2 in [22]. □

Il seguente corollario (Cor. 2.3, [22]) analizza invece il caso generale con  $\alpha_k$  variabili.

**Corollario 2.1.21.** *Sia  $f$  una funzione che soddisfi le stesse ipotesi del teorema 2.1.20 sull'insieme  $\mathcal{L}$ . Sia  $x_{k+1} = x_k - \alpha_k B_k^{-1} \nabla f(x_k)$ . Sia  $x_0 \in \mathcal{L}$  tale che  $(x_k) \subseteq D$  e  $x_k \rightarrow x^*$ . Se vale (2.36), allora  $x_k$  converge con velocità superlineare se e solo se  $\alpha_k \rightarrow 1$ .*

Concentriamoci adesso sulla condizione (2.36): quando è verificata? Innanzitutto, le matrici  $B_k$  devono essere ben definite, dunque  $y_k^T s_k$  deve essere positivo. Ciò vale se la funzione è sufficientemente regolare, come mostrato nel seguente lemma.

**Lemma 2.1.22.** *Sia  $f$  una funzione che soddisfi le stesse ipotesi del teorema 2.1.20 sull'insieme  $\mathcal{L}$ . Supponiamo inoltre che l'hessiana  $\nabla^2 f(x)$  sia lipschitziana in un intorno di  $x^*$ . Se  $(x_k) \subseteq D$  e converge a  $x^*$ , allora per ogni  $k \geq k_0$  vale  $y_k^T s_k > 0$  e, data  $B_{k_0}$  simmetrica definita positiva, le successive matrici  $(B_k)_{k \geq k_0}$  sono simmetriche definite positive. Inoltre esistono costanti  $\beta_3, \beta_4 > 0$  e  $\beta \in (0, 1]$  tali che*

$$\|B_{k+1} - \nabla^2 f(x^*)\|_M \leq [(1 - \beta \mu_k^2)^{1/2} + \beta_3 \sigma_k] \|B_k - \nabla^2 f(x^*)\|_M + \beta_4 \sigma_k \quad (2.37)$$

con  $\sigma_k = \max\{\|x_{k+1} - x^*\|^p, \|x_k - x^*\|^p\}$ ,  $M = \nabla^2 f(x^*)^{-1/2}$  e

$$\mu_k = \frac{\|M[B_k - \nabla^2 f(x^*)]s_k\|}{\|B_k - \nabla^2 f(x^*)\|_M \|M^{-1}s_k\|}$$

*Dimostrazione.* Si rimanda al Lemma 3.2 in [22] □

La superlinearità non è direttamente dimostrabile tramite la stima (2.37), ma lo diventa aggiungendo alle ipotesi la seguente disuguaglianza.

$$\sum_{k \geq 0} \|x_k - x^*\| < +\infty \quad (2.38)$$

**Teorema 2.1.23.** *Siano vere le ipotesi del Lemma 2.1.22. Se inoltre vale (2.38), allora*

$$\lim_{k \rightarrow \infty} \frac{(B_k - \nabla^2 f(x^*))s_k}{\|s_k\|} = 0$$

*Dimostrazione.* Si veda Teorema 3.4 in [22] □

È interessante notare che l'ultimo teorema non pone particolari vincoli riguardo alla scelta degli step  $\alpha_k$ . Le uniche condizioni sono la convergenza della serie (2.38) e che  $\alpha_k$  tenda ad 1.

In [23] viene mostrato che scegliere  $\alpha_k = 1$  per ogni  $k$  è sufficiente per avere convergenza superlineare, a patto che le inizializzazioni  $x_0$  e  $B_0$  siano abbastanza vicine al minimo  $x^*$  e alla sua hessiana  $\nabla^2 f(x^*)$ .

Anche con ipotesi più forti, ad esempio considerando  $f$  convessa e definendo gli  $\alpha_k$  tramite ricerca lineare esatta, la velocità di convergenza non aumenta, come dimostrato in [24], Teorema 2.

**Teorema 2.1.24.** *Sia  $f \in C^2(\mathbb{R}^n)$ , uniformemente convessa, con hessiana lipschitziana. Sia  $x^* \in \mathcal{L}$  il punto di minimo. Sia  $x_{k+1} = x_k - \alpha_k B_k^{-1} \nabla f(x_k)$ , con  $\alpha_k = \operatorname{argmin}_{\alpha > 0} f(x_k + \alpha p_k)$ . Allora  $x_k$  converge a  $x^*$  con velocità superlineare.*

## 2.2 Metodi per funzioni non differenziabili

Fino a questo momento abbiamo analizzato problemi di minimo per funzioni differenziabili non vincolate. Consideriamo adesso il problema di minimizzazione vincolata, con vincolo di uguaglianza lineare

$$x^* = \operatorname{argmin}\{f(x), \text{ t.c. } Ax = b\}, \quad (2.39)$$

con  $x \in \mathbb{R}^n$ ,  $A \in \mathbb{R}^{m \times n}$  e  $f: \mathbb{R}^n \rightarrow \mathbb{R}$ .

Il metodo dei moltiplicatori di Lagrange risolve il problema esprimendo il vincolo attraverso l'introduzione di un termine di penalità lineare. Si ottiene così una funzione detta Lagrangiana.

**Definizione 2.2.1.** La Lagrangiana associata al problema (2.39) è la funzione

$$L(x, \rho) = f(x) + \rho^T (Ax - b).$$

La variabile duale  $\rho$  è legata al problema duale  $\rho^* = \operatorname{argmax} g(\rho)$ , con  $g(\rho) = \inf_x L(x, \rho)$  e  $\nabla g(\rho) = Ax - b$ . Il valore ottimale di  $x$  è  $x^* = \operatorname{argmin}_x L(x, \rho^*)$ . La risoluzione è ottenuta a partire da valori iniziali  $x_0$  e  $\rho_0$ , tramite il procedimento iterativo di dual ascent:

$$x_{k+1} := \operatorname{argmin}_x L(x, \rho_k) \quad (2.40)$$

$$\rho_{k+1} := \rho_k + \alpha_k (Ax_{k+1} - b), \quad (2.41)$$

con  $\alpha_k$  step positivo. Affinché questo metodo converga alla soluzione  $x^*$  sono necessarie ipotesi piuttosto restrittive sulla funzione  $f$ .

L'Augmented Lagrangian Method (ALM), sviluppato in [29] e [30], rende il dual ascent più robusto. Innanzitutto, la lagrangiana  $L(x, \rho)$  definita in 2.2.1 viene modificata aggiungendo un nuovo termine di penalità, quadratico.

**Definizione 2.2.2.** La Lagrangiana Aumentata associata al problema (2.39) è la funzione

$$L(x, \rho; \beta) = f(x) + \rho^T(Ax - b) + \frac{\beta}{2}\|Ax - b\|_2^2,$$

dove  $\beta > 0$  è detto parametro di penalità.

L'ALM genera le successioni  $(x_k)$  e  $(\rho_k)$  mediante le iterazioni:

$$x_{k+1} := \operatorname{argmin}_x L(x, \rho_k; \beta) \tag{2.42}$$

$$\rho_{k+1} := \rho^k + \beta(Ax_{k+1} - b). \tag{2.43}$$

Il parametro  $\rho$  è usato quindi non solo per pesare il termine di penalità nella definizione di  $L(x, \rho; \beta)$ , ma anche come step relativo all'aggiornamento della variabile  $\rho$ .

L'iterazione converge sotto condizioni ben più generali rispetto al dual ascent. Inoltre  $Ax_{k+1} - b$ , cioè il termine residuo legato al vincolo, converge a 0 al tendere di  $k$  a infinito. Dunque il punto di accumulazione  $x^*$ , come desiderato, non solo è un punto di minimo per  $f$ , ma rispetta anche il vincolo  $Ax = b$ .

Nel caso in cui la funzione  $f(x)$  sia separabile, cioè esistano  $x_1, \dots, x_N$  sottovettori di  $x$ , con  $x_i \in \mathbb{R}^{n_i}$ , tali che  $f(x) = \sum_{i=1}^N f_i(x_i)$ , anche la lagrangiana è separabile, dunque la procedura di dual ascent permette di spezzare il problema di minimo (2.40) nella variabile  $x$  in  $N$  problemi di minimo nelle variabili  $x_1, \dots, x_N$ . Questa semplificazione non è possibile, invece, per l'ALM, poiché la lagrangiana aumentata presenta un termine di penalità quadratico, non separabile.

L'Alternating Direction Method of Multipliers (ADMM), originariamente proposto in [28] e sviluppato in [27], è una variante dell'ALM che permette di sfruttare la separabilità delle funzioni  $f(x)$ , semplificando la risoluzione del problema di minimo (2.42).

## 2.2.1 Il metodo ADMM

Siano  $\theta_1(\cdot)$  e  $\theta_2(\cdot)$  due funzioni e  $A$  un operatore lineare continuo. Consideriamo il problema di minimo

$$x^* = \operatorname{argmin} f(x) \quad \text{con} \quad f(x) = \theta_1(x) + \theta_2(Ax). \tag{2.44}$$

Introducendo una variabile ausiliaria  $y$ , riscriviamo (2.44) come l'equivalente problema di minimo vincolato in due variabili

$$x^* = \operatorname{argmin}\{\theta_1(x) + \theta_2(y), \text{ t.c. } y = Ax\}.$$

La relativa Lagrangiana Aumentata è la funzione

$$L(x, y, \rho; \beta) = \theta_1(x) + \theta_2(y) - \rho^T(Ax - y) + \frac{\beta}{2}\|Ax - y\|_2^2.$$

L'ADMM definisce le successioni  $(x_k)$ ,  $(y_k)$ ,  $(\rho_k)$  attraverso lo schema iterativo:

$$\begin{aligned} x_{k+1} &= \underset{x}{\operatorname{argmin}} L(x, y_k, \rho_k; \beta) \\ y_{k+1} &= \underset{y}{\operatorname{argmin}} L(x_{k+1}, y, \rho_k; \beta) \\ \rho_{k+1} &= \rho_k - \beta(Ax_{k+1} - y_{k+1}). \end{aligned} \quad (2.45)$$

A differenza del metodo dei moltiplicatori, nell'ADMM gli aggiornamenti delle variabili  $x$  e  $y$  non sono svolti in contemporanea, risolvendo il problema  $\operatorname{argmin}_{x,y} L(x, y, \rho_k; \beta)$ , ma in sequenza. Ciò permette di trattare anche funzioni separabili.

Diversi risultati di convergenza sono stati ottenuti per l'ADMM. Nel caso semplice in cui  $\theta_1$  e  $\theta_2$  siano funzioni chiuse, proprie e convesse, assumendo che la funzione  $L(x, y, \rho; 0)$  abbia un punto di sella, è stato dimostrato che:

- il residuo  $Ax_k - y_k$  tende a 0;
- la funzione  $\theta_1(x_k) + \theta_2(y_k)$  tende al minimo di  $f(x)$ ;
- la variabile duale  $\rho_k$  tende al valore ottimale  $\rho^*$ .

## 2.2.2 Un'applicazione al denoising di immagini: il modello TV- $\ell_2$

Il metodo ADMM è stato utilizzato con successo in diversi modelli variazionali nel campo dell'analisi delle immagini. Presentiamo una di queste applicazioni, sviluppata in [26].

Sia data un'immagine degradata  $b = Kx^* + \omega \in \mathbb{R}^{n^2}$ , ottenuta tramite un operatore di convoluzione  $K \in \mathbb{R}^{n^2 \times n^2}$  e un rumore additivo  $\omega \in \mathbb{R}^{n^2}$ .

L'obiettivo è definire un processo di denoising per ricostruire un'immagine il più possibile vicina all'originale  $x^*$ , supponendo di conoscere soltanto l'input  $b$  e l'operatore  $K$ .

Consideriamo il modello TV- $\ell_2$  definito dal problema di minimo

$$\hat{x} = \underset{x}{\operatorname{argmin}} f(x) \quad \text{con} \quad f(x) = \sum_{i=1}^{n^2} \|(Dx)_i\|_2 + \frac{\mu}{2}\|Kx - b\|_2^2, \quad (2.46)$$

Il termine regolarizzante misura la grandezza  $(Dx)_i \in \mathbb{R}^2$ , cioè la differenza finita al primo ordine di  $x$  nel pixel  $i$ , in direzione sia orizzontale sia verticale, da cui il nome Variazione Totale. Il termine di fedeltà ai dati iniziale misura la distanza tra la convoluzione di  $x$  e l'immagine in input  $b$ , tramite una norma  $\ell_2$ .

Introducendo le variabili ausiliarie  $y_i \in \mathbb{R}^2$ , l'espressione (2.46) si può riscrivere come il problema di minimo vincolato

$$(\hat{x}, \hat{y}) = \underset{x,y}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n^2} \|y_i\|_2 + \frac{\mu}{2}\|Kx - b\|_2^2; \text{ con } y_i = (Dx)_i \text{ per ogni } i = 1, \dots, n^2 \right\}, \quad (2.47)$$

avente come Lagrangiana aumentata la funzione

$$L(x, y, \rho; \beta) = \sum_i \left( \|y_i\| - \rho_i^T (y_i - (Dx)_i) + \frac{\beta}{2} \|y_i - Dx_i\|^2 \right) + \frac{\mu}{2} \|Kx - b\|^2. \quad (2.48)$$

Il metodo ADMM definisce quindi le successioni  $(y^k)$ ,  $(x^k)$  e  $(\rho^k)$  tramite il seguente schema iterativo:

$$\begin{aligned} y^{k+1} &= \operatorname{argmin}_y L(x^k, y, \rho^k; \beta) \\ x^{k+1} &= \operatorname{argmin}_x L(x, y^{k+1}, \rho^k; \beta) \\ \rho^{k+1} &= \rho^k - \beta(y^{k+1} - Dx^{k+1}). \end{aligned} \quad (2.49)$$

La minimizzazione in  $y$  si separa in  $n^2$  problemi bidimensionali della forma

$$y_i^{k+1} = \operatorname{argmin}_{y_i \in \mathbb{R}^2} \left\{ \|y_i\| + \frac{\beta}{2} \left\| y_i - \left( (Dx^k)_i + \frac{1}{\beta} (\rho^k)_i \right) \right\|^2 \right\}, \quad i = 1, \dots, n^2,$$

aventi come unica soluzione

$$y_i^{k+1} = \max \left\{ \left\| (Dx^k)_i + \frac{1}{\beta} (\rho^k)_i \right\| - \frac{1}{\beta}, 0 \right\} \frac{(Dx^k)_i + \frac{1}{\beta} (\rho^k)_i}{\left\| (Dx^k)_i + \frac{1}{\beta} (\rho^k)_i \right\|}, \quad i = 1, \dots, n^2.$$

La minimizzazione in  $x$  diventa un problema ai minimi quadrati, con soluzioni in  $x$  del sistema lineare

$$\left( D^T D + \frac{\mu}{\beta} K^T K \right) x = D^T \left( y^{k+1} - \frac{1}{\beta} \rho^k \right) + \frac{\mu}{\beta} K^T b.$$

Nel caso in cui ad  $x$  siano imposte condizioni al bordo periodiche, le matrici  $D^T D$  e  $K^T K$  sono diagonalizzabili e il costo computazionale per la risoluzione del sistema lineare attraverso trasformata discreta di Fourier 2D è dell'ordine di  $O(n^2 \log n)$ .

L'algoritmo termina quando la variazione relativa di  $x^k$  diventa sufficientemente piccola, cioè quando

$$\frac{\|x^{k+1} - x^k\|}{\max\{\|x^k\|, 1\}} < \epsilon.$$

Il problema (2.46) può essere esteso al caso di immagini a colori, in cui  $\bar{x} = [\bar{x}^{(1)}, \bar{x}^{(2)}, \bar{x}^{(3)}]$  è un'immagine di dimensioni  $n \times n$  dove  $\bar{x}^{(j)}$  rappresenta il  $j$ -esimo canale. In tal caso l'operatore di blurring è  $K = [K_{k,l}]_{k,l=1}^3$  di dimensioni  $3n^2 \times 3n^2$ . L'energia da minimizzare diventa

$$x^* = \operatorname{argmin}_x \sum_i \|(I_3 \otimes D_i)x\| + \frac{\mu}{2} \|Kx - b\|^2, \quad (2.50)$$

con  $I_3$  matrice identità e  $\otimes$  prodotto di Kronecker. Per semplicità, indichiamo con  $G_i = I_3 \otimes D_i$ . Anche in questo caso, è possibile applicare il metodo ADMM in modo analogo a quanto fatto nel caso  $m = 1$ . Dopo aver calcolato la lagrangiana aumentata e aver risolto i problemi di minimo del corrispondente sistema iterativo, si ottiene il seguente algoritmo:

- Per ogni  $i = 1, \dots, n^2$ , calcolo di  $y_i^{k+1} \in \mathbb{R}^6$  come

$$y_i^{k+1} = \max \left\{ \left\| G_i x^k + \frac{1}{\beta} (\rho^k)_i \right\| - \frac{1}{\beta}, 0 \right\} \frac{G_i x^k + \frac{1}{\beta} (\rho^k)_i}{\|G_i x^k + \frac{1}{\beta} (\rho^k)_i\|}; \quad (2.51)$$

- Calcolo di  $x^{k+1}$  risolvendo il sistema

$$\left( G^T G + \frac{\mu}{\beta} K^T K \right) x = G^T \left( y^{k+1} - \frac{1}{\beta} \rho^k \right) + \frac{\mu}{\beta} K^T b; \quad (2.52)$$

- Aggiornamento della variabile duale  $\rho \in \mathbb{R}^{6n^2}$  come

$$\lambda^{k+1} = \lambda^k - \beta (y^{k+1} - Gx^{k+1}). \quad (2.53)$$

In [26] è mostrato come questo metodo sia stabile, efficiente e veloce per ricostruire immagini sia in scala di grigi sia a colori.

Da queste osservazioni è possibile trarre spunto per costruire modelli variazionali per la ricostruzione di superfici. Ad esempio, un adattamento del problema (2.46) è stato studiato in [7].

Nel prossimo capitolo, presenteremo un nuovo modello per il denoising di mesh, analizzando in dettaglio come applicare l'ADMM e risolvere i conseguenti problemi di minimizzazione. Infine, nel quarto capitolo, mostreremo i risultati ottenuti, concentrandoci sulla sua convergenza e sul confronto con altri modelli presenti in letteratura.



# Capitolo 3

## Il modello variazionale

### 3.1 Formulazione

Sia  $M = (V, T, E)$  una mesh triangolata.

- $V \in \mathbb{R}^{n_V \times 3}$  è l'insieme dei suoi  $n_V$  vertici  $v_1, \dots, v_{n_V}$ .
- $E \in \mathbb{R}^{n_E \times 2}$  è l'insieme degli  $n_E$  edge  $e_1, \dots, e_{n_E}$ ; l'espressione  $e = (i, j)$  indica che l'edge  $e$  unisce i vertici  $v_i$  e  $v_j$ .
- $T \in \mathbb{R}^{n_T \times 3}$  è l'insieme delle  $n_T$  facce  $\tau_1, \dots, \tau_{n_T}$ ; l'espressione  $\tau = (i, j, k)$  indica che la faccia è formata dai vertici  $v_i, v_j$  e  $v_k$ , ordinati in senso antiorario.

Per ogni vertice  $v_i$  definiamo i relativi primo disco e primo anello come rispettivamente

$$\begin{aligned}\mathcal{D}(v_i) &= \{\tau_m \mid v_i \in \tau_m\} && \text{in ordine antiorario} \\ \mathcal{N}(v_i) &= \{v_j \mid (v_i, v_j) \in E\} && \text{in ordine antiorario}\end{aligned}$$

Sia  $V^0$  la matrice dei vertici di una superficie triangolata soggetta a rumore. Il metodo variazionale proposto ha come obiettivo la ricostruzione della mesh originale attraverso la risoluzione del problema di ottimizzazione

$$V^* \in \arg \min_{V \in \mathbb{R}^{n_V \times 3}} \mathcal{J}(V; \lambda, a)$$

$$\mathcal{J}(V; \lambda, a) = \frac{\lambda}{2} \|V - V^0\|_2^2 + \sum_{\substack{m=1 \\ \tau_m=(v_i, v_j, v_k)}}^{n_T} \phi \left( \left\| \frac{\nabla \cdot \left( (v_j - v_i) \times (v_k - v_i) \right)}{\|(v_j - v_i) \times (v_k - v_i)\|_2} \right\|_2; a \right) \quad (3.1)$$

Con abuso di notazione, usiamo  $\|\cdot\|_2$  per indicare sia la classica norma vettoriale  $\ell_2$ , sia la norma matriciale di Frobenius  $\|\cdot\|_F$ .

Il primo addendo è il termine di fedeltà. Al crescere del parametro  $\lambda > 0$ , si impone che la soluzione sia sempre più vicina ai dati iniziali.

Il secondo addendo, cioè il termine di penalità o regolarizzante, coinvolge la funzione  $\phi(t; a)$ , che sarà definita e analizzata nella prossima sezione.

### 3.1.1 Il termine di penalità

Il rumore cui è soggetta la mesh in input  $V^0$  produce, a livello geometrico, un aumento di curvatura.

Il processo di denoising consiste quindi, a grandi linee, nel diminuire la curvatura della triangolazione  $V^0$ , definita come la variazione del campo vettoriale  $\mathcal{N}$  normale alla superficie.

Poiché il campo normale è costante su ogni triangolo, possiamo pensarlo come una funzione

$$\mathcal{N} : \mathbb{R}^{n_V \times 3} \rightarrow \mathbb{R}^{n_T \times 3} \quad (3.2)$$

che ad ogni faccia  $\tau_m = (v_i, v_j, v_k)$  associa la rispettiva normale esterna. In formule,

$$\begin{aligned} \mathcal{N}(V) &:= [\mathcal{N}(V; 1), \dots, \mathcal{N}(V; n_T)]^T, \\ \text{con } \mathcal{N}(V; m) &:= \left( \frac{(v_j - v_i) \times (v_k - v_i)}{\|(v_j - v_i) \times (v_k - v_i)\|_2} \right)^T \in \mathbb{R}^3. \end{aligned} \quad (3.3)$$

Il campo normale varia solo in corrispondenza degli edge, unici punti in cui il suo gradiente sarà non nullo. Definiamo quindi  $\nabla \mathcal{N}$  come una funzione

$$(\nabla \mathcal{N})(V) : \mathbb{R}^{n_V \times 3} \rightarrow \mathbb{R}^{n_E \times 3} \quad (3.4)$$

che associa all'edge  $e_i$  la distanza tra le normali alle facce cui  $e_i$  appartiene, pesata con la lunghezza dell'edge. In formule, se  $e_i = \tau_j \cap \tau_k$ , con  $j < k$ , il corrispondente gradiente è il vettore

$$(\nabla \mathcal{N}(V))_i := \sqrt{l_i} (\mathcal{N}(V; j) - \mathcal{N}(V; k)). \quad (3.5)$$

Inoltre è utile notare che  $\mathcal{N}(V; m)$  ha norma unitaria ed è una funzione continua in  $V$ , per ogni  $m = 1, \dots, n_T$ . Queste osservazioni permettono di riscrivere il termine regolarizzante dell'energia (3.1) tramite una sommatoria sugli edge:

$$\mathcal{J}(V; \lambda, a) = \frac{\lambda}{2} \|V - V^0\|_2^2 + \sum_{j=1}^{n_E} \phi(\|(\nabla \mathcal{N}(V))_j\|_2; a) \quad (3.6)$$

Nei modelli variazionali, la funzione di regolarizzazione è svolta dal termine di penalità, costruito tramite una funzione  $\rho$  del gradiente della normale (ovvero della curvatura), non negativa e che si annulla solo quando  $\|\nabla \mathcal{N}\|_2 = 0$ .

La definizione di tale funzione risponde all'esigenza di rimuovere il rumore preservando al contempo gli spigoli e le regioni lisce della mesh originaria.

La scelta più immediata è data dalla pseudo-norma  $\ell_0$  di un vettore  $t \in \mathbb{R}^n$ :

$$\rho(t) = \|t\|_0 = \#\{i = 1, \dots, n \text{ t.c. } t_i \neq 0\}$$

Questa funzione misura direttamente la sparsità del vettore  $t$ , ma non è continua. La sua natura combinatoria rende difficile la risoluzione del problema di minimo associato. Oltretutto, questo approccio riesce a preservare gli spigoli originari, ma altera le regioni lisce della superficie, appiattendole o generando falsi spigoli.

I modelli di Total Variation usano un'altra definizione del termine di penalità, tramite la norma  $\ell_1$ :

$$\rho(t) = \|t\|_1 := |t_1| + \dots + |t_n|$$

Questa funzione è continua in  $\mathbb{R}^n$  e strettamente convessa, dunque il problema di minimo ha un'unica soluzione. Al contrario della norma  $\ell_0$ , la norma  $\ell_1$  non produce falsi edge. Tuttavia tende a generare effetto staircase e a minimizzare l'area coperta dalla mesh, producendo tagli negli angoli o negli spigoli.

Un terzo approccio consiste nel definire il termine di penalità tramite funzioni che crescono più lentamente rispetto alla norma  $\ell_1$ , o tramite funzioni limitate. Ad esempio:

- “log”:

$$\rho(t; a) = \log(1 + a|t|)/a$$

- “atan”:

$$\rho(t; a) = \begin{cases} \frac{2 \arctan\left(\frac{(2a|t|+1)}{\sqrt{3}}\right) - \frac{\pi}{6}}{a\sqrt{3}} & \text{per } a > 0, \\ |t| & \text{per } a = 0, \end{cases}$$

- “rat”:

$$\rho(t; a) = \frac{|t|}{1 + \frac{a}{2}|t|}$$

- “exp”:

$$\rho(t; a) = \frac{1 - e^{-a|t|}}{a}$$

Si considerino adesso tutte le funzioni  $\rho(t; \mu)$  continue in  $\mathbb{R}$ , derivabili in  $\mathbb{R}^*$ , con i seguenti vincoli:

- $\rho'(t; \mu) = 0$  per ogni  $t \geq \gamma\mu$ , cioè  $\rho$  è costante per  $t$  abbastanza grande;
- $\rho'(0+; \mu) = \mu$ ,

con  $\gamma > 0$  parametro di regolarità. Sia  $\mathcal{K}(\rho)$  la massima concavità della funzione  $\rho$ , cioè lo scalare

$$\mathcal{K}(\rho) := \sup_{0 < t_1 < t_2} \frac{\rho'(t_1; \mu) - \rho'(t_2; \mu)}{t_2 - t_1}$$

La funzione che minimizza  $\mathcal{K}(\rho)$ , fissati i vincoli descritti in precedenza, è detta Minimax Concave Penalty (*MCP*) [12].

**Definizione 3.1.1.** Fissato  $\gamma > 0$  parametro di regolarizzazione, la funzione Minimax Concave Penalty è

$$MCP(t; \mu) = \mu \int_0^t \left(1 - \frac{x}{\gamma\mu}\right)_+ dx \quad (3.7)$$

Questa funzione può essere pensata come una versione non convessa della norma  $\ell_1$  poiché quando  $\gamma \rightarrow +\infty$ ,  $MCP$  tende alla norma  $\ell_1$ .

Sviluppando l'integrale in (3.7), si ottiene una definizione per parti della funzione:

$$MCP(t; \mu) = \begin{cases} -\frac{1}{2\gamma} t^2 + \mu t & \text{per } t \in [0, \gamma\mu), \\ \frac{\mu^2\gamma}{2} & \text{per } t \in [\gamma\mu, +\infty), \end{cases} \quad (3.8)$$

La funzione  $\phi(t; a)$  che interviene nel modello variazionale in (3.1) è così definita:

$$\phi(t; a) = \begin{cases} -\frac{a}{2} t^2 + \sqrt{2a} t & \text{per } t \in [0, \sqrt{2/a}), \\ 1 & \text{per } t \in [\sqrt{2/a}, +\infty). \end{cases} \quad (3.9)$$

Si osserva che questa è semplicemente la funzione  $MCP$  riscalata e riparametrizzata. Lo scalare  $a > 0$  è detto parametro di concavità.

Al contrario della norma  $\ell_1$ , la funzione  $\phi(t; a)$  non è convessa, dunque non è possibile dimostrare fin da subito la convessità della funzione obiettivo  $\mathcal{J}(V; \lambda, a)$  definita in (3.1).

La figura (3.1) mostra come varia il grafico di  $\phi(t; a)$  al variare del parametro di concavità  $a$ . Si osserva che quando  $a \rightarrow +\infty$ ,  $\phi$  tende alla norma  $\ell_0$ .

La figura (3.2) confronta invece il grafico di  $\phi$  con quello della norma  $\ell_1$  e delle altre funzioni definite in precedenza.

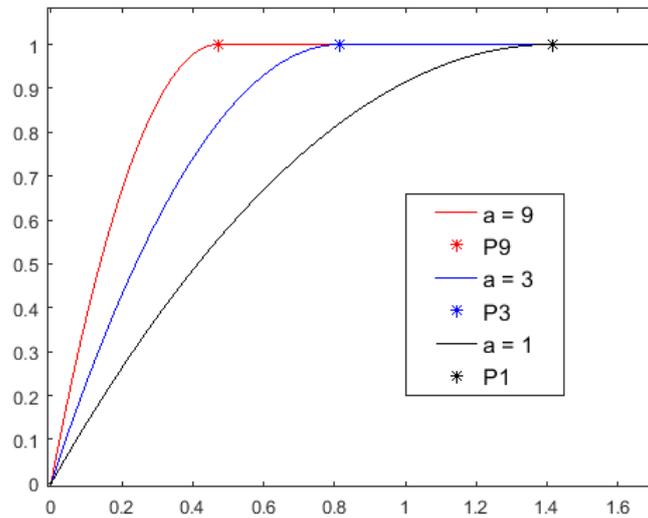


Figura 3.1: Grafici della funzione  $\phi(t, a)$ , per 3 diversi valori di  $a$ .  $P1, P3, P9$  sono i rispettivi punti con coordinate  $(\sqrt{2/a}, 1)$  che distinguono i due intervalli su cui è definita la funzione.

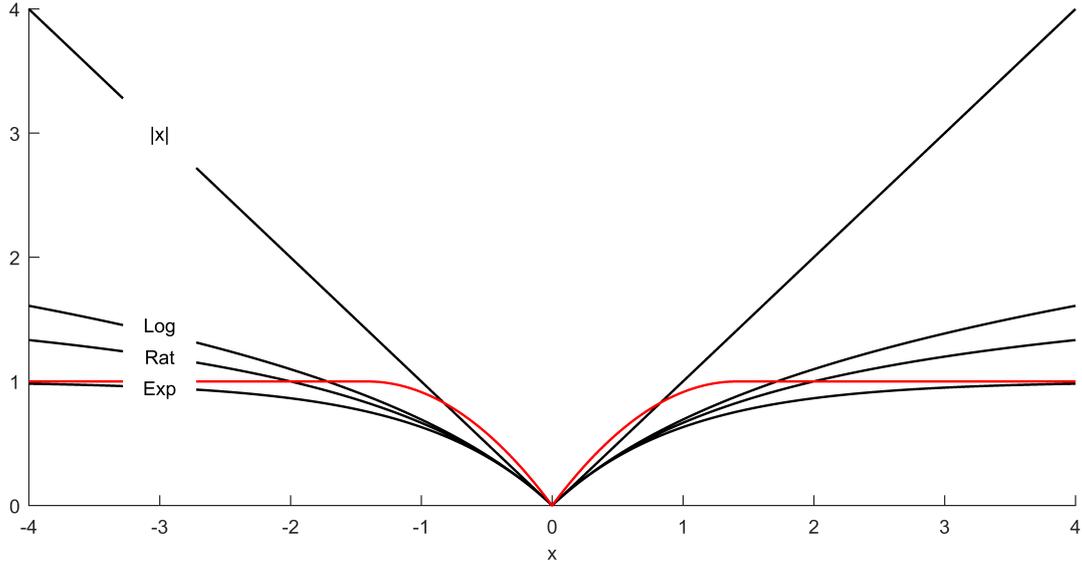


Figura 3.2: Grafici della norma  $\ell_1$  e, con  $a = 1$ , delle funzioni “log”, “atan”, “rat”, “exp”. In rosso, la funzione  $\phi(x, 1)$ .

## 3.2 La discretizzazione

L’equazione (3.1) appare già discretizzata per quanto riguarda le grandezze  $V$ ,  $V^0$  e  $T$ . L’unico termine di cui bisogna ancora definire un’approssimazione è  $\nabla \mathcal{N}$ .

Dalla sua definizione in (3.5), deduciamo che può essere descritto come il prodotto  $D \mathcal{N}$ , con  $D \in \mathbb{R}^{n_E \times n_T}$  così definita

$$D_{ij} = \begin{cases} \sqrt{l_i} & \text{se } \tau_j \cap \tau_k = e_i, k > j, \\ -\sqrt{l_i} & \text{se } \tau_j \cap \tau_l = e_i, k < j, \\ 0 & \text{altrimenti,} \end{cases} \quad (3.10)$$

con  $l_i = |e_i|$ ,  $i = 1, \dots, n_E$  lunghezze degli edge. La matrice  $D$  così costruita è sparsa, avendo solo  $2n_E$  elementi non nulli. Poiché la topologia della mesh e le relazioni di appartenenza tra vertici, edge e facce non cambiano, la matrice  $D$  può essere decomposta come prodotto  $D = L\bar{D}$ .  $\bar{D}$  è una matrice sparsa di dimensioni  $n_E \times n_T$  e costante, indipendente dalla lunghezza degli edge.  $L = \text{diag}(\sqrt{l_1}, \dots, \sqrt{l_{n_E}})$  è invece la matrice diagonale delle lunghezze degli edge.

Queste osservazioni permettono di riscrivere il termine regolarizzante dell’energia tramite una sommatoria sugli edge:

$$\mathcal{J}(V; \lambda, a) = \frac{\lambda}{2} \|V - V^0\|_2^2 + \sum_{j=1}^{n_E} \phi(\|(D \mathcal{N}(V))_j\|_2; a) \quad (3.11)$$

### 3.3 La minimizzazione del problema (3.1)

Per risolvere il problema (3.1), occorre innanzitutto osservare se l'energia  $\mathcal{J}(V; \lambda, a)$  ha un minimo.

**Osservazione 3.3.1.** *La funzione  $\|V - V^0\|_2^2$  è propria, continua, non negativa e coerciva.*

*Dimostrazione.* La funzione è quadratica, dunque chiaramente continua e non negativa. È coerciva poiché  $\|V - V^0\|_2^2 \rightarrow +\infty$  per  $\|V\|_2 \rightarrow +\infty$ . È propria poiché non raggiunge mai il valore  $-\infty$ . Infatti, il termine di fedeltà è una norma  $\ell_2$  ed è quindi limitato inferiormente da 0.  $\square$

**Proposizione 3.3.2.** *Sia  $e_h = (v_i, v_j)$ . Per ogni  $a > 0$ , la funzione*

$$\phi(\|(D\mathcal{N}(V))_h\|_2; a)$$

*è propria, continua e non negativa.*

*Dimostrazione.* Indichiamo con  $\tau_1$  e  $\tau_2$  le due facce cui l'edge  $e_h$  appartiene. Per come è stato definito il gradiente, possiamo riscrivere la funzione come

$$\phi(\|\mathcal{N}(V; \tau_1) - \mathcal{N}(V; \tau_2)\|_2 \|v_i - v_j\|_2; a) \quad (3.12)$$

Questa funzione è continua in  $V$  perché lo sono  $\mathcal{N}(V; \tau_1)$ ,  $\mathcal{N}(V; \tau_2)$ , la norma  $\ell_2$  e  $\phi(t; a)$ . È non negativa e propria poiché, per definizione,  $\phi(t; a) \in [0, 1]$  per ogni  $t \in \mathbb{R}$   $\square$

**Corollario 3.3.3.** *L'energia  $\mathcal{J}(V; \lambda, a)$  è propria, continua, limitata inferiormente da 0, coerciva, dunque ammette un minimo locale.*

*Dimostrazione.* Per quanto visto nelle proposizioni (3.3.1) e (3.3.2), l'energia  $\mathcal{J}(V; \lambda, a)$  è continua, propria e limitata inferiormente da 0 poiché somma di  $1 + n_E$  funzioni con tali proprietà. Inoltre è coerciva poiché somma di una funzione coerciva e di  $n_E$  funzioni continue e limitate.  $\square$

L'esistenza di minimi locali è quindi dimostrata, ma poiché la funzione  $\mathcal{J}(V; \lambda, a)$  non è convessa non è possibile verificare se tali minimi sono anche globali.

Per risolvere il problema di ottimizzazione, applichiamo il metodo ADMM (Alternating Directions Method of Multipliers). Questo approccio, attraverso l'introduzione di nuove variabili ausiliarie, permette di ottenere un insieme equivalente di problemi di minimo, più semplici da risolvere.

Una prima variabile ausiliaria è il campo normale  $N \in \mathbb{R}^{n_T \times 3}$ . La seconda è il suo gradiente  $t = D N$ . L'equivalente problema di minimo vincolato è

$$\{V^*, N^*, t^*\} \in \underset{V, N, t}{\operatorname{argmin}} \{ \mathcal{J}(V, N, t; \lambda, a) \quad \text{t.c.} \quad t = D N, \quad N = \mathcal{N}(V) \} \quad (3.13)$$

$$\mathcal{J}(V, N, t; \lambda, a) = \frac{\lambda}{2} \|V - V^0\|_2^2 + \sum_{j=1}^{n_E} \phi(\|t_j\|_2; a) \quad (3.14)$$

Il problema di minimo iniziale (3.1) nella sola variabile  $V$  è così diventato un problema di minimo vincolato in tre variabili  $V$ ,  $N$  e  $t$ .

Per risolverlo, si consideri il funzionale Lagrangiano aumentato

$$\begin{aligned} \mathcal{L}(V, N, t, \rho_1, \rho_2; \lambda, \beta_1, \beta_2, a) = & \\ & \frac{\lambda}{2} \|V - V^0\|_2^2 + \sum_{j=1}^{n_E} \left[ \phi(\|t_j\|_2; a) - \langle \rho_{1j}, t_j - (D N)_j \rangle + \frac{\beta_1}{2} \|t_j - (D N)_j\|_2^2 \right] + \\ & + \sum_{m=1}^{n_T} \left[ -\langle \rho_{2m}, N_m - \mathcal{N}(V; m) \rangle + \frac{\beta_2}{2} \|N_m - \mathcal{N}(V; m)\|_2^2 \right] \end{aligned} \quad (3.15)$$

dove  $\rho_1 \in \mathbb{R}^{n_E \times 3}$ ,  $\rho_2 \in \mathbb{R}^{n_T \times 3}$  sono i moltiplicatori di Lagrange corrispondenti rispettivamente a  $t$  e  $N$ , mentre  $\beta_1 > 0$  e  $\beta_2 > 0$  sono parametri assegnati ai termini di vincolo.

La soluzione è ottenuta iterativamente, inizializzando  $V^{(0)}$ ,  $N^{(0)}$ ,  $t^{(0)}$ ,  $\rho_1^{(0)}$  e  $\rho_2^{(0)}$  in questo modo:

- $V^{(0)} = V^0$  vertici della mesh in input;
- $N^{(0)}$  normali della mesh in input;
- $t^{(0)} = D N^{(0)}$  gradiente della normale;
- $\rho_1^{(0)} = 0 * t$ ;
- $\rho_2^{(0)} = 0 * N^{(0)}$ .

I successivi aggiornamenti seguono lo schema:

$$t^{(k+1)} \leftarrow \arg \min_{t \in \mathbb{R}^{n_T \times 3}} \mathcal{L}(V^{(k)}, N^{(k)}, t, \rho_1^{(k)}, \rho_2^{(k)}; \lambda, \beta_1, \beta_2, a) \quad (3.16)$$

$$N^{(k+1)} \leftarrow \arg \min_{N \in \mathbb{R}^{n_T \times 3} \quad \|N_m\|_2=1} \mathcal{L}(V^{(k)}, N, t^{(k+1)}, \rho_1^{(k)}, \rho_2^{(k)}; \lambda, \beta_1, \beta_2, a) \quad (3.17)$$

$$V^{(k+1)} \leftarrow \arg \min_{V \in \mathbb{R}^{n_V \times 3}} \mathcal{L}(V, N^{(k+1)}, t^{(k+1)}, \rho_1^{(k)}, \rho_2^{(k)}; \lambda, \beta_1, \beta_2, a) \quad (3.18)$$

$$\rho_1^{(k+1)} \leftarrow \rho_1^{(k)} - \beta_1 (t^{(k+1)} - D N^{(k+1)}) \quad (3.19)$$

$$\rho_2^{(k+1)} \leftarrow \rho_2^{(k)} - \beta_2 (N^{(k+1)} - \mathcal{N}(V^{(k+1)})), \quad (3.20)$$

Mentre gli aggiornamenti di  $\rho_1$  e  $\rho_2$  sono ottenuti in maniera diretta, quelli di  $t$ ,  $N$  e  $V$  consistono nella risoluzione di tre problemi di minimo, che saranno studiati in dettaglio nelle prossime sezioni.

### 3.3.1 La consistenza delle normali

La fase di ricostruzione dei vertici ha come obiettivo forzare l'uguaglianza tra il vettore normale al triangolo  $\tau = (v_i, v_j, v_k)$ , cioè  $\mathcal{N}(V; \tau) = \frac{(v_j - v_i) \times (v_k - v_i)}{\|(v_j - v_i) \times (v_k - v_i)\|_2}$ , e la normale aggiornata  $N_\tau$ .

Molti modelli, come [15] e [14], ottengono questo risultato penalizzando la non ortogonalità tra la normale  $N_\tau$  e gli edge  $(v_i, v_j)$ , attraverso un'espressione della forma

$$\min_V \sum_{\tau} \sum_{(v_i, v_j) \in \tau} s_\tau \langle N_\tau, v_i - v_j \rangle^2 \quad (3.21)$$

con  $s_\tau$  area del triangolo  $\tau$ . In questo modo il modello riesce ad identificare la direzione delle normali, ma non il loro verso. Infatti  $N_\tau$  e  $-N_\tau$  non sono distinguibili, poiché il prodotto scalare è elevato al quadrato. Ciò produce errori nella ricostruzione.

Un'idea alternativa è stata sviluppata in [8]. Viene massimizzato l'allineamento tra  $\frac{(v_j - v_i) \times (v_k - v_i)}{\|(v_j - v_i) \times (v_k - v_i)\|_2}$  e la normale  $N_\tau$ , risolvendo il problema di minimo

$$\min_V \left[ \frac{\lambda}{2} \|V - V^0\|_2^2 - \sum_{\tau=(v_i, v_j, v_k)} s_\tau \left\langle N_\tau, \frac{(v_j - v_i) \times (v_k - v_i)}{\|(v_j - v_i) \times (v_k - v_i)\|_2} \right\rangle \right] \quad (3.22)$$

In questo caso, il prodotto scalare tiene conto della differenza di orientamento di  $N_\tau$ , garantendo la consistenza delle normali.

Nel nostro modello, la ricostruzione dei vertici è legata al sottoproblema (3.18), cioè alla minimizzazione della funzione

$$J(V) = \frac{\lambda}{2} \|V - V^0\|_2^2 + \sum_{\substack{m=1 \\ \tau_m=(v_i, v_j, v_k)}}^{n_T} \left[ \left\langle (\rho_2)_m, \frac{(v_j - v_i) \times (v_k - v_i)}{\|(v_j - v_i) \times (v_k - v_i)\|_2} \right\rangle + \frac{\beta_2}{2} \|N_m\|^2 + \frac{\beta_2}{2} - \beta_2 \left\langle N_m, \frac{(v_j - v_i) \times (v_k - v_i)}{\|(v_j - v_i) \times (v_k - v_i)\|_2} \right\rangle \right] \quad (3.23)$$

ottenuta ignorando i termini indipendenti da  $V$ . Tale funzione differisce da (3.22) per la presenza dei termini costanti  $\frac{\beta_2}{2} \|N_m\|^2$  e  $\frac{\beta_2}{2}$  (che non influenzano il problema di minimo) e soprattutto per il prodotto scalare

$$\left\langle (\rho_2^{(k)})_\tau, \frac{(v_j - v_i) \times (v_k - v_i)}{\|(v_j - v_i) \times (v_k - v_i)\|_2} \right\rangle.$$

Per capire se il modello preserva le normali, è necessario dare un'interpretazione geometrica del vettore  $(\rho_2^{(k)})_\tau$ . Ricordiamo il suo aggiornamento:

$$\rho_2^{(0)} = 0 * N^{(0)} \quad (3.24)$$

$$\begin{aligned} (\rho_2^{(k)})_\tau &= (\rho_2^{(k-1)})_\tau - \beta_2 \left( N_m^{(k)} - \frac{(v_j - v_i) \times (v_k - v_i)}{\|(v_j - v_i) \times (v_k - v_i)\|_2} \right), \\ \text{per } m &= 1, \dots, n_T, \quad \tau_m = (v_i, v_j, v_k); \quad v_i \in V^{(k)}. \end{aligned} \quad (3.25)$$

**Remark 3.3.4.** Poiché che  $N_\tau^{(k)}$  è il vettore normale al triangolo e  $\frac{(v_j-v_i) \times (v_k-v_i)}{\|(v_j-v_i) \times (v_k-v_i)\|_2}$  è la sua ricostruzione tramite i vertici, è ragionevole supporre che  $(\rho_2^{(k)})_\tau$  sia piccolo in norma. Inoltre, in quanto differenza di due vettori di norma unitaria, la variabile duale  $(\rho_2^{(k)})_\tau$  tende ad essere ortogonale alla corrispondente variabile primale  $N_\tau$ . Dunque il prodotto scalare in (3.23) assume valori vicini a 0, mentre  $(\rho_2^{(k)})_\tau$  viene forzato a giacere sul piano definito dal triangolo  $\tau$ . La proprietà di consistenza delle normali continua a valere.

È possibile verificare l'accuratezza di queste osservazioni su  $\rho_2$  numericamente, dopo aver implementato l'algoritmo.

### 3.3.2 Il problema per (3.16)

Trascurando i termini indipendenti da  $t$ , la funzione (3.15) diventa

$$A(t) = \sum_{j=1}^{n_E} \phi(\|t_j\|_2; a) - \langle \rho_1^{(k)}, t \rangle + \frac{\beta_1}{2} \|t - D N^{(k)}\|_2^2, \quad (3.26)$$

Osservando che i termini  $t$  e  $D N^{(k)}$  sono matrici di dimensione  $n_E \times 3$  e che la norma dell'ultimo termine è di tipo Frobenius, il problema si può risolvere separatamente su ogni edge. Otteniamo dunque  $n_E$  problemi di minimizzazione tridimensionali del tipo

$$t_j^{k+1} = \operatorname{argmin}_{t_j \in \mathbb{R}^3} \underbrace{\left( \phi(\|t_j\|_2; a) + \frac{\beta_1}{2} \|t_j - r_j\|_2^2 \right)}_{\theta(t_j)} \quad (3.27)$$

dove i termini costanti sono stati raggruppati in  $r_j$ , ponendo

$$r_j = (D N^{(k)})_j - \frac{1}{\beta_1} \rho_{1j}^{(k)}$$

La seguente proposizione studia le condizioni affinché la funzione  $\theta(t_j)$  sia convessa e abbia dunque un unico minimo.

**Proposizione 3.3.5.** *Sia  $\phi(\cdot; a)$  la funzione definita in (3.9),  $r_j \in \mathbb{R}^m$  un vettore costante,  $a, \beta_1 \in \mathbb{R}^+$  parametri positivi. Allora la funzione  $\theta(t_j)$  è strettamente convessa se e solo se  $\beta_1 > a$ .*

La proprietà si ricava semplicemente dalla Proposition 2 in [13]. Supponendo dunque  $\beta_1 > a$ , possiamo ridurre ancora la dimensione del problema.

**Proposizione 3.3.6.** *Sia  $t_j$  un generico vettore in  $\mathbb{R}^3$  e sia  $t_\pi = \|t_j\|_2 \frac{r_j}{\|r_j\|_2}$  il vettore di norma  $\|t_j\|_2$  nella direzione  $r_j$ . Allora  $\theta(t_j) \geq \theta(t_\pi)$ .*

*Dimostrazione.*  $\phi(\|t_j\|_2; a) = \phi(\|t_\pi\|_2; a)$  poiché i due vettori hanno la stessa norma. Dunque

$$\begin{aligned}\theta(t_j) - \theta(t_\pi) &= \frac{\beta_1}{2} (\|t_j - r_j\|_2^2 - \|t_\pi - r_j\|_2^2) = \\ &= \frac{\beta_1}{2} (\|t_j\|_2^2 + \|r_j\|_2^2 - 2\langle t_j, r_j \rangle - \|t_\pi\|_2^2 - \|r_j\|_2^2 + 2\langle t_\pi, r_j \rangle) = \\ &= \beta_1 \langle t_\pi - t_j, r_j \rangle = \\ &= \beta_1 \|t_\pi - t_j\|_2 \|r_j\|_2 \cos(\widehat{Ot_\pi t_j}) > 0\end{aligned}$$

poiché l'angolo  $\widehat{Ot_\pi t}$  è sempre acuto.  $\square$

Dunque ci si può ridurre a studiare la funzione  $\theta$  nella direzione  $r_j$ , cioè per punti  $t_j = c r_j$ , risolvendo il problema di minimo nella sola variabile  $c > 0$ .

**Proposizione 3.3.7.** *Il problema  $\operatorname{argmin}_{t_j \in \mathbb{R}^3} \theta(t_j)$  ha l'unica soluzione  $t^* = c^* r_j$ , con*

$$c^* = \min \left\{ \max \left\{ \frac{\beta_1}{\beta_1 - a} - \frac{\sqrt{2a}}{(\beta_1 - a)\|r_j\|_2}, 0 \right\}, 1 \right\} \quad (3.28)$$

*Dimostrazione.* Per la proprietà (3.3.6), basta studiare il problema  $\operatorname{argmin}_{c>0} f(c)$ , con

$$f(c) := \phi(\|r_j\|_2 c; a) + \frac{\beta_1}{2} \|r_j\|_2^2 (c^2 - 2c) \quad (3.29)$$

Se  $\beta_1 > a$ , questa funzione è differenziabile e strettamente convessa, poiché è la restrizione della funzione  $\theta$  in (3.3.5). Inoltre è crescente per  $c > 1$  poiché lo sono entrambi i suoi addendi. Dunque un punto  $c \in (0, 1)$  è minimo globale di  $f$  se e solo se

$$0 = f'(c) = \|r_j\|_2 [\phi'(\|r_j\|_2 c; a) + \beta_1 \|r_j\|_2 (c - 1)] \quad (3.30)$$

$f'(c)$  è continua e strettamente crescente in  $[0, 1]$  e agli estremi vale:

$$\begin{aligned}f'(0^+) &= \|r_j\|_2 \left[ \sqrt{2a} - \beta_1 \|r_j\|_2 \right] \begin{cases} \geq 0 & \text{per } \|r_j\|_2 \leq \sqrt{2a}/\beta_1 \\ < 0 & \text{altrimenti} \end{cases} \\ f'(1) &= \|r_j\|_2 \phi'(\|r_j\|_2; a) \begin{cases} > 0 & \text{per } \|r_j\|_2 < \sqrt{2/a} \\ = 0 & \text{altrimenti} \end{cases}\end{aligned}$$

Poiché  $\beta_1 > a$ , si ha  $\sqrt{2a}/\beta_1 < \sqrt{2/a}$ , dunque studiamo tre casi:

$$\begin{aligned}a) \quad & \|r_j\|_2 \leq \sqrt{2a}/\beta_1 \implies f'(0^+) \geq 0, f'(1) > 0 \\ b) \quad & \sqrt{2a}/\beta_1 < \|r_j\|_2 \leq \sqrt{2/a} \implies f'(0^+) < 0, f'(1) > 0 \\ c) \quad & \sqrt{2/a} \leq \|r_j\|_2 \implies f'(0^+) < 0, f'(1) = 0\end{aligned} \quad (3.31)$$

Nel caso a),  $c^* = 0$ ; nel caso c),  $c^* = 1$ . Nel caso b), invece,  $c^* \in (0, 1)$  e dall'equazione (3.30) si ricava il suo valore:

$$c^* = \frac{\beta_1}{\beta_1 - a} - \frac{\sqrt{2a}}{(\beta_1 - a)\|r_j\|_2}$$

$\square$

### 3.3.3 Il problema per (3.17)

Come nel caso precedente, omettiamo dalla formula (3.15) i termini indipendenti da  $N$ , ottenendo così la funzione

$$B(N) = \frac{\beta_1}{2} \|t^{(k+1)} - D N\|_2^2 + \langle \rho_1^{(k)}, D N \rangle - \langle \rho_2^{(k)}, N \rangle + \frac{\beta_2}{2} \|N - \mathcal{N}(V^{(k)})\|_2^2. \quad (3.32)$$

che è differenziabile. Dunque calcoliamo la prima condizione di ottimalità

$$-\beta_1 D^T (t^{(k+1)} - D N) + D^T \rho_1^{(k)} - \rho_2^{(k)} + \beta_2 (N - \mathcal{N}(V^{(k)})) = 0, \quad (3.33)$$

mediante la quale si ricava  $N^{(k+1)}$  dalla soluzione del sistema lineare

$$\underbrace{(\beta_1 D^T D + \beta_2 I)}_{LHS} N = \underbrace{\beta_2 \mathcal{N}(V^{(k)}) + \rho_2^{(k)} + \beta_1 D^T (t^{(k+1)} - \frac{1}{\beta_1} \rho_1^{(k)})}_{RHS}. \quad (3.34)$$

Le dimensioni delle matrici  $LHS$  e  $RHS$  sono rispettivamente  $n_T \times n_T$  e  $n_T \times 3$ . Inoltre, poiché  $\beta_1, \beta_2 > 0$ , la matrice  $LHS$  è simmetrica definita positiva. Dunque esiste un'unica matrice  $N^{(k+1)}$  che soddisfa la prima condizione di ottimalità ed è il punto di minimo della funzione  $B(N)$ . La risoluzione del sistema può essere svolta attraverso un'unica decomposizione di Cholesky. Ad ogni iterazione di ADMM la matrice  $D$  viene aggiornata ricalcolando la matrice  $L$  delle lunghezze degli edge.

Infine, ogni riga di  $N^{(k+1)}$  viene normalizzata, in modo da rappresentare un vettore di norma unitaria.

### 3.3.4 Il problema per (3.18)

I termini di (3.15) dipendenti da  $V$  compongono l'espressione

$$J(V) = \frac{\lambda}{2} \|V - V^0\|_2^2 + \sum_{\substack{m=1 \\ \tau_m=(v_i, v_j, v_k)}}^{n_T} \left[ \left\langle (\rho_2^{(k)})_m, \frac{(v_j - v_i) \times (v_k - v_i)}{\|(v_j - v_i) \times (v_k - v_i)\|_2} \right\rangle + \frac{\beta_2}{2} \left\| N_m^{(k+1)} - \frac{(v_j - v_i) \times (v_k - v_i)}{\|(v_j - v_i) \times (v_k - v_i)\|_2} \right\|_2^2 \right] \quad (3.35)$$

Sviluppiamo l'ultimo termine. Osservando che  $\frac{(v_j - v_i) \times (v_k - v_i)}{\|(v_j - v_i) \times (v_k - v_i)\|_2}$  ha norma pari a 1 per definizione, mentre  $\|N_\tau^{(k+1)}\|_2 = 1$  per costruzione, otteniamo

$$J(V) = \frac{\lambda}{2} \|V - V^0\|_2^2 + \sum_{\tau=(v_i, v_j, v_k)} \left[ \left\langle (\rho_2^{(k)})_\tau, \frac{(v_j - v_i) \times (v_k - v_i)}{\|(v_j - v_i) \times (v_k - v_i)\|_2} \right\rangle + \beta_2 - \beta_2 \left\langle N_\tau^{(k+1)}, \frac{(v_j - v_i) \times (v_k - v_i)}{\|(v_j - v_i) \times (v_k - v_i)\|_2} \right\rangle \right] \quad (3.36)$$

Analizziamo questa funzione.

Si considerino i termini della sommatoria relativi al triangolo  $\tau$ . L'espressione

$$J_\tau(V) := \left\langle (\rho_2^{(k)})_\tau, \mathcal{N}(V; \tau) \right\rangle + \beta_2 - \beta_2 \langle N_\tau^{(k+1)}, \mathcal{N}(V; \tau) \rangle \quad (3.37)$$

è continua, differenziabile e limitata. Infatti, poiché  $\mathcal{N}(V; \tau)$  è un vettore di norma unitaria, si ha

$$|J_\tau(V)| \leq \|(\rho_2^{(k)})_\tau\|_2 + \beta_2 + \beta_2 \|N_\tau^{(k+1)}\|_2 \quad (3.38)$$

Il termine di fedeltà ai dati iniziali è una funzione propria, continua, differenziabile, non negativa e coerciva, come visto nella Prop. 3.3.1.

Dunque  $J(V)$  è propria, continua, differenziabile, inferiormente limitata e coerciva. Ha quindi un minimo. La sua unicità sarà studiata al termine della sezione, attraverso il calcolo della matrice hessiana. Adesso invece ci concentriamo sul calcolo del gradiente di  $J(V)$  rispetto a  $v_i$ , per ogni vertice  $v_i$  della mesh. Osserviamo due proprietà preliminari.

**Lemma 3.3.8.** *Per ogni  $w \in \mathbb{R}^3$  costante, vale*

$$\nabla_{v_i}(\langle w, (v_j - v_i) \times (v_k - v_i) \rangle) = w \times (v_k - v_j)$$

*Dimostrazione.* Sfruttando la linearità dei prodotti vettoriale e scalare ed eliminando i termini indipendenti da  $v_i$

$$\begin{aligned} \nabla_{v_i}(\langle w, (v_j - v_i) \times (v_k - v_i) \rangle) &= \nabla_{v_i}(\langle w, -v_j \times v_i - v_i \times v_k \rangle) = \\ &= \nabla_{v_i}(\langle w, v_i \times (v_j - v_k) \rangle) = \\ &= \nabla_{v_i}(\langle v_i, (v_j - v_k) \times w \rangle) = \\ &= (v_j - v_k) \times w = w \times (v_k - v_j) \end{aligned}$$

□

**Lemma 3.3.9.** *Per ogni  $u, w \in \mathbb{R}^3$  costanti, vale*

$$\nabla_{v_i} \left( \left\langle \frac{w}{\|(v_j - v_i) \times (v_k - v_i)\|_2}, u \right\rangle \right) = -\langle w, u \rangle \frac{(v_j - v_i) \times (v_k - v_i) \times (v_k - v_j)}{\|(v_j - v_i) \times (v_k - v_i)\|_2^3}$$

*Dimostrazione.*

$$\begin{aligned} \nabla_{v_i} \left( \left\langle \frac{w}{\|(v_j - v_i) \times (v_k - v_i)\|_2}, u \right\rangle \right) &= \langle w, u \rangle \nabla_{v_i} \left( \frac{1}{\|(v_j - v_i) \times (v_k - v_i)\|_2} \right) = \\ &= -\frac{\langle w, u \rangle \nabla_{v_i} (\|(v_j - v_i) \times (v_k - v_i)\|_2)}{\|(v_j - v_i) \times (v_k - v_i)\|_2^2} = \\ &= -\frac{\langle w, u \rangle \nabla_{v_i} (\langle (v_j - v_i) \times (v_k - v_i), (v_j - v_i) \times (v_k - v_i) \rangle^{1/2})}{\|(v_j - v_i) \times (v_k - v_i)\|_2^2} = \\ &= -\frac{\langle w, u \rangle}{\|(v_j - v_i) \times (v_k - v_i)\|_2^2} \frac{2(v_j - v_i) \times (v_k - v_i) \times (v_k - v_j)}{2\|(v_j - v_i) \times (v_k - v_i)\|_2} = \\ &= -\langle w, u \rangle \frac{(v_j - v_i) \times (v_k - v_i) \times (v_k - v_j)}{\|(v_j - v_i) \times (v_k - v_i)\|_2^3} \end{aligned}$$

□

Adesso possiamo calcolare il gradiente di  $J(V)$  rispetto al vertice  $v_i$ .

**Proposizione 3.3.10.**

$$\begin{aligned} \nabla_{v_i} J(V) = \lambda(v_i - v_i^0) + \sum_{\substack{\tau \in \mathcal{D}(v_i) \\ \tau = (v_i, v_j, v_k)}} & \left[ \frac{\left( \rho_2^{(k)} - \beta_2 N^{(k+1)} \right)_\tau \times (v_k - v_j)}{\|(v_j - v_i) \times (v_k - v_i)\|_2} - \right. \\ & \left. - \frac{\left\langle \left( \rho_2^{(k)} - \beta_2 N^{(k+1)} \right)_\tau, (v_j - v_i) \times (v_k - v_i) \right\rangle [(v_j - v_i) \times (v_k - v_i) \times (v_k - v_j)]}{\|(v_j - v_i) \times (v_k - v_i)\|_2^3} \right] \end{aligned} \quad (3.39)$$

*Dimostrazione.* Ricordiamo che

$$J(V) = \frac{\lambda}{2} \|V - V^0\|_2^2 + \sum_{\tau} J_{\tau}(V)$$

Il gradiente del primo termine è semplicemente  $\lambda(v_i - v_i^0)$ . I termini  $J_{\tau}(V)$  nella sommatoria dipendono da  $v_i$  se e solo se  $v_i \in \tau$ , cioè se  $\tau$  appartiene al primo disco  $\mathcal{D}(v_i)$ . Dunque

$$\nabla_{v_i} J(V) = \lambda(v_i - v_i^0) + \sum_{\tau \in \mathcal{D}(v_i)} \nabla_{v_i} J_{\tau}(V)$$

Calcoliamo  $\nabla_{v_i} J_{\tau}(V)$ , per  $\tau = (v_i, v_j, v_k)$ . Il termine  $\beta_2$  è costante, indipendente da  $V$ , dunque non dà contributo al gradiente. Vale

$$\begin{aligned} \nabla_{v_i} J_{\tau}(V) &= \nabla_{v_i} \left( \left\langle \left( \rho_2^{(k)} - \beta_2 N^{(k+1)} \right)_\tau, \frac{(v_j - v_i) \times (v_k - v_i)}{\|(v_j - v_i) \times (v_k - v_i)\|_2} \right\rangle \right) \\ &= \nabla_{v_i} \left( \left\langle \frac{\left( \rho_2^{(k)} - \beta_2 N^{(k+1)} \right)_\tau}{\|(v_j - v_i) \times (v_k - v_i)\|_2}, (v_j - v_i) \times (v_k - v_i) \right\rangle \right) \end{aligned}$$

Calcoliamo il gradiente tramite la regola di derivazione di un prodotto scalare. Considerando costante il vettore di sinistra e variabile quello di destra, ricadiamo nel caso del Lemma 3.3.8, con

$$w = \frac{\left( \rho_2^{(k)} - \beta_2 N^{(k+1)} \right)_\tau}{\|(v_j - v_i) \times (v_k - v_i)\|_2}.$$

Viceversa, considerando costante il vettore di destra e variabile quello di sinistra, sfruttiamo il Lemma 3.3.9 ponendo

$$w = \left( \rho_2^{(k)} - \beta_2 N^{(k+1)} \right)_\tau \quad \text{e} \quad u = (v_j - v_i) \times (v_k - v_i)$$

Otteniamo così il gradiente rispetto a  $v_i$  di  $J_\tau(V)$ :

$$\begin{aligned} \nabla_{v_i} J_\tau(V) = & \frac{\left(\rho_2^{(k)} - \beta_2 N^{(k+1)}\right)_\tau}{\|(v_j - v_i) \times (v_k - v_i)\|_2} \times (v_k - v_j) - \\ & - \left\langle \left(\rho_2^{(k)} - \beta_2 N^{(k+1)}\right)_\tau, (v_j - v_i) \times (v_k - v_i) \right\rangle \frac{(v_j - v_i) \times (v_k - v_i) \times (v_k - v_j)}{\|(v_j - v_i) \times (v_k - v_i)\|_2^3} \end{aligned} \quad (3.40)$$

Sommando al variare di  $\tau \in \mathcal{D}(v_i)$  e aggiungendo il termine  $\lambda(v_i - v_i^0)$  otteniamo l'espressione finale di  $\nabla_{v_i} J(V)$ .  $\square$

Ricordando che:

1.

$$\|(v_j - v_i) \times (v_k - v_i)\|_2 = 2s_\tau, \text{ con } s_\tau \text{ area del triangolo } \tau;$$

2.

$$\frac{(v_j - v_i) \times (v_k - v_i)}{\|(v_j - v_i) \times (v_k - v_i)\|_2} = \frac{(v_j - v_i) \times (v_k - v_i)}{2s_\tau} = \mathcal{N}(V; \tau).$$

il gradiente può essere scritto nella forma

$$\begin{aligned} \nabla_{v_i} J(V) = & \lambda(v_i - v_i^0) + \\ & \sum_{\tau \in \mathcal{D}(v_i)} \frac{\left[ \left(\rho_2^{(k)} - \beta_2 N^{(k+1)}\right)_\tau - \left\langle \left(\rho_2^{(k)} - \beta_2 N^{(k+1)}\right)_\tau, \mathcal{N}(V^{(k)}; \tau) \right\rangle \mathcal{N}(V^{(k)}; \tau) \right]}{2s_\tau} \times (v_k - v_j) \end{aligned} \quad (3.41)$$

L'espressione (3.41) sarà utilizzata nell'algoritmo per risolvere il sottoproblema per  $V$ .

È possibile anche ottenere una formulazione più semplice del gradiente, assumendo che i termini non lineari nella forma

$$\frac{1}{\|(v_j - v_i) \times (v_k - v_i)\|_2} = \frac{1}{2s_\tau^{(k)}}$$

siano costanti, indipendenti da  $V$ . In tal caso, l'energia da minimizzare diventa dunque

$$\begin{aligned} J(V) = & \frac{\lambda}{2} \|V - V^0\|_2^2 + \sum_{\substack{m=1 \\ \tau_m=(v_i, v_j, v_k)}}^{n_\tau} \left[ \left\langle \left(\rho_2^{(k)}\right)_m, \frac{(v_j - v_i) \times (v_k - v_i)}{2s_\tau^{(k)}} \right\rangle + \right. \\ & \left. + \frac{\beta_2}{2} \left\| N_m^{(k+1)} - \frac{(v_j - v_i) \times (v_k - v_i)}{2s_\tau^{(k)}} \right\|_2^2 \right] \end{aligned} \quad (3.42)$$

Il calcolo del gradiente risulta così più immediato.

**Proposizione 3.3.11.** Considerando costanti, indipendenti da  $V$  i termini  $s_\tau^{(k)}$ , il gradiente della funzione  $J(V)$  rispetto al vertici  $v_i$  è dato dall'espressione

$$\begin{aligned} \nabla_{v_i} J(V) &= \lambda(v_i - v_i^0) + \\ &+ \sum_{\substack{\tau \in \mathcal{D}(v_i) \\ \tau = (v_i, v_j, v_k)}} \frac{(\rho_2^{(k)})_\tau - \beta_2 N_\tau^{(k+1)} + \beta_2 \mathcal{N}(V; \tau)}{2s_\tau^{(k)}} \times (v_k - v_j) \end{aligned} \quad (3.43)$$

*Dimostrazione.* Il gradiente del primo addendo non cambia, resta  $\lambda(v_i - v_i^0)$ . Per quanto riguarda i termini della sommatoria, osserviamo che dipendono da  $v_i$  solo se  $v_i \in \tau$ , cioè se  $\tau$  appartiene al primo disco  $\mathcal{D}(v_i)$ . Sfruttando la linearità di prodotto scalare e prodotto vettoriale, omettiamo i termini indipendenti da  $v_i$ .

$$\begin{aligned} \nabla_{v_i} J(V) &= \lambda(v_i - v_i^0) + \sum_{\substack{\tau \in \mathcal{D}(v_i) \\ \tau = (v_i, v_j, v_k)}} \left[ \nabla_{v_i} \left( \left\langle (\rho_2^{(k)})_\tau, \frac{(v_j - v_i) \times (v_k - v_i)}{2s_\tau^{(k)}} \right\rangle \right) + \right. \\ &+ \frac{\beta_2}{2} \nabla_{v_i} \left( -2 \left\langle N_\tau^{(k+1)}, \frac{(v_j - v_i) \times (v_k - v_i)}{2s_\tau^{(k)}} \right\rangle \right) + \\ &+ \left. \frac{\beta_2}{2} \nabla_{v_i} \left( \left\langle \frac{(v_j - v_i) \times (v_k - v_i)}{2s_\tau^{(k)}}, \frac{(v_j - v_i) \times (v_k - v_i)}{2s_\tau^{(k)}} \right\rangle \right) \right] = \\ &= \lambda(v_i - v_i^0) + \sum_{\substack{\tau \in \mathcal{D}(v_i) \\ \tau = (v_i, v_j, v_k)}} \left[ \frac{(\rho_2^{(k)})_\tau}{2s_\tau^{(k)}} \times (v_k - v_j) - \frac{\beta_2}{2s_\tau^{(k)}} N_\tau^{(k+1)} \times (v_k - v_j) + \right. \\ &+ \left. \frac{\beta_2}{(2s_\tau^{(k)})^2} ((v_j - v_i) \times (v_k - v_i)) \times (v_k - v_j) \right]. \end{aligned}$$

dove l'ultima uguaglianza è un'applicazione del Lemma 3.3.8. □

Il problema di ottimizzazione per  $V$  (3.18) viene risolto mediante un metodo iterativo di discesa per la minimizzazione di funzioni differenziabili. Inizializzando  $\bar{V}^{(0)} = V^{(k)}$ , i successivi aggiornamenti sono dati da

$$\bar{V}^{(j+1)} = \bar{V}^{(j)} + \alpha_j p_j$$

$p_j$  è la direzione di discesa e può essere scelta nei modi descritti nella sezione 2.1.

Applicando il metodo di discesa gradiente, si ha  $p_j = -\nabla J(\bar{V}^{(j)})$ .

In alternativa, è possibile usare l'algoritmo BFGS, costruendo la matrice  $H_j$  che approssima l'inversa dell'Hessiana, tramite l'espressione in (2.22). Prima di applicarla, è bene notare che nella sezione 2.1.4 il metodo BFGS era stato definito per funzioni scalari su  $\mathbb{R}^n$ . In questo caso invece,  $V$  è una matrice  $n_V \times 3$ , che deve quindi essere trasformata

in un vettore con dimensioni  $3n_V \times 1$ , rappresentato dai 3 blocchi delle coordinate scalari  $x$ ,  $y$  e  $z$  dei vertici. Una volta fatto ciò, si può proseguire con le seguenti operazioni:

$$s_{j-1} = \bar{V}^{(j)} - \bar{V}^{(j-1)} \quad (3.44)$$

$$y_{j-1} = \nabla J(\bar{V}^{(j)}) - \nabla J(\bar{V}^{(j-1)}) \quad (3.45)$$

$$\rho_{j-1} = 1/(y_{j-1}^T s_{j-1}) \quad (3.46)$$

$$H_j = (I - \rho_{j-1} s_{j-1} y_{j-1}^T) H_{j-1} (I - \rho_{j-1} y_{j-1} s_{j-1}^T) + \rho_{j-1} s_{j-1} s_{j-1}^T \quad (3.47)$$

La direzione  $p_j$  è dunque data dal prodotto matrice per vettore

$$p_j = -H_j \nabla J(\bar{V}^{(j)}) \quad (3.48)$$

Infine  $\bar{V}^{(j+1)} = \bar{V}^{(j)} + \alpha_j p_j$ .

Al termine del processo iterativo, occorre ritrasformare il vettore  $V$  in una matrice  $n_V \times 3$ , con un procedimento inverso a quello usato in precedenza.

Lo step  $\alpha_j$  può essere considerato come un parametro del problema, costante in ogni iterazione. Un'alternativa più efficace è aggiornarlo ad ogni passo tramite l'algoritmo di Back-Tracking Line Search descritto in 2.1.1. Nel nostro caso, abbiamo fissato i parametri iniziali in questo modo:

- Lo step massimo  $\bar{\alpha} = 1$ ;
- Il fattore di riduzione  $\rho = 1/2$ ;
- La costante relativa all'Armijo Condition  $c_1 = 10^{-1}$ .

I due algoritmi descritti giungono ad un punto di minimo per  $J(V)$ . Resta ancora ignoto se tale punto di minimo è unico.

L'idea più semplice per studiare tale unicità è osservare se  $J(V)$  è o non è convessa. Il termine di fedeltà ai dati iniziali è quadratico, fortemente convesso. Consideriamo quindi solo un generico termine della sommatoria, relativo al triangolo  $\tau = (v_i, v_j, v_k)$ :

$$F(V) = \left\langle (\rho_2^{(k)} - \beta_2 N^{(k+1)})_\tau, \frac{(v_j - v_i) \times (v_k - v_i)}{\|(v_j - v_i) \times (v_k - v_i)\|_2} \right\rangle \quad (3.49)$$

Per semplicità di notazione, introduciamo

$$\vec{n} = (v_j - v_i) \times (v_k - v_i) \quad (3.50)$$

$$w = (\rho_2^{(k)} - \beta_2 N^{(k+1)})_\tau \quad (3.51)$$

L'idea più immediata per studiare la convessità di  $F(V)$  è calcolarne la matrice hessiana delle derivate seconde rispetto al vertice  $v_i$ . Il gradiente, già calcolato in (3.40), con la notazione appena definita diventa

$$\nabla_{v_i} F(V) = \frac{(w - \langle w, \mathcal{N}(V; \tau) \rangle \mathcal{N}(V; \tau)) \times (v_k - v_j)}{\|\vec{n}\|_2} \quad (3.52)$$

Tale gradiente è un vettore in  $\mathbb{R}^3$ . Per il calcolo dell'hessiana, calcoliamo separatamente il gradiente rispetto a  $v_i$  delle sue componenti.

Cominciando dalla prima, grazie alle proprietà del triplo prodotto possiamo riscriverla in una forma più semplice.

$$\begin{aligned}
(\nabla_{v_i} F(V))_x &= \langle \nabla_{v_i} F(V), (1, 0, 0) \rangle = \\
&= \left\langle \frac{w - \langle w, \mathcal{N}(V; \tau) \rangle \mathcal{N}(V; \tau)}{\|\vec{n}\|_2} \times (v_k - v_j), (1, 0, 0) \right\rangle = \\
&= \left\langle \frac{w - \langle w, \mathcal{N}(V; \tau) \rangle \mathcal{N}(V; \tau)}{\|\vec{n}\|_2}, (v_k - v_j) \times (1, 0, 0) \right\rangle = \\
&= \left\langle \frac{w - \langle w, \mathcal{N}(V; \tau) \rangle \mathcal{N}(V; \tau)}{\|\vec{n}\|_2}, a_x \right\rangle \tag{3.53}
\end{aligned}$$

Quest'ultima espressione è utile perché permette di calcolare il gradiente più facilmente e soprattutto per una sola componente. Infatti, per le componenti  $y$  e  $z$  basterà modificare il vettore  $a_x$  nei corrispondenti  $a_y$  e  $a_z$ .

**Proposizione 3.3.12.**

$$\begin{aligned}
\nabla_{v_i} ((\nabla_{v_i} F(V))_x) &= \left[ 3 \langle w, \mathcal{N}(V; \tau) \rangle \langle a_x, \mathcal{N}(V; \tau) \rangle \mathcal{N}(V; \tau) - \langle w, a_x \rangle \mathcal{N}(V; \tau) - \right. \\
&\quad \left. - \langle w, \mathcal{N}(V; \tau) \rangle a_x - \langle a_x, \mathcal{N}(V; \tau) \rangle w \right] \times \frac{(v_k - v_j)}{\|\vec{n}\|_2^2}
\end{aligned}$$

*Dimostrazione.* La dipendenza da  $v_i$  nell'espressione (3.53) è nascosta nei termini:

1.  $\|\vec{n}\|_2$  al denominatore,
2. il primo  $\mathcal{N}(V; \tau)$ , nel prodotto scalare con  $w$ ,
3. il secondo  $\mathcal{N}(V; \tau)$ .

Per sfruttare le regole di derivazione consideriamo, a turno, due di questi termini come costanti e l'altro variabile. Infine andremo a sommare le tre espressioni trovate per ottenere il gradiente finale.

1. Consideriamo  $\|\vec{n}\|_2$  variabile. Riscriviamo  $(\nabla_{v_i} F(V))_x$  come

$$\langle w - \langle w, \mathcal{N}(V; \tau) \rangle \mathcal{N}(V; \tau), a_x \rangle \frac{1}{\|\vec{n}\|_2}$$

Solo il termine  $1/\|\vec{n}\|_2$  è supposto variabile, dunque la derivata rispetto a  $v_i$ , per il Lemma 3.3.9, è

$$- \langle w - \langle w, \mathcal{N}(V; \tau) \rangle \mathcal{N}(V; \tau), a_x \rangle \frac{\mathcal{N}(V; \tau) \times (v_k - v_j)}{\|\vec{n}\|_2^2} \tag{3.54}$$

2. Consideriamo il primo  $\mathcal{N}(V; \tau)$  variabile. Riscriviamo  $(\nabla_{v_i} F(V))_x$  come

$$\left\langle \frac{w}{\|\vec{n}\|_2}, a_x \right\rangle - \left\langle \frac{\mathcal{N}(V; \tau)}{\|\vec{n}\|_2}, a_x \right\rangle \langle w, \mathcal{N}(V; \tau) \rangle$$

L'unico termine considerato variabile è  $\langle w, \mathcal{N}(V; \tau) \rangle$ . Tramite gli stessi calcoli usati in (3.3.10), la derivata rispetto a  $v_i$  è quindi

$$- \left\langle \frac{\mathcal{N}(V; \tau)}{\|\vec{n}\|_2}, a_x \right\rangle \frac{(w - \langle w, \mathcal{N}(V; \tau) \rangle \mathcal{N}(V; \tau)) \times v_k - v_j}{\|\vec{n}\|_2} \quad (3.55)$$

3. Consideriamo il secondo  $\mathcal{N}(V; \tau)$  variabile. Riscriviamo  $(\nabla_{v_i} F(V))_x$  come

$$\left\langle \frac{w}{\|\vec{n}\|_2}, a_x \right\rangle - \frac{\langle w, \mathcal{N}(V; \tau) \rangle}{\|\vec{n}\|_2} \langle \mathcal{N}(V; \tau), a_x \rangle$$

In questo caso il termine variabile è  $\langle \mathcal{N}(V; \tau), a_x \rangle$ . La derivata è

$$- \left\langle \frac{\mathcal{N}(V; \tau)}{\|\vec{n}\|_2}, w \right\rangle \frac{(a_x - \langle a_x, \mathcal{N}(V; \tau) \rangle \mathcal{N}(V; \tau)) \times v_k - v_j}{\|\vec{n}\|_2} \quad (3.56)$$

Sommando le espressioni (3.54), (3.55) e (3.56), otteniamo infine la tesi.  $\square$

Lo stesso identico procedimento può essere svolto per calcolare  $\nabla_{v_i} \left( (\nabla_{v_i} F(V))_y \right)$  e  $\nabla_{v_i} \left( (\nabla_{v_i} F(V))_z \right)$ , sostituendo al termine  $a_x$  rispettivamente i termini

$$a_y = (v_k - v_j) \times (0, 1, 0) \quad a_z = (v_k - v_j) \times (0, 0, 1)$$

Abbiamo così ottenuto la matrice hessiana della funzione  $F(V)$ :

$$H(V) = \begin{pmatrix} \nabla_{v_i} \left( (\nabla_{v_i} F(V))_x \right) \\ \nabla_{v_i} \left( (\nabla_{v_i} F(V))_y \right) \\ \nabla_{v_i} \left( (\nabla_{v_i} F(V))_z \right) \end{pmatrix} \quad (3.57)$$

**Proposizione 3.3.13.** *Se la matrice  $H(V)$  è semidefinita positiva, allora la funzione  $J(V)$  è convessa e ha un unico minimo.*

A causa della complessità di tale matrice, risulta particolarmente difficile studiarne la positività dal punto di vista teorico, dunque l'unicità del minimo per la funzione  $J(V)$  non è dimostrata. Tuttavia può essere studiata numericamente, durante l'esecuzione dell'algoritmo.

Allo stesso modo, è possibile osservare l'evoluzione lungo le iterazioni sia della funzione  $J(V)$  sia dell'energia complessiva  $\mathcal{J}(V; \lambda, a)$  per verificare la convergenza rispettivamente del sottoproblema per  $V$  e dell'ADMM.

## 3.4 L'algoritmo

---

**Algorithm 2** ADMM per la risoluzione di (3.1)

---

**input:** Vertici  $V^0 \in \mathbb{R}^3$ , edge  $E$ , facce  $T$ ;  
**output:** Vertici  $V^* \in \mathbb{R}^3$   
**parametri:** MODELLO:  $\cdot$  parametro di concavità  $a \geq 0$  per  $\phi(\cdot; a): \mathbb{R}_+ \rightarrow \mathbb{R}_+$   
 $\cdot$  parametro di fedeltà ai dati iniziali  $\lambda > 0$   
ADMM:  $\cdot \beta_1, \beta_2 > 0$

Calcolo dell'operatore gradiente  $D \in \mathbb{R}^{n_E \times n_T}$

Inizializzazione:

$V^{(0)} = V^0$ ,  $N^{(0)} = \mathcal{N}(V^0)$ ,  $t^{(0)} = D N^{(0)}$ ,  $\rho_1^{(0)} = 0 \cdot t$ ,  $\rho_2^{(0)} = 0 \cdot N^0$ ,  $k = 0$ ;

**Finché**  $k < IterMax_{ADMM}$  e  $\|V^{(k+1)} - V^{(k)}\|_2 / \|V^{(k)}\|_2 > tol_{ADMM}$  :

$\cdot k = k + 1$ ;

**% sottoproblema per t:**

$\cdot$  calcola  $t_j^{(k+1)}$  da (3.3.7)

**% sottoproblema per N:**

$\cdot$  calcola le matrici *LHS* e *RHS* del sistema lineare (3.34)

$\cdot$  calcola  $N^{(k+1)}$  risolvendo il sistema lineare (3.34)

$\cdot$  normalizza  $N^{(k+1)}$

**% sottoproblema per V:** tramite discesa del gradiente oppure BFGS

$\cdot$  Inizializzazione  $\bar{V}^{(0)} = V^{(k)}$ ,  $j = 0$

**Finché**  $j < IterMax_{GD}$  e  $\|\bar{V}^{(j+1)} - \bar{V}^{(j)}\|_2 / \|\bar{V}^{(j)}\|_2 > tol_{GD}$  :

$\cdot j = j + 1$ ;

$\cdot$  Calcolo del gradiente (3.41)

$\cdot$  Calcolo della direzione di discesa  $p_j$  (discesa del gradiente o tramite BFGS)

$\cdot$  Calcolo dello step  $\alpha_j$  (costante oppure tramite Back-Tracking Line Search)

$\cdot$  Aggiornamento del vertice:  $\bar{V}_i^{(j+1)} = \bar{V}_i^{(j)} + \alpha_j p_j$

$\cdot$  Aggiorna le lunghezze degli edge  $L$  e il gradiente  $D$

**Fine**

$\cdot V^{(k+1)} = \bar{V}^{(j+1)}$

**% aggiorna le variabili duali**  $\rho_1$  e  $\rho_2$ : (3.19) (3.20)

**% aggiorna le normali**  $\mathcal{N}$  mediante (3.3)

**Fine**

$V^* = V^{(k+1)}$ ,  $N^* = \mathcal{N}^{(k+1)}$

---



# Capitolo 4

## Validazione del modello

Mesh	$n_V$	$n_E$	$n_T$	Regolarità	Topologia
Cube-hole	10232	30720	20480	lineare a tratti	chiusa
Twelve	4610	13824	9216	lineare a tratti	chiusa
Trim-star	5192	15576	10384	mista	chiusa
Fandisk	6475	19419	12946	mista	chiusa
Block	8771	26352	17550	mista	chiusa
Hand38k	38219	114657	76438	liscia	chiusa
Bunny	34834	104288	69451	liscia	chiusa
Sharp-Sphere	10443	31323	20882	liscia	chiusa
Teddy	9548	28638	19092	liscia	chiusa
Foot	10010	29983	19974	liscia	aperta
Julius	36201	108112	71912	liscia	aperta
Angel	24566	72690	48090	reale	aperta
Rabbit	37394	111073	73679	reale	aperta

Tabella 4.1: Mesh di prova

L'algoritmo è stato validato sulle mesh descritte in Tabella 4.1, caratterizzate da diverse proprietà, sia a livello di topologia e regolarità della superficie, sia rispetto alla raffinatezza delle triangolazioni con cui sono definite.

In particolare, identifichiamo come aperte o chiuse le mesh rispettivamente con o senza bordo. Per le superfici aperte, la definizione di  $(\nabla\mathcal{N})_i$  in (3.5) non è ben posta se  $e_i$  è uno degli edge al bordo, perché  $e_i$  appartiene ad una sola faccia. Risolviamo questo problema imponendo le condizioni di Neumann, cioè definendo  $(\nabla\mathcal{N})_i = 0$ .

Per quanto riguarda la regolarità, identifichiamo tre categorie: lineare a tratti, mista, liscia. **Cube-hole** e **Twelve** sono lineari a tratti, utili per capire come si comporta l'algoritmo su aree piatte e spigoli netti, anche in presenza di buchi. **Trim-star**, **Fandisk** e **Block** sono mesh miste, poiché possiedono anche parti curve e regolari. **Hand38k**, **Bunny** e **Sharp-sphere** sono superfici regolari, senza spigoli evidenti ne buchi.

Infine, le mesh aperte utilizzate sono invece *Foot*, *Julius*, *Angel* e *Rabbit*. Le ultime due sono ottenute da scansioni 3D di oggetti reali.

L'accuratezza del modello è stata verificata perturbando le mesh con diversi livelli di rumore, applicando l'algoritmo 2 e osservando sia qualitativamente sia quantitativamente la distanza tra le mesh originali e la loro ricostruzione. In tutti gli esperimenti, i parametri per i criteri d'arresto sono stati fissati con i seguenti valori:

$$IterMax_{ADMM} = 200, \quad tol_{ADMM} = 10^{-6}; \quad IterMax_{GD} = 50, \quad tol_{GD} = 10^{-7}.$$

## 4.1 Modello matematico del degrado della superficie

La mesh in input è ottenuta modificando la posizione dei vertici  $V^*$  della mesh originale, priva di rumore, in questo modo:

$$v_i^0 = v_i + c_i(\gamma \vec{n}_i + \delta \vec{d}_i) \quad (4.1)$$

$c_i$  è uno scalare ottenuto da una distribuzione normale con media 0 e deviazione standard  $L^{(m)}$ , cioè la lunghezza media degli edge della mesh.  $\vec{n}_i$  è la normale esterna al vertice  $v_i$ , ottenuta normalizzando il vettore

$$\sum_{\substack{\tau \in \mathcal{D}(v_i) \\ \tau = (v_i, v_j, v_k)}} \frac{(v_j - v_i) \times (v_k - v_i)}{\|(v_j - v_i) \times (v_k - v_i)\|_2}. \quad (4.2)$$

$\vec{d}_i$  invece introduce aleatorietà anche nella direzione. In particolare, generato un vettore random  $\vec{r}_i$ ,  $\vec{d}_i$  è la sua proiezione lungo il piano ortogonale a  $\vec{n}_i$ , normalizzata.

Dunque il rumore è una combinazione lineare dei due vettori tra loro ortogonali  $\vec{n}_i$  e  $\vec{d}_i$ , avente come coefficienti i due parametri  $\gamma, \delta \in [0, 1)$ .

La figura 4.1 mostra come varia la mesh in input al crescere di  $\gamma$ , fissato  $\delta = 0$ , dunque con rumore diretto lungo le normali. Nella figura 4.2, invece,  $\gamma$  è costante e  $\delta$  variabile. Si osserva quindi l'influenza del termine tangente nel definire la direzione del rumore.

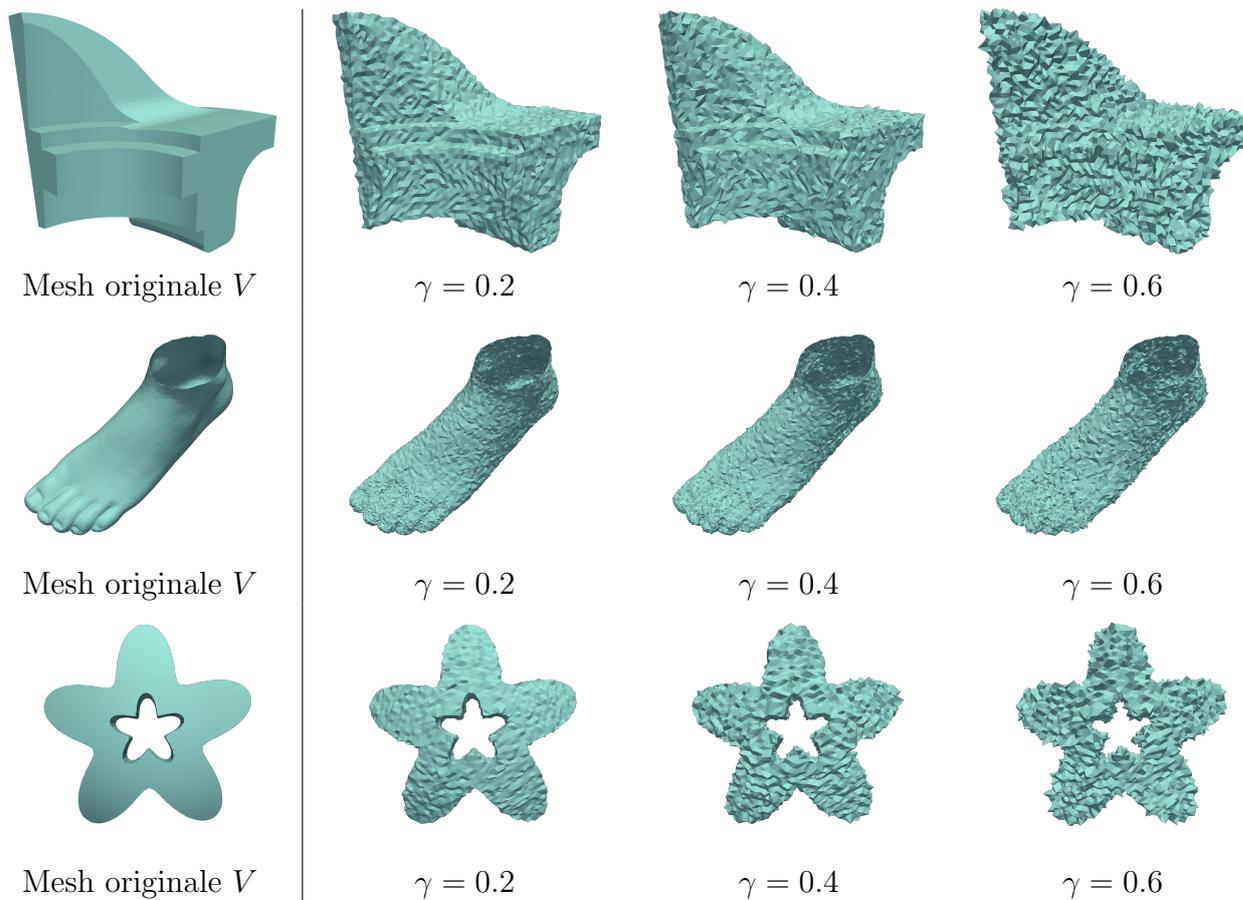


Figura 4.1: L'effetto del rumore diretto lungo le normali, fissato  $\delta = 0$ , al variare di  $\gamma$ .

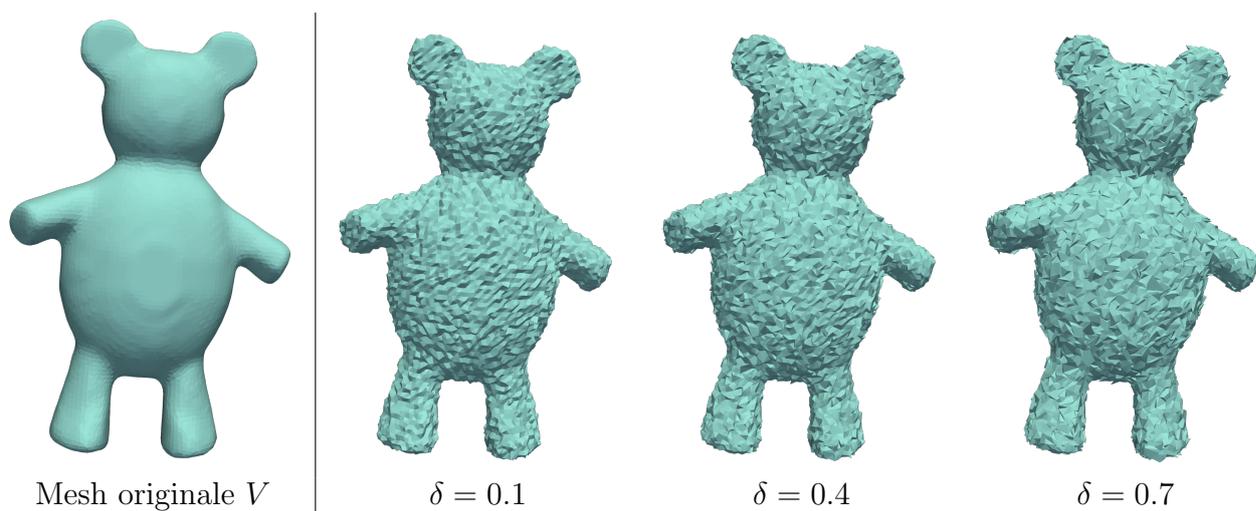


Figura 4.2: L'effetto del rumore in direzione random, fissato  $\gamma = 0.3$ , al variare di  $\delta$ .

### 4.1.1 Figure di merito per valutare l'errore

Un'analisi qualitativa dell'accuratezza delle ricostruzioni può essere ottenuta tramite due metriche di valutazione dell'errore, che misurano quanto la mesh di output differisce da quella originale.

L'**Errore angolare medio (MSAE)** è così definito: per ogni faccia, sia  $\angle(N_\tau^*, N_\tau)$  l'angolo formato dalla normale originaria e da quella ricostruita. L'errore è quindi definito come

$$\mathbf{MSAE} = \mathbf{E}[\angle(N, N^*)^2]$$

Questa metrica misura in particolare l'accuratezza del sottoproblema per  $N$ .

La seconda metrica riguarda invece direttamente i vertici. Poiché le relazioni di appartenenza tra vertici, edge e facce restino invariate, è ben definita la quantità

$$\mathbf{V2V} = \frac{\|V^* - V\|_2}{n_V}$$

che misura direttamente la distanza tra i vertici ricostruiti  $V$  e quelli originari  $V^*$ .

## 4.2 Convergenza empirica

La convergenza dell'algorithmo di ADMM ad un minimo locale della funzione obiettivo  $\mathcal{J}(V; \lambda, a)$  definita in (3.1) può essere verificata numericamente osservando l'evoluzione di  $\mathcal{J}(V; \lambda, a)$  lungo le iterazioni. I grafici in figura 4.3 mostrano come la funzione sia sempre decrescente, fino a stabilizzarsi su un valore limite finale.

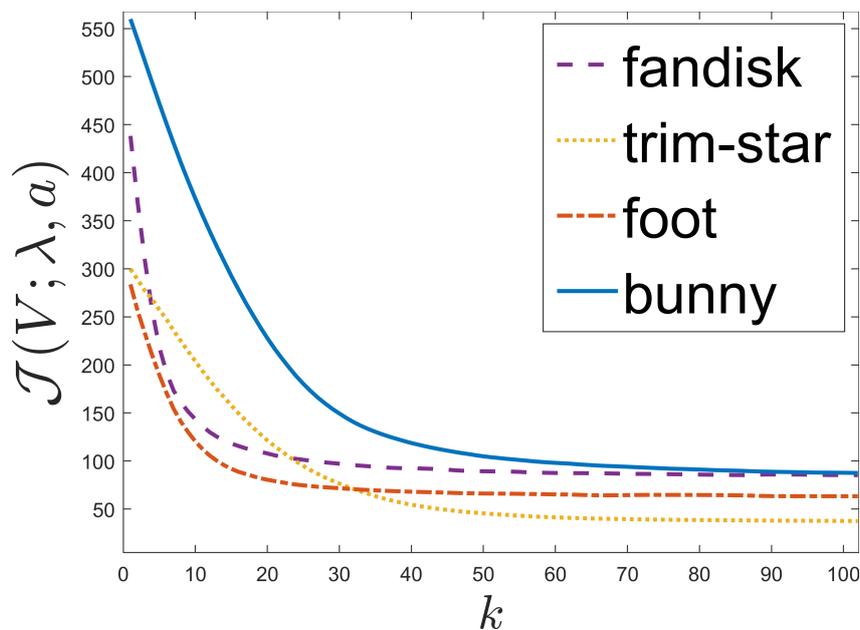


Figura 4.3: Convergenza empirica dell'algorithmo ADMM per alcune mesh.

La figura 4.4 mostra invece la convergenza del sottoproblema per  $V$ , al variare dei metodi risolutivi. Ricordiamo che non è stato possibile dimostrare la convessità della funzione  $J(V)$ , dunque i risultati teorici sulla convergenza lineare della discesa gradiente e superlineare di BFGS non sono utilizzabili.

La discesa gradiente, per costruzione, usa come  $p_j$  le direzioni di maggior discesa della funzione  $J(V)$  e produce una decrescita più regolare rispetto al metodo BFGS. Inoltre il backtracking line search determina una rapida convergenza senza la difficoltà di dover determinare a priori un valore ottimale per lo step.

Le mesh nelle prossime sezioni sono tutte ottenute usando la discesa gradiente con backtracking, sia perché si ottengono ricostruzioni migliori, sia per il minore costo computazionale. Infatti, mentre la discesa gradiente richiede solo il calcolo di  $\nabla J(V)$ , il metodo BFGS necessita anche la costruzione della matrice  $H_j$  e il prodotto matrice per vettore  $p_j = -H_j \nabla J(\bar{V}^{(j)})$ .

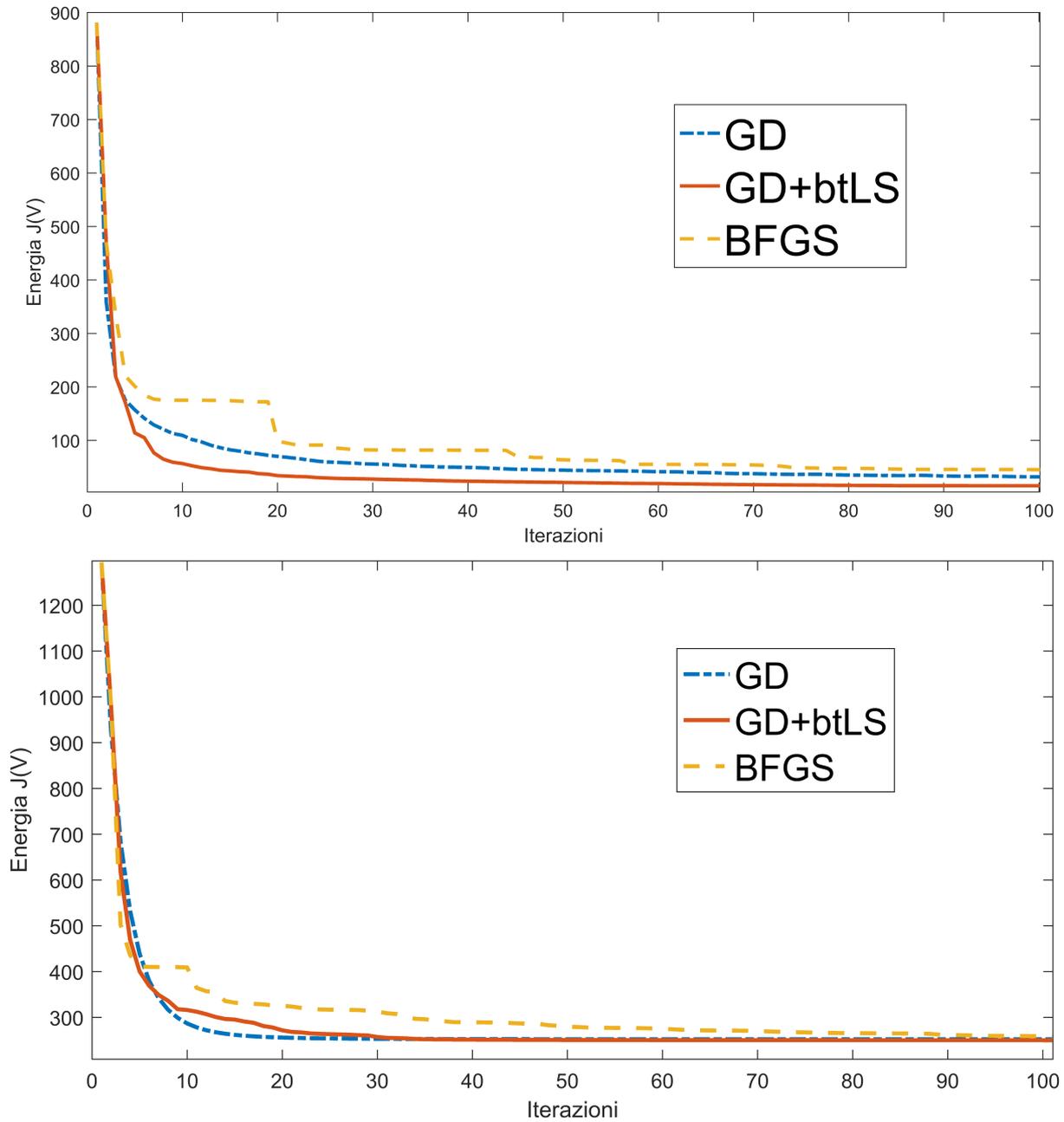


Figura 4.4: Convergenza empirica del sottoproblema per  $V$  per le mesh **Trim-star** e **Block**, con rumore diretto lungo le normali con intensità  $\gamma = 0.2$ . È mostrata l'evoluzione di  $J(V)$  lungo le iterazioni  $j$ , al variare dei metodi di risoluzione: discesa gradiente (GD), con o senza backtracking line search (btLS), e BFGS.

### 4.3 Dipendenza dal parametro $\lambda$

Nella definizione dell'energia  $\mathcal{J}(V; \lambda, a)$  in (3.1), il parametro  $\lambda$  determina il peso relativo del termine di fedeltà ai dati iniziali. Al crescere di  $\lambda$ , l'effetto di regolarizzazione è minore e dunque la ricostruzione tende a restare più simile alla mesh in input, senza riuscire a rimuovere il rumore, soprattutto se di alta intensità. Al contrario, se  $\lambda$  è scelto troppo piccolo, la mesh risultante è fin troppo regolare e le proprietà geometriche dell'originale, come spigoli o creste, non vengono preservate correttamente.

Nella figura 4.5 si osserva come varia la ricostruzione delle mesh **fandisk**, **cube-hole** e **block**, fissando il parametro di concavità  $a$ , per 3 differenti valori di  $\lambda$ . Le mesh in input sono ottenute con un rumore diretto lungo le normali e con intensità  $\gamma$ .

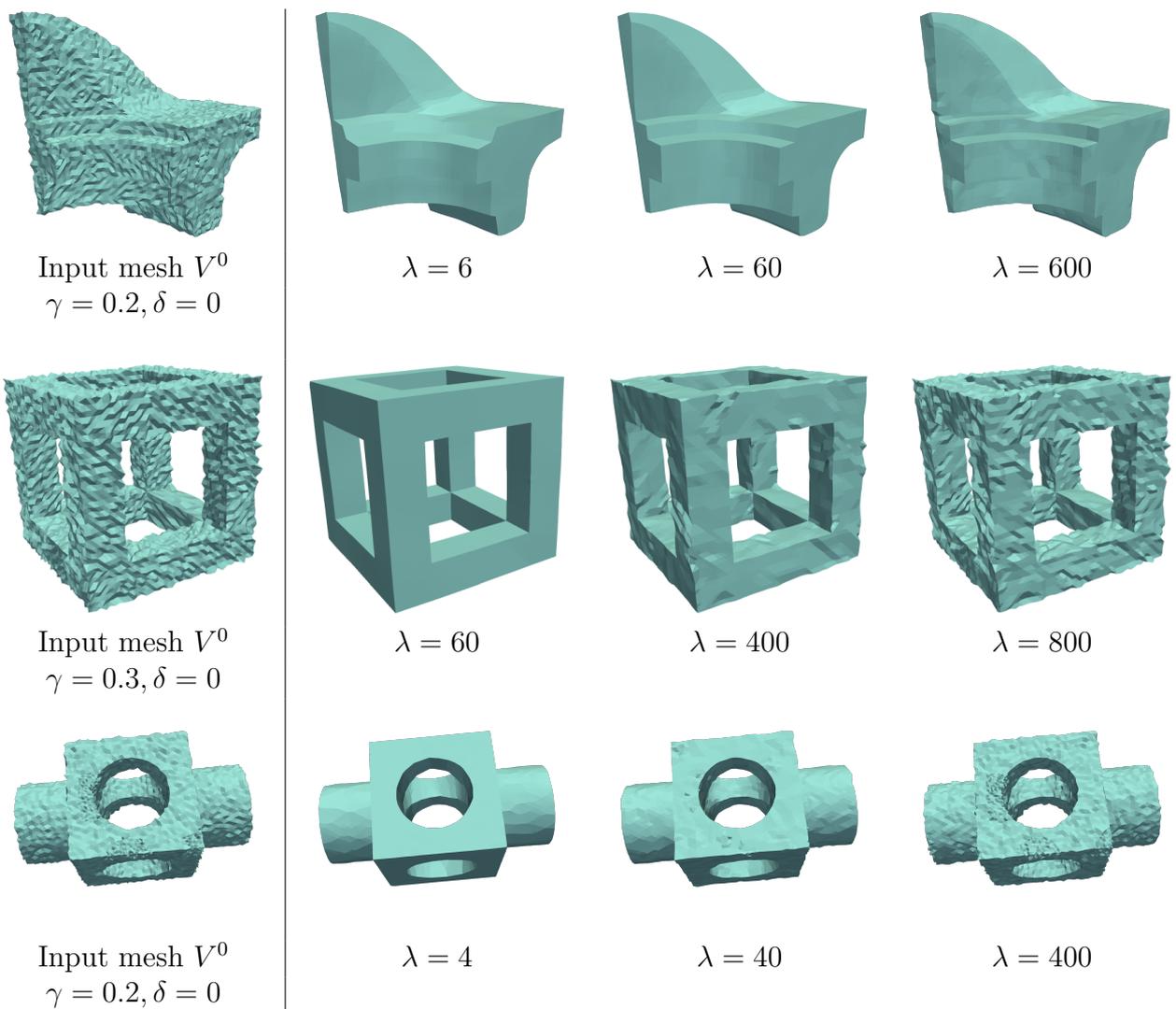


Figura 4.5: Influenza del parametro di fedeltà  $\lambda$  per fissati valori di  $a$ . Prima riga: **Fandisk** con  $a = 0.05$ . Seconda riga: **Cube-hole** con  $a = 0.8$ . Terza riga: **Block** con  $a = 0.5$ .

## 4.4 L'influenza del parametro $a$

Come descritto nella sezione 3.1.1, sono possibili diverse definizioni per il termine di penalità. Il prossimo esempio mostra i vantaggi dati dall'uso della funzione  $\phi(t; a)$ , invece che dalle norme  $\ell_0$  o  $\ell_1$ .

La regolarizzazione  $\ell_0$ , teoricamente ideale per indurre sparsità nel vettore

$$(\|\nabla\mathcal{N}(V)\|_2) = (\|(\nabla\mathcal{N}(V))_1\|_2, \dots, \|(\nabla\mathcal{N}(V))_{n_E}\|_2) ,$$

può produrre errori nella ricostruzione e triangolazioni non uniformi, anche con pieghe o sovrapposizioni di facce e vertici. Inoltre, data la sua natura combinatoria, è computazionalmente costosa.

La regolarizzazione  $\ell_1$  invece tende a sottostimare i valori di intensità maggiore, risultando quindi inefficace in caso di alti livelli di rumore e producendo effetti di contrazione o di staircase.

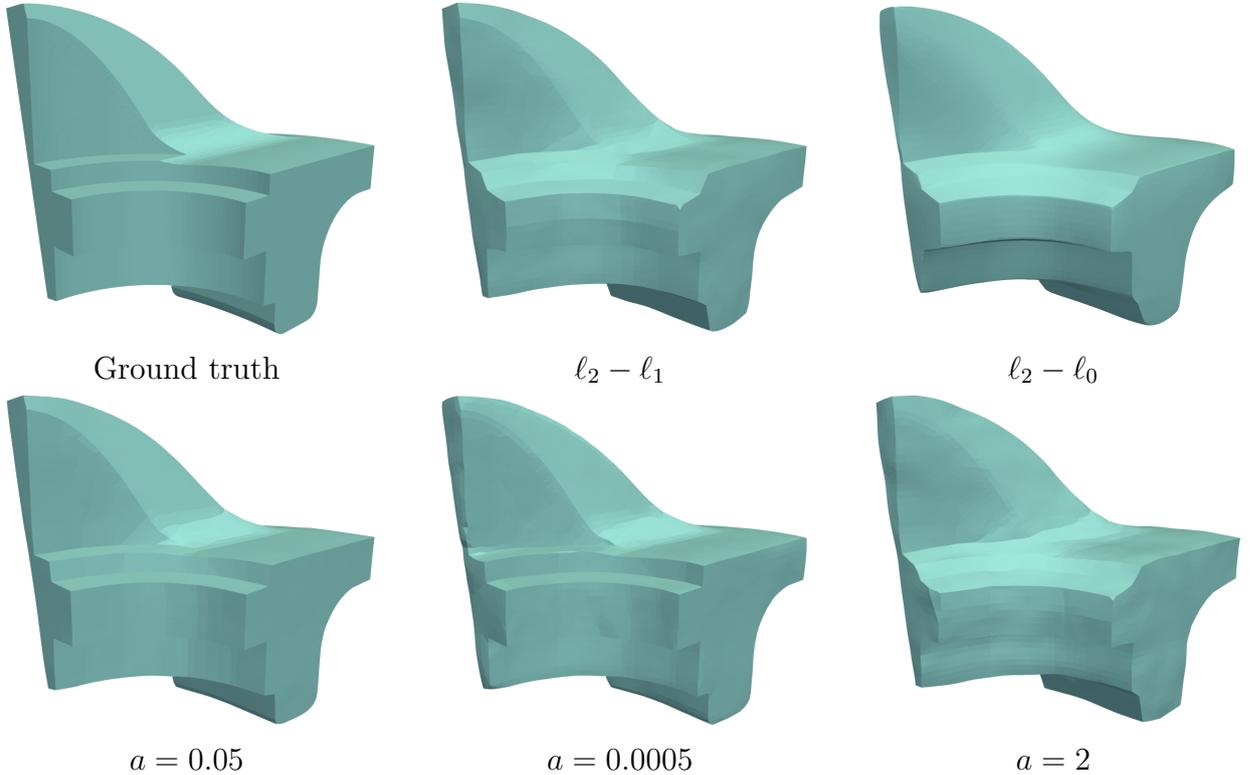


Figura 4.6: Influenza del parametro  $a$  e confronto con  $\ell_0$  e  $\ell_1$ . Le ricostruzioni sono ottenute a partire da una mesh in input con rumore di parametri  $\gamma = 0.2$  e  $\delta = 0$ .

Invece, la funzione  $\phi(t; a)$  permette, tramite il parametro di  $a$ , di regolare il grado di non-convessità e controllare più accuratamente la sparsità del vettore  $\|\nabla\mathcal{N}(V)\|_2$ . Il risultato è una ricostruzione che riesce a preservare gli spigoli della superficie, anche con alti livelli di rumore, e che produce una triangolazione più uniforme. Nella figura 4.6, sono presentati la mesh originale **fandisk** e le sue ricostruzioni tramite regolarizzazione  $\ell_0$ ,  $\ell_1$

e  $\phi$ , a partire da una mesh in input con rumore diretto lungo le normali con intensità  $\gamma = 0.2$ .

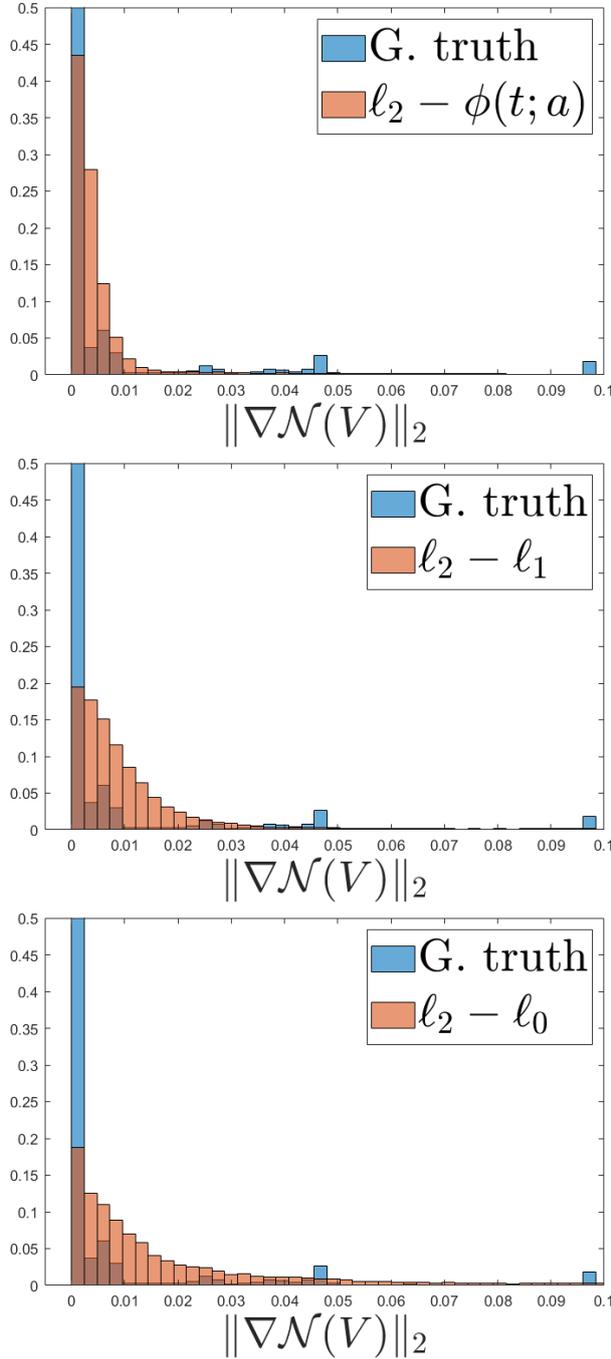


Figura 4.7: Istogrammi di  $\|\nabla\mathcal{N}(V)\|_2$  per la mesh originale e, rispettivamente, per le ricostruzioni con  $\phi(t; 0.05)$ ,  $l_1$ ,  $l_0$ .

La ricostruzione  $l_1$  produce notevoli effetti di staircase, mentre  $l_0$  non riesce a preservare spigoli e creste della superficie.

Le tre ricostruzioni con  $\phi$  mostrano l'influenza del parametro  $a$ . La funzione  $\phi(t; a)$  tende ad approssimare la norma  $l_1$  per  $a \rightarrow 0$  e la norma  $l_0$  per  $a \rightarrow +\infty$ . La scelta ottimale, cioè  $a = 0.05$ , genera una ricostruzione precisa sia nelle aree piatte, sia in quelle curve e mantiene correttamente le caratteristiche geometriche della superficie.

A conferma di queste osservazioni, gli istogrammi in figura 4.7 evidenziano la maggiore accuratezza della ricostruzione con  $\phi$  rispetto a quelle con  $l_1$  e  $l_0$ .

## 4.5 Confronto con altri modelli

Le prossime figure metteranno a confronto il nostro modello con alcuni dei principali metodi di ricostruzione. In dettaglio:

- [15] - X. Sun, P. Rosin, R. Martin, and F. Langbein, *Fast and effective feature-preserving mesh denoising*. IEEE Trans. Vis. Comput. Graphics, vol. 13, no. 5, pp. 925–938, Sept./Oct. 2007
- [18] Youyi Zheng, Hongbo Fu, Oscar Kin-Chung Au, and Chiew-Lan Tai. 2011. Bilateral Normal Filtering for Mesh Denoising. IEEE Transactions on Visualization and Computer Graphics 17, 10 (October 2011), 1521-1530.
- [5] Lei He and Scott Schaefer. 2013. Mesh denoising via L0 minimization. ACM Trans. Graph. 32, 4, Article 64 (July 2013).
- [19] Wangyu Zhang, Bailin Deng, Juyong Zhang, Sofien Bouaziz, Ligang Liu. Guided mesh normal filtering. Pacific Graphics 2015.

I relativi algoritmi, disponibili sul sito <https://github.com/bldeng/GuidedDenoising>, sono stati realizzati dagli autori di [19].

Le ricostruzioni sono ottenute scegliendo opportunamente i rispettivi parametri e considerando i risultati visivamente migliori.

Nelle figure 4.8, 4.9 e 4.10 le mesh sono colorate in base alla curvatura media. Inoltre, negli zoom sono mostrate alcune sezioni di ogni mesh, evidenziandone i vertici e gli edge.

Sulle aree piatte o lisce delle mesh il nostro modello ottiene risultati paragonabili a quelli degli altri metodi, come visibile in figura 4.8.

Per quanto riguarda la corretta ricostruzione degli spigoli, [15] e [18] non riescono a preservarli, mentre [5] e [19] producono triangolazioni poco uniformi, con facce degeneri e sovrapposizioni di vertici. Il nostro modello, invece, non soffre di questi problemi, come mostrato negli zoom della figura 4.9.

La figura 4.10 sottolinea come queste proprietà continuino a valere non solo nel caso di spigoli netti, ma anche per quelli meno evidenti, come la cresta in **Fandisk**.

Infine, nella figura 4.11, la mesh in input, corrotta da rumore in direzione random, con  $\delta = 0.4$  e  $\gamma = 0.3$ , presenta molte facce sovrapposte. La nostra ricostruzione riesce meglio delle altre a rendere più regolare e uniforme la triangolazione.

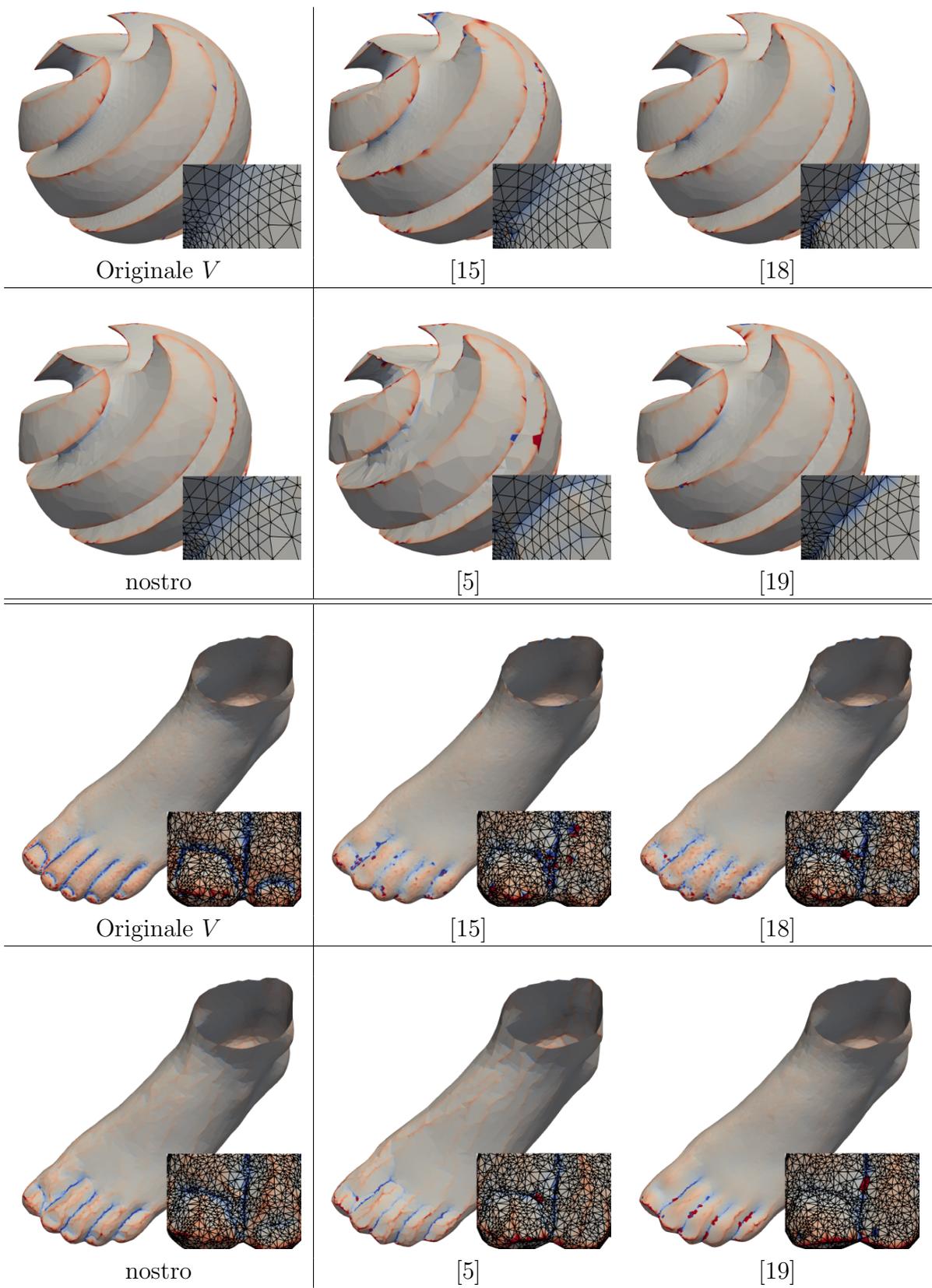


Figura 4.8: Ricostruzioni di Sharp-sphere e Foot, con  $\delta = 0$  e  $\gamma$  rispettivamente di valori  $\gamma = 0.15$  e  $\gamma = 0.2$ .

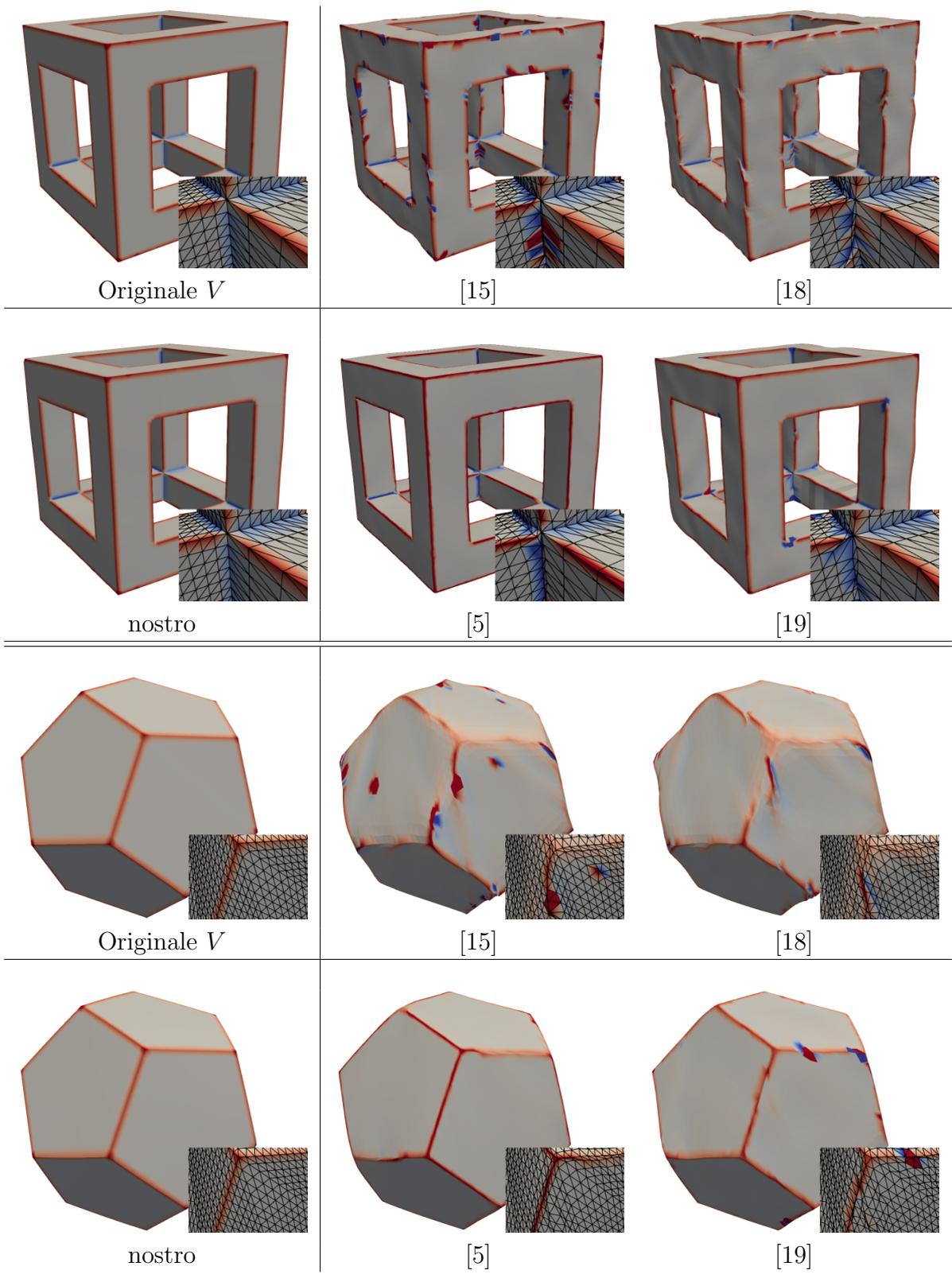


Figura 4.9: Ricostruzioni di Cube-hole e Twelve, in entrambi i casi con  $\gamma = 0.3$  e  $\delta = 0$ .

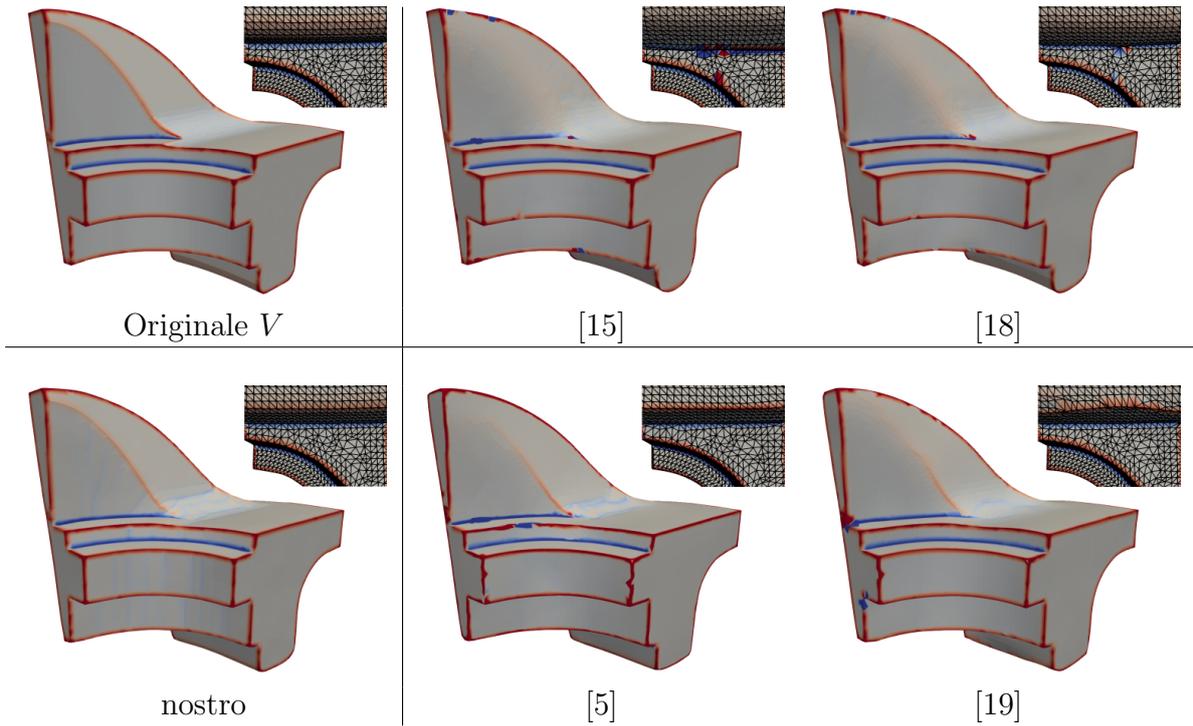


Figura 4.10: Ricostruzioni di **fandisk**, con  $\gamma = 0.2$  e  $\delta = 0$ .

A prova della robustezza del nostro modello, le figure 4.12, 4.13 e 4.14 riportano le ricostruzioni di altre mesh, con rumore di intensità crescente  $\gamma = \{0.2, 0.3, 0.4, 0.5, 0.6\}$ , diretto lungo le normali.

In aggiunta al confronto puramente grafico, in queste figure ad ogni ricostruzione sono associate le relative metriche d'errore MSAE ( $\times 10^2$ ) e V2V ( $\times 10^6$ ). Anche quantitativamente, si osserva come il nostro modello approssimi efficacemente sia i vertici sia le normali della mesh originale.

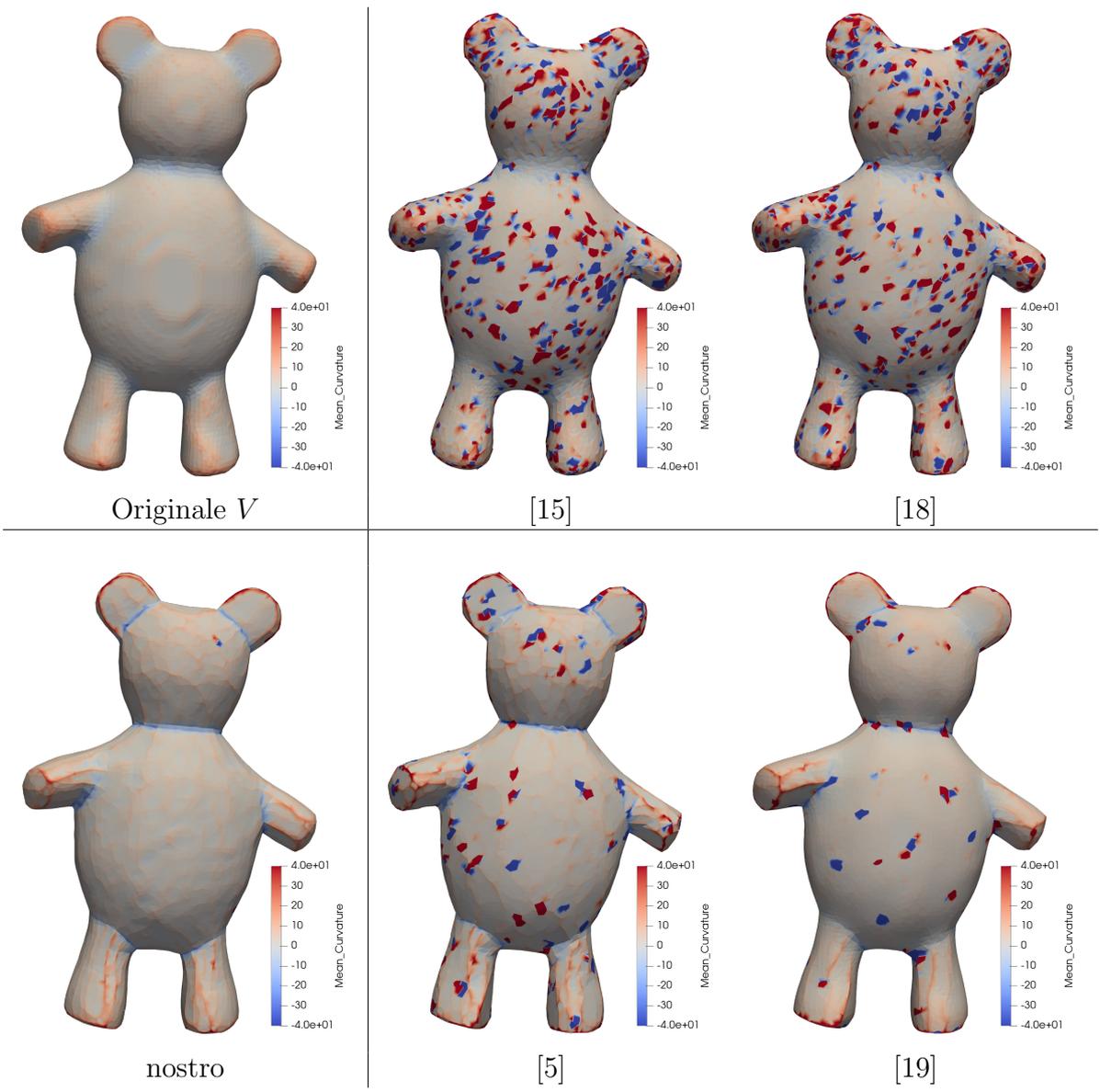


Figura 4.11: Ricostruzioni di Teddy, con  $\gamma = 0.3$  e  $\delta = 0.4$ .

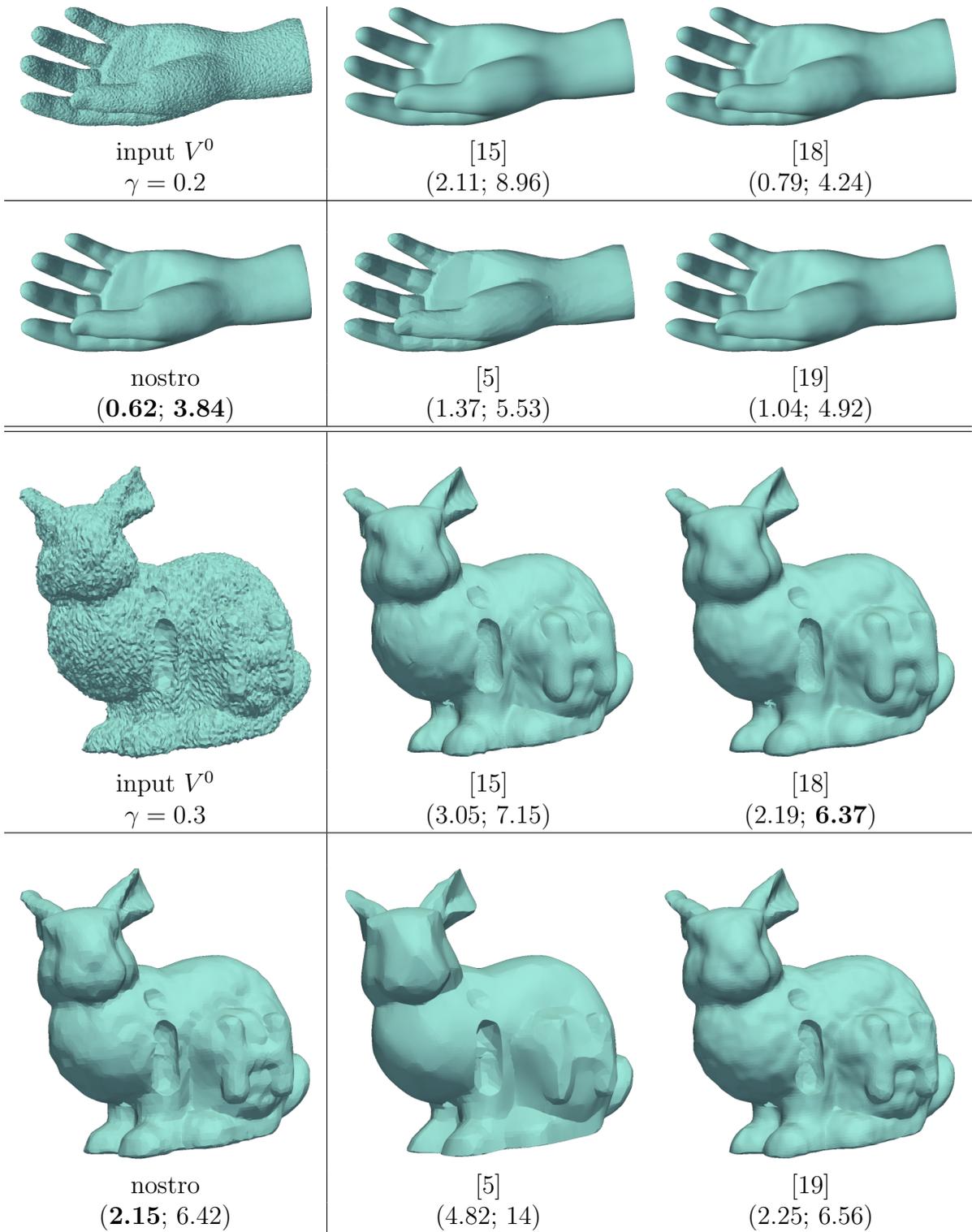


Figura 4.12: Ricostruzioni di Hand e Bunny, con  $\delta = 0$  e rumore in direzione normale.

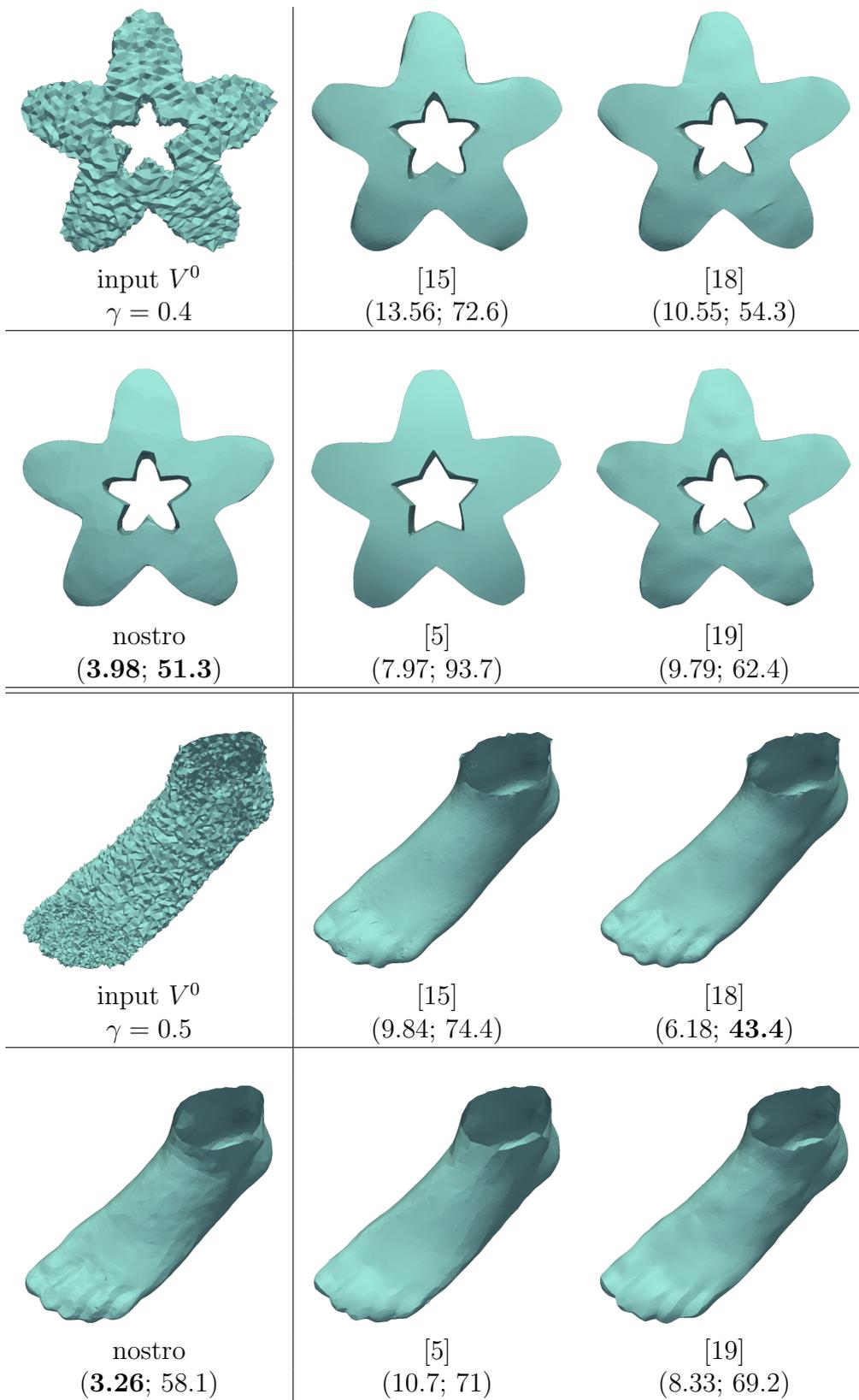


Figura 4.13: Ricostruzioni di Trim-star e Foot, con  $\delta = 0$  e rumore in direzione normale.

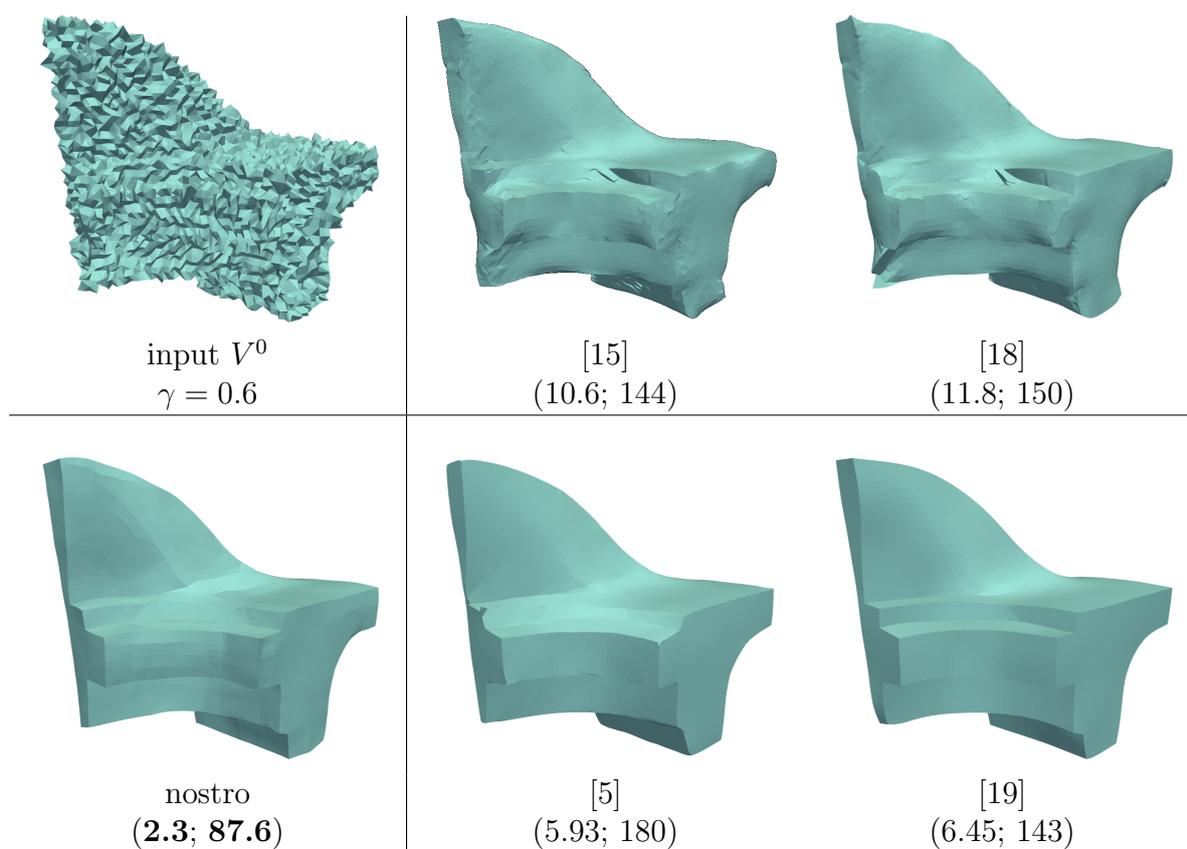


Figura 4.14: Ricostruzioni di Fandisk, con  $\delta = 0$  e rumore in direzione normale

Fino a questo momento, le mesh sono state perturbate solo in direzione normale, mantenendo  $\delta = 0$ . Le figure 4.15 e 4.16 mostrano invece le ricostruzioni delle mesh *Julius* e *Teddy* a partire da un rumore in direzione random. Infine, le figure 4.17 e 4.18 prendono in input mesh ottenute da dati reali, attraverso uno scanner 3D. In questo caso, naturalmente, le metriche di errore non sono ben definite poiché non sono note le posizioni originali dei vertici.

A livello qualitativo, si può osservare che il nostro modello è efficace nel rimuovere il rumore e preservare al contempo gli spigoli dell'oggetto.

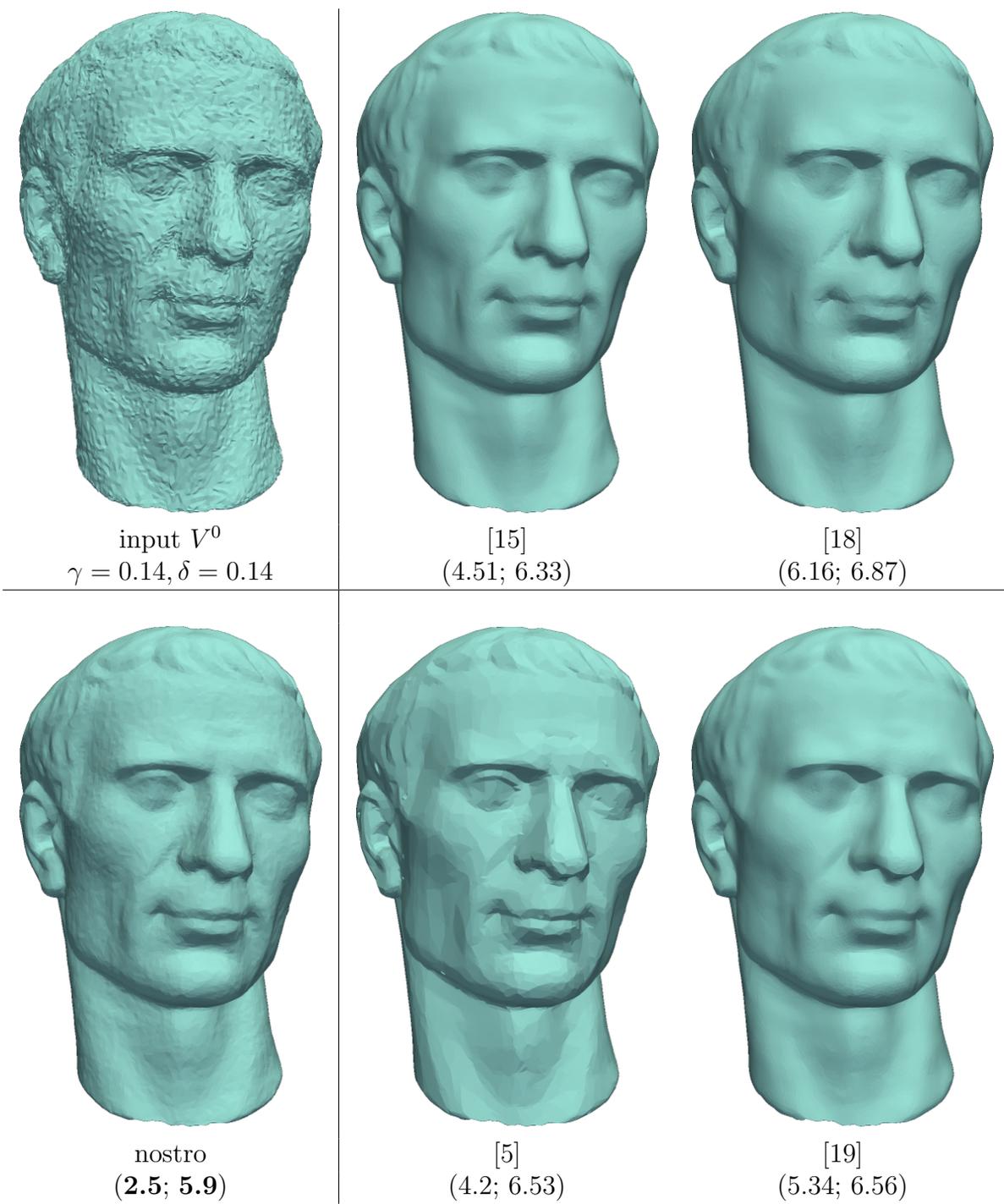


Figura 4.15: Ricostruzioni di Julius, con rumore di parametri  $\gamma = 0.14, \delta = 0.14$ .

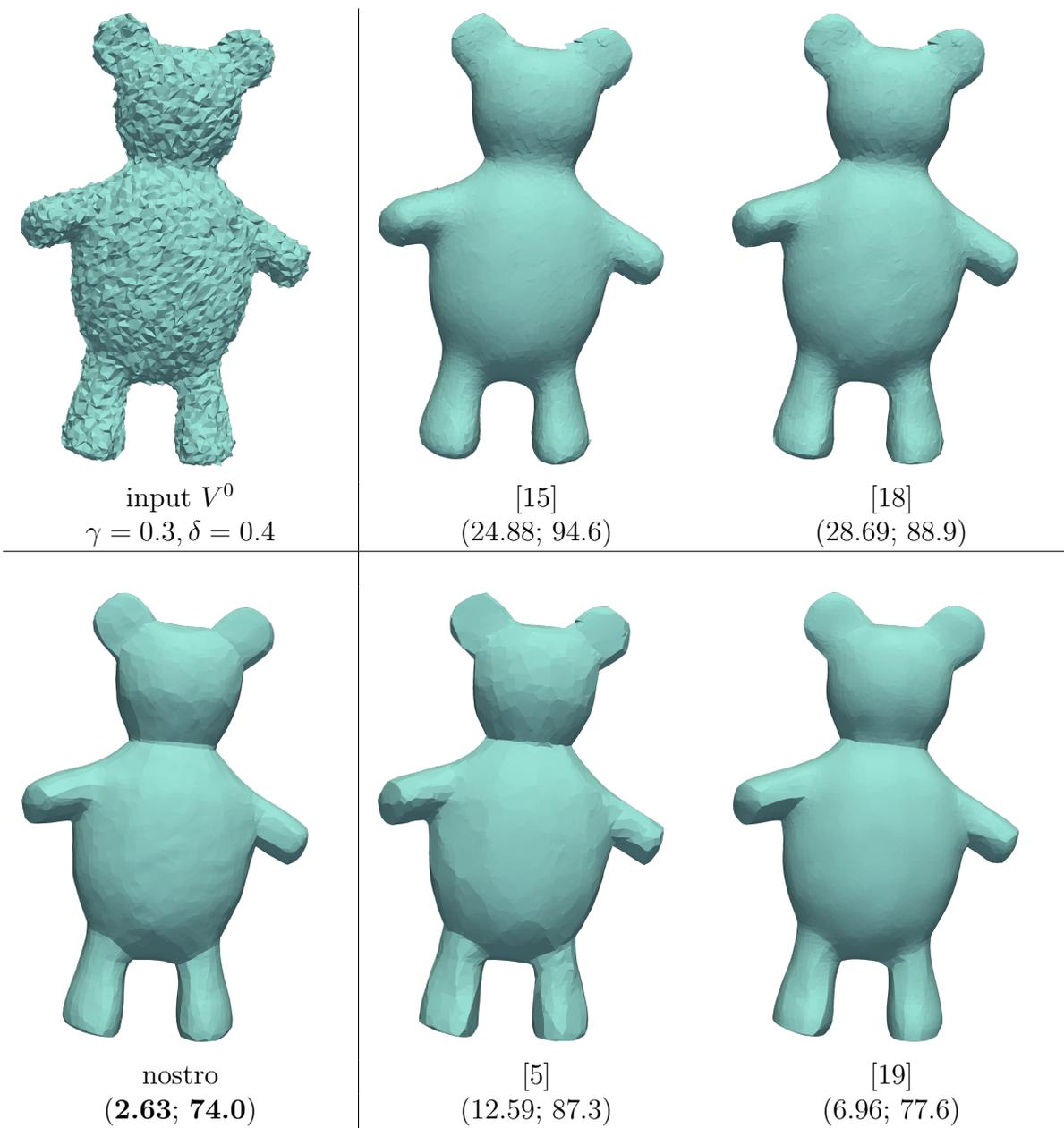


Figura 4.16: Ricostruzioni di Teddy, con rumore di parametri  $\gamma = 0.3$  e  $\delta = 0.4$ .



input  $V^0$



nostro



[15]



[18]

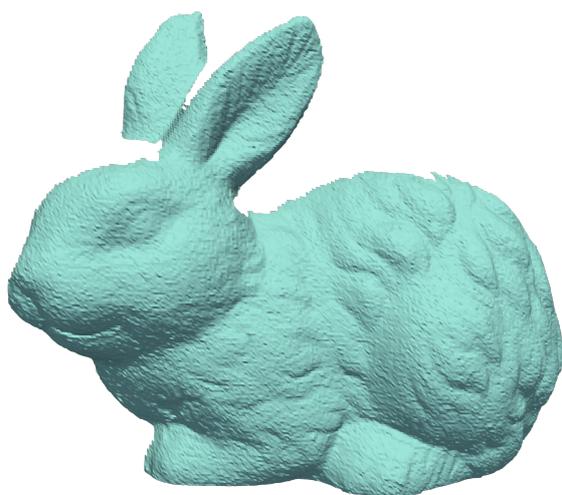


[5]

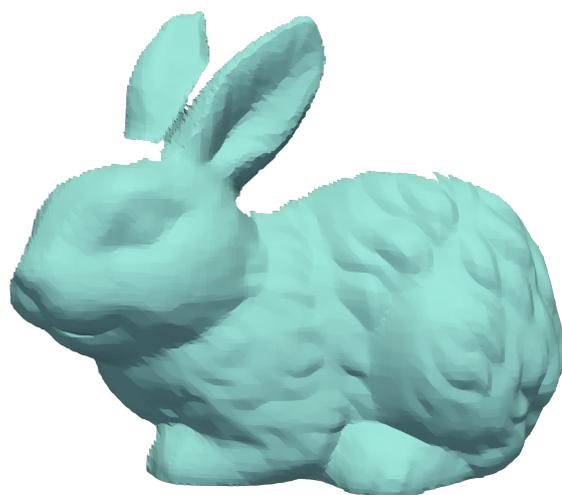


[19]

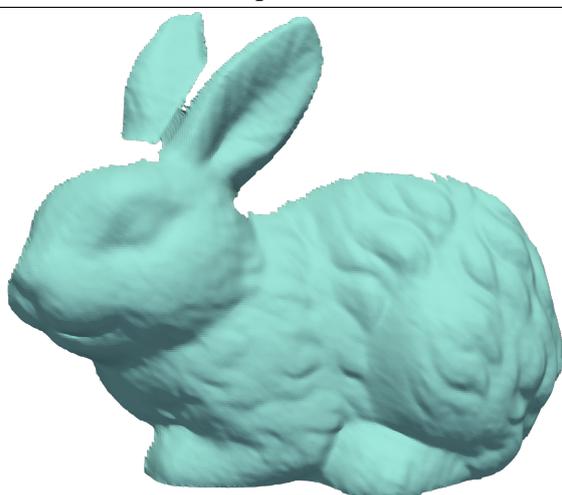
Figura 4.17: Ricostruzioni della mesh reale Angel.



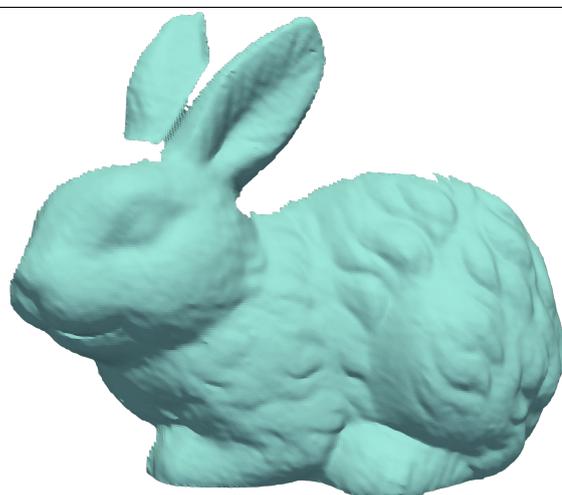
input  $V^0$



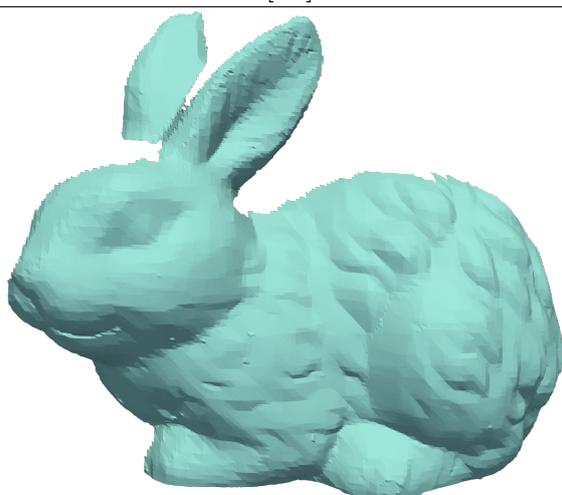
nostro



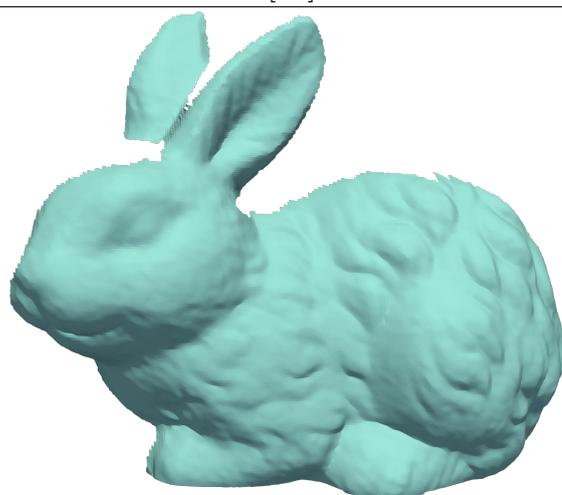
[15]



[18]



[5]



[19]

Figura 4.18: Ricostruzioni della mesh reale Rabbit.



# Conclusione

In questa tesi, abbiamo proposto un modello variazionale per il denoising di mesh, in grado di rimuovere il rumore e preservare le proprietà geometriche della superficie. Abbiamo definito un'unica funzione costo, come somma di un termine di fedeltà ai dati iniziali, in norma  $\ell_2$  con parametro  $\lambda$ , e di un termine regolarizzante, che dipende da  $\nabla\mathcal{N}(V)$ , gradiente del campo normale, attraverso una funzione  $\phi(t; a)$ .

Quest'ultima funzione, una versione riscalata e riparametrizzata della Minimax Concave Penalty (MCP), è stata scelta per la sua "elasticità", poiché il parametro  $a$  permette di regolarne il grado di non-convessità e di controllare la sparsità di  $\nabla\mathcal{N}(V)$ .

Il gradiente della normale è stato discretizzato definendolo su ogni edge della mesh come la differenza tra le normali alle facce cui l'edge appartiene, pesata con la lunghezza dell'edge.

Data la non differenziabilità di  $\phi(t; a)$ , l'ottimizzazione è stata sviluppata tramite l'ADMM, costruendo un processo iterativo costituito da tre sottoproblemi di minimo, rispettivamente nelle variabili  $V$ ,  $N$ ,  $t$ . I problemi relativi a  $N$  e  $t$  sono stati risolti in maniera esatta, calcolando esplicitamente i punti di minimo. Invece, il problema in  $V$  è risolto iterativamente, calcolando le direzioni di discesa tramite discesa gradiente oppure metodo BFGS, mentre gli step sono scelti tramite ricerca lineare con backtracking oppure considerati costanti.

La convergenza sia dell'iterazione interna, relativa al sottoproblema per  $V$ , sia dell'iterazione esterna, legata all'ADMM, è dimostrata per via empirica, osservando l'evoluzione lungo le iterazioni delle rispettive funzioni costo.

I risultati sperimentali mostrano che il modello produce ricostruzioni più precise rispetto a quelle ottenute con altri metodi di denoising, sia con bassi che con alti livelli di rumore. In particolare, riesce a preservare più efficacemente gli spigoli e gli angoli delle superfici, anche quando non particolarmente netti. Inoltre, non produce falsi edge e l'effetto staircase è notevolmente ridotto, anche grazie alla possibilità di regolare la non-convessità del termine di penalità attraverso il parametro  $a$ .

Questo miglioramento appare ancora più evidente se il rumore non è diretto lungo le normali, ma in direzioni random. In questo caso, infatti, gli altri modelli ottengono ricostruzioni definite da triangolazioni poco regolari, con molte facce sovrapposte o degeneri. Il nostro metodo, invece, riesce non solo a ottenere risultati visivamente migliori, ma anche a rendere la triangolazione più uniforme.

Accanto al confronto qualitativo, per dare una valutazione quantitativa dei risultati abbiamo usato due metriche di errore, che hanno confermato l'accuratezza delle ricostruzioni ottenute.



# Ringraziamenti

Desidero ringraziare la Professoressa Morigi, per avermi guidato e supportato in questo percorso con estrema dedizione e professionalità. La ringrazio per aver rafforzato il mio interesse verso l'analisi numerica e, più in generale, verso il campo della matematica applicata. Ringrazio Martin per tutti i suggerimenti, i consigli, i feedback e le risposte ad ogni mio dubbio. La vostra costante disponibilità ha permesso di abbattere l'ostacolo rappresentato dai 1000Km che mi separavano da Bologna.

Ringrazio mio padre per la continua spinta a diventare uno studente e soprattutto una persona migliore. Ringrazio mia madre per l'immane supporto, testimoniato dai quotidiani "come stai?".

Ringrazio infine il ClubOplà = {*Giacomo, Fede, Salva, Giova*} per essere stati fonte di svago, relax e divertimento in un periodo altrimenti grigio e monotono. Ogni videochiamata e ogni "allora sabato corriamo?" sono stati preziosi come ossigeno per riprendere fiato e recuperare energie fisiche e mentali.



# Bibliografia

- [1] H.Hoppe, M.Eck; *Automatic Reconstruction of B-spline surfaces of Arbitrary Topological Types*. Computer Graphics Proceedings, 1996.
- [2] M.Desbrun, M.Meyer, P.Schroder, A.H.Barr; *Implicit Fairing of Irregular Meshes using Diffusion and Curvature Flow*. ACM SIGGRAPH, 1999
- [3] Koji Fujiwara. *Eigenvalues of Laplacians on a closed riemannian manifold and its nets*. In Proceedings of AMS 123, pages 2585–2594, 1995
- [4] G. Taubin; *A signal processing approach to fair surface design*. ACM SIGGRAPH, 1995
- [5] L.He, S.Schaefer; *Mesh Denoising via  $\ell_0$  Minimization*. ACM Trans. Graph. 2013
- [6] Wu X., Zheng J., Cai Y., Fu C.W.; *Mesh Denoising using Extended ROF Model with  $\ell_1$  Fidelity*. Comput. Graph. Forum, 2015
- [7] H. Zhang, C. Wu, J. Zhang, J. Deng; *Variational Mesh Denoising Using Total Variation and Piecewise Constant Function Space*. IEEE Transactions on Visualization and Computer Graphics, 2015
- [8] L.Zheng, L.Rongjie, H.Zhang, C.Wu; *Triangulated Surface Denoising using High Order Regularization with Dynamic Weights*. SIAM Journal on Scientific Computing, 2017
- [9] Zhong S, Xie Z, Liu J, Liu Z.; *Robust Mesh Denoising via Triple Sparsity*. Sensors, 2019
- [10] Lu X., Deng Z., Chen W.; *A Robust Scheme for Feature-Preserving Mesh Denoising*. IEEE Trans. Vis. Comput. Graph, 2015
- [11] Lu X., Chen W., Schaefer S.; *Robust mesh denoising via vertex pre-filtering and  $\ell_1$ -median normal filtering*. Comput. Aided Geom. Des. 2017
- [12] C. H. Zhang; *Nearly unbiased variable selection under minimax concave penalty*. Annals of Statistics, 2010
- [13] P.Y. Chen, I.W. Selesnick; *Group-sparse signal denoising: non-convex regularization, convex optimization*. IEEE Trans. on Sign. Proc., 2014

- [14] G. Taubin, *Linear anisotropic mesh filters*. IBM Research Report RC22213(W0110-051), IBM T.J. Watson Research, October 2001.
- [15] X. Sun, P. Rosin, R. Martin, and F. Langbein, *Fast and effective feature-preserving mesh denoising*. IEEE Trans. Vis. Comput. Graphics, vol. 13, no. 5, pp. 925–938, Sept./Oct. 2007
- [16] Shachar Fleishman, Iddo Drori, and Daniel Cohen-Or. *Bilateral mesh denoising*. ACM Trans. Graph. 22, 3 (July 2003), 950-953.
- [17] Thouis R. Jones, Frédo Durand, and Mathieu Desbrun. *Non-iterative, feature-preserving mesh smoothing*. ACM Trans. Graph. 22, 3 (July 2003), 943-949.
- [18] Youyi Zheng, Hongbo Fu, Oscar Kin-Chung Au, and Chiew-Lan Tai. *Bilateral Normal Filtering for Mesh Denoising*. IEEE Transactions on Visualization and Computer Graphics 17, 10 (October 2011), 1521-1530.
- [19] Wangyu Zhang, Bailin Deng, Juyong Zhang, Sofien Bouaziz, Ligang Liu. *Guided mesh normal filtering*. Pacific Graphics 2015.
- [20] Jorge Nocedal, Stephen Wright, *Numerical Optimization*. Springer, 2006
- [21] Michael Shub. *Global stability of dynamical systems*. Springer Science, Business Media, 1987.
- [22] J. E. Dennis, Jorge J. Moré. *A Characterization of Superlinear Convergence and Its Application to Quasi-Newton Methods*. Mathematics of Computation, Apr. 1974.
- [23] C. G. Broyden, J. E. Dennis, J. J. Moré. *On the Local and Superlinear Convergence of Quasi-Newton Methods*, Cornell Computer Science Technical Report 72-137, 1972.
- [24] M. J. D. Powell. *On the convergence of the variable metric algorithm*. J. Inst. Math. Appl., v. 7, 1971, pp. 21-36.
- [25] J. Lee, M. Simchowitz, M. Jordan, B. Recht. *Gradient Descent Converges to Minimizers*. Proc. Conf. Comput. Learn. Theory, 2016
- [26] Tao Min, Yang Junfeng. *Alternating direction algorithms for total variation deconvolution in image reconstruction*. Optimization Online. 2009
- [27] D. Gabay, B. Mercier. *A dual algorithm for the solution of nonlinear variational problems via finite element approximation*. Computers Mathematics with Applications, Volume 2, Issue 1, Pages 17-40, 1976.
- [28] R. Glowinski, A. Marrocco. *Sur l'approximation par Elements finis d'ordre 1 et la resolution par penalisation-dualité, d'une classe de problemes de Dirichlet non-lineaires*. C.R. hebd. SLanc. Acad. Sci., Paris. 278, Sér. A, 1649-1652. 1974.

- [29] M. Hestenes. *Multiplier and gradient methods*. JOTA 4(5). 1969.
- [30] M. J. D. Powell. *A method for nonlinear optimization in minimization problems*. Optimization (R. Fletcher Ed.) Academic Press. 1969
- [31] Morigi, S., Rucci, M., Sgallari, F.: *Nonlocal surface fairing*. In: Bruckstein, A.M., ter Haar Romeny, B.M., Bronstein, A.M., Bronstein, M.M. (eds.) SSVM. pp. 38-49. Springer Berlin Heidelberg, Berlin, Heidelberg. 2012