

UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Matematica "Tullio Levi-Civita"

Corso di Laurea Triennale in Matematica

Tesi di Laurea

Sulla filtered hyperinterpolation in domini multivariati

Relatore

prof. Alvisè Sommariva

Laureando

Enrico Caregnato

1220509

Anno Accademico 2021/2022

16/12/2022

Sommario

Il proposito di questo lavoro consiste nel discutere alcuni tipi di iperinterpolazione su triangoli sferici, confrontandone le peculiarità ed evidenziando i loro vantaggi e svantaggi.

A tal proposito ricordiamo che il concetto di *iperinterpolazione* è stato introdotto nel 1995 in un lavoro di Ian Sloan. La chiara intenzione dell'autore consiste nell'approssimare una funzione continua tramite minimi-quadrati discreti, costruiti con una formula di quadratura con grado di precisione sufficientemente alto. In particolare, per approssimare i coefficienti di Fourier rispetto a una base polinomiale di grado totale n , l'autore considera una formula a pesi positivi e nodi interni, avente grado algebrico di precisione $2n$. In tale lavoro, mostra pure che con questa scelta l'errore in norma L^2 converge a zero al tendere di n a $+\infty$, e come in certi casi l'approssimante introdotte coincida con una interpolante.

Qualche anno dopo, nel 2012, introdusse in domini sferici la cosiddetta *iperinterpolazione filtrata*, in cui i coefficienti di Fourier calcolati numericamente sono *smorzati*, utilizzando opportunamente una funzione filtro. Inizialmente si applicò tale teoria a domini sferici, passando successivamente a casi più generali.

I due approcci, pur molto diversi, hanno in comune che le loro approssimazioni a grado n risultano proiezioni su spazi polinomiali, rispettivamente di grado n e $\lfloor n/2 \rfloor$. Si mostra teoricamente il vantaggio dell'interpolazione filtrata qualora la funzione analizzata sia soggetta a un rumore non trascurabile, questione evidenziata in alcuni articoli su classici domini quale intervallo, disco, sfera, cubo, in cui le basi ortonormali sono note.

In questo lavoro consideriamo l'applicazione di tali due approcci a un caso meno classico quale un generico triangolo sferico. Lo svantaggio rispetto ai domini citati, è la mancanza di una esplicita base ortonormale $\{p_k\}$, che viene calcolata numericamente, essendo peraltro di tipo *triangolare*, proprietà che si mostra essere fondamentale per poter calcolare l'iperinterpolante filtrata. Quale formula di quadratura a pesi positivi utilizziamo quella recentemente introdotta nel lavoro *Near-algebraic Tchakaloff-like quadrature on spherical triangles*.

Abbiamo implementato in MATLAB dei codici che permettano il calcolo delle approssimanti, verificando numericamente le proprietà di proiezione ed evidenziando la capacità di fornire una migliore approssimazione da parte della variante filtrata nel caso di funzioni soggette a rilevanti rumori di tipo gaussiano o impulsivo, come dichiarato dalla teoria.

I codici utilizzati sono descritti nell'appendice di questo lavoro e resi disponibili open-source.

Indice

Indice	ii
1 Introduzione	1
1.1 Nozioni preliminari	1
2 Iperinterpolazione	3
2.1 Analisi dell'errore	4
3 Iperinterpolazione filtrata	7
3.1 Analisi dell'errore	10
4 Problemi di approssimazione ai minimi quadrati	15
4.1 Problema dei minimi quadrati "standard"	15
4.2 Problema dei minimi quadrati regolarizzati	16
5 Esempi di approssimazione su domini multivariati	19
5.1 L'intervallo	19
5.2 Il disco unitario	20
5.3 La sfera unitaria	20
5.4 Triangoli sferici	21
6 Risultati numerici	25
A Codici	33
Bibliografia	45

Capitolo 1

Introduzione

In questa tesi viene presa in considerazione una tecnica di approssimazione polinomiale detta *iperinterpolazione* e in particolare ci si sofferma su una sua variante, la *iperinterpolazione filtrata*, andando a studiare gli aspetti per i quali questa seconda versione risulta competitiva.

L'iperinterpolazione è stata introdotta da I. Sloan nel 1995 (cf. [8]). L'approssimante di una data funzione continua f in un sottoinsieme compatto di \mathbb{R}^n viene determinata tramite opportune formule di quadratura approssimando i coefficienti, espressi nella forma di Fourier, della proiezione ortogonale di f dallo spazio delle funzioni continue allo spazio dei polinomi \mathbb{P}_L di grado totale al più L .

La iperinterpolazione filtrata, invece, prevede di filtrare i coefficienti dell'approssimante mediante alcune *funzioni filtro*, processo che rende il metodo talora più efficiente quando si utilizzano dati soggetti a rumore.

Per quanto riguarda la struttura di questa tesi, dopo i primi capitoli in cui verranno presentati i due metodi di approssimazione, l'attenzione verterà su uno studio della relazione tra problemi ai minimi quadrati e iperinterpolazione filtrata; infine verranno proposti degli esempi numerici di applicazioni della iperinterpolazione filtrata su domini multivariati, che mettano in evidenza quanto descritto nei capitoli precedenti.

1.1 Nozioni preliminari

Per iniziare la trattazione dell'argomento vengono richiamate alcune definizioni di base e concetti fondamentali riguardanti la teoria dell'approssimazione.

Si consideri un dominio compatto $\Omega \subset \mathbb{R}^n$, $n \geq 1$ e si assuma che Ω abbia misura finita rispetto a una misura data $d\omega$, ovvero sia tale che

$$\int_{\Omega} d\omega = V < +\infty.$$

L'obiettivo è quello di approssimare una data funzione $f \in C(\Omega)$ (eventualmente soggetta a rumore), con un polinomio in \mathbb{P}_L , che è lo spazio dei polinomi di grado al più L .

Si definisce quindi il prodotto interno, rispetto alla misura $d\omega$, tra due funzioni u e v su Ω come

$$\langle u, v \rangle = \int_{\Omega} uv \, d\omega. \tag{1.1}$$

Osservando che tale definizione coinvolge il calcolo di un integrale, in vista del fatto che si cerca una soluzione numerica al problema, si assume sia data una formula di quadratura del tipo

$$\sum_{j=1}^N w_j g(\mathbf{x}_j) \approx \int_{\Omega} g d\omega, \quad (1.2)$$

dove $\chi_n = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \subset \Omega$ è l'insieme dei punti di campionamento e i w_j sono in pesi di quadratura, positivi per ogni $1 \leq j \leq N$. Si richiede inoltre che tale formula abbia la proprietà di essere esatta per tutti i polinomi di grado $\leq 2L$, vale a dire

$$\sum_{j=1}^N w_j g(\mathbf{x}_j) = \int_{\Omega} g d\omega \quad \forall g \in \mathbb{P}_{2L}. \quad (1.3)$$

Basandosi su queste ipotesi, si definisce il "prodotto interno discreto"

$$\langle u, v \rangle_N = \sum_{j=1}^N w_j u(\mathbf{x}_j) v(\mathbf{x}_j), \quad (1.4)$$

che corrisponde al prodotto definito in (1.1), dove l'integrale esatto è stato sostituito dalla formula di quadratura (1.2).

Capitolo 2

Iperinterpolazione

Sia $\Omega \subset \mathbb{R}^n$ un insieme compatto, $\mathbb{P}_L = \mathbb{P}_L(\Omega)$ lo spazio dei polinomi di grado totale n su Ω . Sia $d = \dim(\mathbb{P}_L)$ e $\{p_1, \dots, p_d\} \subset \mathbb{P}_L$ una base ortonormale di \mathbb{P}_L (rispetto a una funzione peso w), cioè tale che

$$\langle p_i, p_j \rangle = \int_{\Omega} p_i p_j d\omega = \delta_{ij}, \quad 1 \leq i, j \leq d,$$

dove δ_{ij} è il delta di Kronecker.

L'iperinterpolazione di $f \in C(\Omega)$ in \mathbb{P}_L consiste in una discretizzazione della proiezione ortogonale rispetto al prodotto interno (1.1) su \mathbb{P}_L .

Più precisamente

Definizione 2.1 (Iperinterpolazione filtrata). *Data una formula di quadratura a pesi positivi (1.2) con grado di precisione $2L$, l'operatore di iperinterpolazione $\mathcal{L}_L : C(\Omega) \rightarrow \mathbb{P}_L(\Omega)$ è definito da*

$$\mathcal{L}_L f := \sum_{\ell=1}^d \langle f, p_{\ell} \rangle_N p_{\ell}.$$

dove $\{p_1, \dots, p_d\} \subset \mathbb{P}_L$ è una base ortonormale di \mathbb{P}_L (rispetto a una funzione peso w).

È importante notare che la base polinomiale risulta ortonormale rispetto a un prodotto scalare che coinvolge una certa funzione peso w che non è necessariamente $w \equiv 1$.

Inoltre, essendo $p_i p_j$ un polinomio di grado al più $2L$, visto che la formula di quadratura ha grado di precisione pari a $2L$, si ottiene

$$\langle p_i, p_j \rangle_N = \langle p_i, p_j \rangle = \delta_{ij}, \quad 1 \leq i, j \leq d. \quad (2.1)$$

Nell'articolo [8], l'autore ha provato i seguenti teoremi.

Teorema 2.1. *Se una formula di quadratura è esatta per tutti i polinomi di grado al massimo $2L$, allora il numero N di punti di quadratura deve soddisfare la condizione $N \geq \dim(\mathbb{P}_L) = d$.*

L'interesse di questo teorema è che offre una condizione necessaria sul numero di nodi che deve avere la formula di quadratura perché possa essere utilizzata per produrre una iperinterpolante.

Si ricorda che una formula di quadratura a N punti che sia esatta per tutti i polinomi di grado $\leq 2L$, si dice *minima* se $N = d$. Si ha il seguente risultato. Si mostra che in tal caso i pesi sono necessariamente espressi da

$$w_k = 1 / \sum_{j=1}^d p_j(\mathbf{x}_k).$$

Teorema 2.2. *La classica formula di interpolazione*

$$\mathcal{L}_L f(\mathbf{x}_j) = f(\mathbf{x}_j), \quad j = 1, \dots, N$$

vale per un'arbitraria $f \in C(\Omega)$ se e solo se la formula di quadratura a pesi positivi che definisce $\langle \cdot, \cdot \rangle_N$, avente nodi $\{\mathbf{x}_i\}_{i=1, \dots, N}$, è minima.

Il teorema 2.2 è particolarmente significativo, perché fornisce una condizione necessaria e sufficiente per stabilire se l'approssimante $\mathcal{L}_L f$ interpola o meno una funzione f nei nodi $\{\mathbf{x}_i\}_{i=1, \dots, N}$.

Per esempio, se $\Omega \subset \mathbb{R}$ è un intervallo chiuso e limitato, si ha che $d = \dim(\mathbb{P}_L(\Omega)) = L+1$ e quindi l'approssimante costruita con una qualsiasi formula di quadratura gaussiana avente $N = d = L+1$ nodi (non necessariamente col peso di Legendre!) è anche interpolante, poiché avente pesi positivi e grado di precisione $2N - 1 = 2(L+1) - 1 = 2L + 1 \geq 2L$ e dunque sono minime.

D'altra parte non esistono formule di questo tipo sulla sfera per grado generico (cf. [8, p.251]).

Infine vale il seguente teorema di proiezione:

Teorema 2.3. *Nelle ipotesi precedenti se $f \in \mathbb{P}_L$ allora $\mathcal{L}_L f = f$.*

Tale risultato non sarà vero nel caso della iperinterpolazione filtrata.

2.1 Analisi dell'errore

In questa sezione si vuole condurre un'analisi relativa alla stima dell'errore dell'iperinterpolazione in norma L_2 , sia nel caso di funzioni continue prive di rumore, sia nel caso di funzioni soggette a rumore.

Sono richiamate ora alcune definizioni di norme di funzioni che compariranno nel seguito. Per ogni funzione $g \in C(\Omega)$, la *norma uniforme* è definita da $\|g\|_\infty = \sup_{x \in \Omega} |g(x)|$, mentre per ogni $g \in L_2(\Omega)$, la *norma L_2* è definita come $\|g\|_2 = (\int_\Omega |g|^2 d\omega)^{1/2}$. Inoltre, data $g \in C(\Omega)$ denoteremo con $p^* \in \mathbb{P}_L$ il *polinomio di miglior approssimazione* di g in \mathbb{P}_L , ovvero il polinomio tale per cui vale $\mathcal{E}_L(g) := \inf_{p \in \mathbb{P}_L} \|g - p\|_\infty = \|g - p^*\|_\infty$.

Dati non soggetti a rumore

Si consideri $f \in C(\Omega)$ e si ponga $y_j = f(\mathbf{x}_j)$.

Segue un lemma con alcune proprietà dell'iperinterpolazione.

Lemma 2.1. *Sia $f \in C(\Omega)$ e sia $\mathcal{L}_L f \in \mathbb{P}_L$ come nella definizione 2.1, dove i nodi di quadratura (tutti in Ω) e i pesi (positivi) nel prodotto interno discreto soddisfano la proprietà (1.3). Allora valgono le seguenti affermazioni:*

- (a) $\langle f - \mathcal{L}_L f, p \rangle_N = 0 \quad \forall p \in \mathbb{P}_L,$
- (b) $\langle \mathcal{L}_L f, \mathcal{L}_L f \rangle_N + \langle f - \mathcal{L}_L f, f - \mathcal{L}_L f \rangle_N = \langle f, f \rangle_N,$
- (c) $\langle \mathcal{L}_L f, \mathcal{L}_L f \rangle_N \leq \langle f, f \rangle_N,$
- (d) $\langle f - \mathcal{L}_L f, f - \mathcal{L}_L f \rangle_N = \min_{p \in \mathbb{P}_L} \langle f - p, f - p \rangle_N.$

Tale lemma risulta importante per dimostrare il seguente teorema di convergenza (cf. [8, p.244]).

Teorema 2.4. *Sotto le ipotesi del lemma 2.1 si ha*

$$\|\mathcal{L}_L f\|_2 \leq V^{1/2} \|f\|_\infty \quad (2.2)$$

e

$$\|\mathcal{L}_L f - f\|_2 \leq 2V^{1/2} \mathcal{E}_L(f). \quad (2.3)$$

Di conseguenza $\|\mathcal{L}_L f - f\|_2 \rightarrow 0$ per $L \rightarrow +\infty$.

Si osservi che il teorema stabilisce che nelle ipotesi richieste l'operatore $\mathcal{L}_L : C(\Omega) \rightarrow L^2(\Omega)$ ha norma limitata da $V = \int_\omega d\omega$, indipendentemente da L .

Dati soggetti a rumore

Si consideri ora una funzione $f \in C(\Omega)$ e $f^\epsilon \in C(\Omega)$ una sua perturbazione continua tale che, sull'insieme di nodi $\{\mathbf{x}_j\}$, $j = 1, \dots, N$, vale $f^\epsilon(\mathbf{x}_j) = f(\mathbf{x}_j) + \epsilon_j$, e si ponga $\boldsymbol{\epsilon} = [\epsilon_1, \dots, \epsilon_N]^T \in \mathbb{R}^N$.

Si enuncia ora un teorema con la stima dell'errore di $\mathcal{L}_L f^\epsilon$.

Teorema 2.5. *Sotto la ipotesi del Lemma 2.1 si ha*

$$\|\mathcal{L}_L f^\epsilon - f\|_2 \leq V^{1/2} \|f - f^\epsilon\|_\infty + 2V^{1/2} \mathcal{E}_L(f). \quad (2.4)$$

Per la dimostrazione si veda [1, Remark 4.7].

Un commento su questo teorema. Assumendo che $\boldsymbol{\epsilon}$ sia determinato, si ha $\|f - f^\epsilon\|_\infty = \|\boldsymbol{\epsilon}\|_\infty = \max_j |\epsilon_j|$ da cui segue che l'errore di approssimazione non si annulla per $L \rightarrow \infty$, come accade nel caso del Teorema 2.4, ma al contrario l'errore può essere più o meno significativo a seconda del rumore sui dati.

Tuttavia la perturbazione sui dati può essere in alcune situazioni casuale, quindi è necessario determinare l'errore di approssimazione in termini di valore atteso e a tale scopo occorre dare una stima a priori del rumore. Questo è possibile se si immagina di ripetere un esperimento più volte e poi di calcolare una media degli errori, oppure se si suppone di conoscere la distribuzione degli ϵ_j .

In tal caso vale $\mathbb{E}(\|f - f^\epsilon\|_\infty) = \mathbb{E}(\|\boldsymbol{\epsilon}\|_\infty)$, in cui il secondo termine dell'uguaglianza può essere stimato analiticamente.

Per esempio, nel caso di ϵ_j distribuito come una *variabile aleatoria sub-gaussiana*, dotata della *norma sub-gaussiana*

$$\|\epsilon_j\|_{\psi_2} := \inf\{T > 0 \mid \mathbb{E}(\exp(\epsilon_j^2/T^2)) \leq 2\},$$

allora si può dimostrare [1, Lemma 4.8] che vale

$$\mathbb{E}(\|\epsilon\|_\infty) \leq cG\sqrt{\log N},$$

dove $G = \max_j \|\epsilon_j\|_{\psi_2}$, ψ_p è la funzione che esprime il p-momento di una variabile aleatoria, cioè $\psi_p(x) = x^p$ e c è una generica costante.

In entrambi i casi, l'errore di approssimazione dipende in maniera lineare dall'errore sui dati.

Capitolo 3

Iperinterpolazione filtrata

Un possibile problema dell'iperinterpolazione, come si è visto nel capitolo 2, è che l'approssimante può essere poco preciso in presenza di rumore, vale a dire quando si lavora con dati relativi alla funzione nei nodi perturbati rispetto ai valori reali.

Un metodo per risolvere questo limite è ricorrere alla iperinterpolazione filtrata.

Definizione 3.1 (Filtro). *Una funzione continua $h : \mathbb{R}^+ \rightarrow \mathbb{R}$ si chiama filtro se soddisfa*

$$h(x) = \begin{cases} 1 & \text{se } x \in [0, \frac{1}{2}], \\ 0 & \text{se } x \in [1, +\infty). \end{cases}$$

Ovviamente, di filtri ce ne possono essere di molti tipi, a seconda del comportamento nell'intervallo $(1/2, 1)$, di natura più o meno regolare.

Di seguito si supporrà che si abbia $\text{supp}(h) = [0, a]$, con $a < 1$, dove al solito con *supp* si indica il supporto di una funzione.

Inoltre, si supporrà che la base polinomiale $\{p_k\}_{k=1, \dots, d}$ di \mathbb{P}_L sia di tipo *triangolare*, intendendo con ciò che per ogni $K \leq L$, posto $d_K = \dim(\mathbb{P}_K)$ si abbia $\mathbb{P}_K = \text{span}(p_1, \dots, p_{d_K})$.

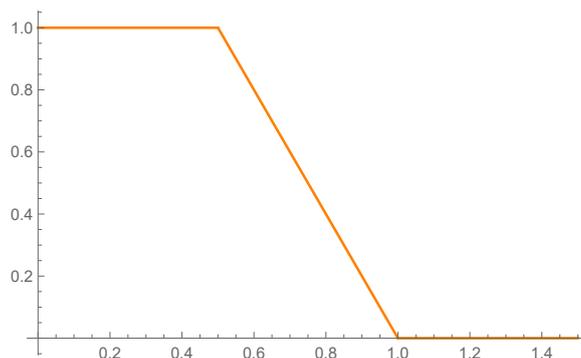
Verranno indicate nel seguito con $\lceil \cdot \rceil$ e $\lfloor \cdot \rfloor$ le parti intere di un numero reale rispettivamente dall'alto e dal basso.

Seguono ora alcuni esempi di possibili filtri presi da [10, p.101] e scalati nell'intervallo $[0, 1]$. Ogni filtro è seguito dal relativo grafico, realizzato con Mathematica.

Si ponga $y(x) := x - \frac{1}{2}$, con $y : (\frac{1}{2}, 1) \rightarrow (0, 1)$.

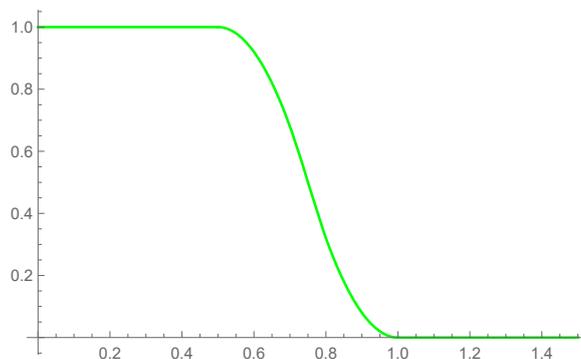
1. *Lineare*

$$h(x) = \begin{cases} 1 & \text{se } x \in [0, \frac{1}{2}], \\ 1 - 2y(x) & \text{se } x \in (\frac{1}{2}, 1), \\ 0 & \text{se } x \in [1, \infty). \end{cases}$$



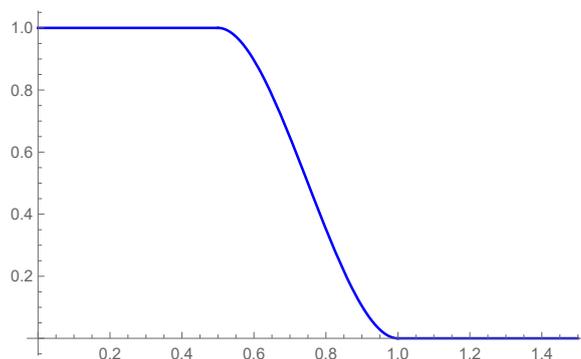
2. Quadratica a tratti.

$$h(x) = \begin{cases} 1 & \text{se } x \in [0, \frac{1}{2}], \\ 1 - 8y^2(x) & \text{se } x \in (\frac{1}{2}, \frac{3}{4}], \\ 8(y(x) - \frac{1}{2})^2 & \text{se } x \in (\frac{3}{4}, 1), \\ 0 & \text{se } x \in [1, \infty). \end{cases}$$



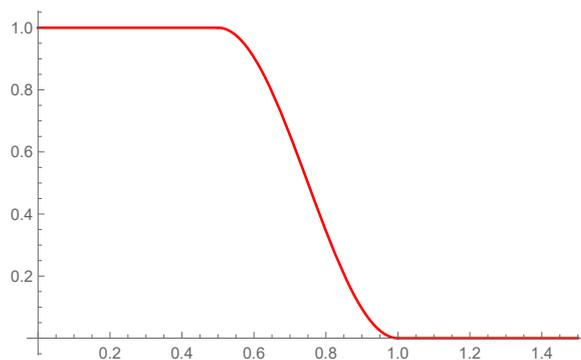
3. Cubica

$$h(x) = \begin{cases} 1 & \text{se } x \in [0, \frac{1}{2}], \\ 1 - 12y^2(x) + 16y^3(x) & \text{se } x \in (\frac{1}{2}, 1), \\ 0 & \text{se } x \in [1, \infty). \end{cases}$$



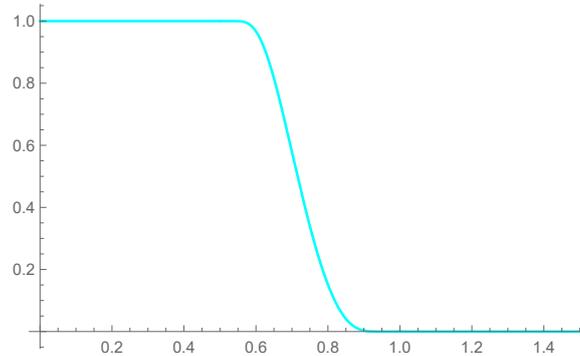
4. Polinomio trigonometrico

$$h(x) = \begin{cases} 1 & \text{se } x \in [0, \frac{1}{2}], \\ \sin^2(\pi(\frac{1}{2} + y(x))) & \text{se } x \in (\frac{1}{2}, 1), \\ 0 & \text{se } x \in [1, \infty). \end{cases}$$

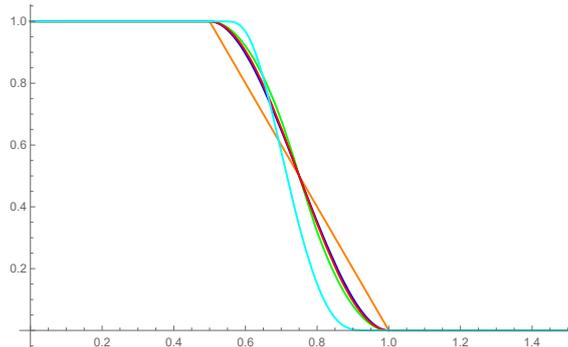


5. Esponenziale

$$h(x) = \begin{cases} 1 & \text{se } x \in [0, \frac{1}{2}], \\ \exp\left(\frac{-2\exp(-\frac{1}{2y(x)})}{\frac{1}{2}-y(x)}\right) & \text{se } x \in (\frac{1}{2}, 1), \\ 0 & \text{se } x \in [1, \infty). \end{cases}$$



Vengono rappresentati ora tutti i filtri su un stesso grafico in modo da poterli confrontare più facilmente



Definizione 3.2 (Iperinterpolazione filtrata). Sia h un filtro avente supporto $\text{supp}(h) = [0, a]$, con $a < 1$. Data una formula di quadratura algebrica avente N nodi in Ω , pesi positivi e grado di precisione $[aL] - 1 + [L/2]$, si definisce la iperinterpolazione filtrata $\mathcal{F}_L f$ di $f \in C(\Omega)$ come

$$\mathcal{F}_L f := \sum_{\ell=1}^d h\left(\frac{\deg(p_\ell)}{L}\right) \langle f, p_\ell \rangle_N p_\ell$$

dove $d = \dim(\mathbb{P}_L(\Omega))$.

Si nota subito che $\mathcal{F}_L f$ è un polinomio di grado totale al più $[aL] - 1$. Infatti se $\deg(p_\ell) \geq aL$ allora $h(\deg(p_\ell)/L) = 0$.

Si osserva inoltre che se $f \in \mathbb{P}_{[L/2]}$ allora $\mathcal{F}_L f = f$. Infatti, essendo la base $\{p_k\}_{k=1, \dots, d}$ triangolare ed ortonormale, posto $d_{[L/2]} = \dim(\mathbb{P}_{[L/2]})$, si ha per

$$f = \sum_{\ell=1}^{d_{[L/2]}} \langle f, p_\ell \rangle p_\ell,$$

essendo il grado di precisione almeno $\lfloor L/2 \rfloor + \lceil aL \rceil - 1$, ed $h(x) = 1$ per $x \in [0, 1/2]$,

$$\begin{aligned}
\mathcal{F}_L f &:= \sum_{\ell=1}^d h\left(\frac{\deg(p_\ell)}{L}\right) \langle f, p_\ell \rangle_N p_\ell \\
&= \sum_{\ell=1}^{\lfloor L/2 \rfloor} h\left(\frac{\deg(p_\ell)}{L}\right) \langle f, p_\ell \rangle_N p_\ell + \sum_{\ell=\lfloor L/2 \rfloor+1}^d h\left(\frac{\deg(p_\ell)}{L}\right) \langle f, p_\ell \rangle_N p_\ell \\
&= \sum_{\ell=1}^{\lfloor L/2 \rfloor} \langle f, p_\ell \rangle_N p_\ell + \sum_{\ell=\lfloor L/2 \rfloor+1}^d h\left(\frac{\deg(p_\ell)}{L}\right) \langle f, p_\ell \rangle_N p_\ell \\
&= \sum_{\ell=1}^{\lfloor L/2 \rfloor} \langle f, p_\ell \rangle p_\ell = f.
\end{aligned}$$

Si osservi che nella penultima uguaglianza si sfrutta il fatto che se la chiusura del supporto di h coincide con $[0, a]$, allora $h\left(\frac{\deg(p_\ell)}{L}\right) = 0$ per $\ell \geq d_{\lceil aL \rceil-1} + 1$, dove $d_{\lceil aL \rceil-1} = \dim(\mathbb{P}_{\lceil aL \rceil-1})$ e in virtù del fatto che il grado di precisione è almeno $\lfloor L/2 \rfloor + \lceil aL \rceil - 1$, allora $\langle f, p_\ell \rangle_N = \langle f, p_\ell \rangle = 0$, quindi si ricava

$$\begin{aligned}
\sum_{\ell=\lfloor L/2 \rfloor+1}^d h\left(\frac{\deg(p_\ell)}{L}\right) \langle f, p_\ell \rangle_N p_\ell &= \sum_{\ell=\lfloor L/2 \rfloor+1}^{d_{\lceil aL \rceil-1}} h\left(\frac{\deg(p_\ell)}{L}\right) \langle f, p_\ell \rangle_N p_\ell \\
&\quad + \sum_{\ell=d_{\lceil aL \rceil-1}+1}^d h\left(\frac{\deg(p_\ell)}{L}\right) \langle f, p_\ell \rangle_N p_\ell \\
&= \sum_{\ell=\lfloor L/2 \rfloor+1}^{d_{\lceil aL \rceil-1}} h\left(\frac{\deg(p_\ell)}{L}\right) \langle f, p_\ell \rangle p_\ell \\
&= 0.
\end{aligned}$$

3.1 Analisi dell'errore

In questa sezione si intende dare risultati per la iperinterpolazione filtrata che siano analoghi a quelli visti nella sezione 2.1.

Tali risultati sono presentati nell'articolo [1, pp. 9-11] applicati però ad una forma diversa di iperinterpolazione, detta lasso. Qui sono stati riportati nel caso di cui si sta trattando in questa tesi.

La notazione usata nel seguito è la stessa adottata in precedenza.

Dati non soggetti a rumore

Come primo passo viene enunciato un lemma con le proprietà della iperinterpolazione filtrata.

Lemma 3.1. *Sia $\Omega \subset \mathbb{R}^n$ un insieme compatto e $f \in C(\Omega)$. Sia h un filtro per cui la chiusura del suo supporto sia $[0, a]$ con $a \in (1/2, 1]$. Introdotto*

$$\mathcal{F}_L f = \sum_{\ell=1}^d h\left(\frac{\deg(p_\ell)}{L}\right) \alpha_\ell p_\ell, \quad \alpha_\ell = \langle f, p_\ell \rangle_N,$$

come da Definizione 3.2 e assumendo che la formula di quadratura utilizzata per definire \mathcal{F}_L abbia grado di precisione $2(\lceil aL \rceil - 1)$ si ha

- (a) $\langle f - \mathcal{F}_L f, \mathcal{F}_L f \rangle_N = K(f)$,
- (b) $\langle \mathcal{F}_L f, \mathcal{L}_L f \rangle_N + \langle f - \mathcal{F}_L f, f - \mathcal{F}_L f \rangle_N = \langle f, f \rangle_N - 2K(f)$,
tale $K(f)$ soddisfa $K(f) \leq \langle f, f \rangle_N / 2$,
- (c) $\langle \mathcal{F}_L f, \mathcal{F}_L f \rangle_N \leq \langle f, f \rangle_N - 2K(f)$,

dove

$$K(f) = \sum_{\ell=1}^d \left(h\left(\frac{\deg(p_\ell)}{L}\right) - h^2\left(\frac{\deg(p_\ell)}{L}\right) \right) \alpha_\ell^2.$$

e posto $d^- = \dim(\mathbb{P}_{\lceil aL \rceil - 1})$, vale

$$\sum_{\ell=1}^d \left(h\left(\frac{\deg(p_\ell)}{L}\right) - h^2\left(\frac{\deg(p_\ell)}{L}\right) \right) \alpha_\ell^2 = \sum_{\ell=1}^{d^-} \left(h\left(\frac{\deg(p_\ell)}{L}\right) - h^2\left(\frac{\deg(p_\ell)}{L}\right) \right) \alpha_\ell^2.$$

Dimostrazione.

L'ultima uguaglianza dell'enunciato è di immediata verifica e segue dal fatto che $h\left(\frac{\deg(p_\ell)}{L}\right) = 0$ per $\ell \geq d^-$.

Si possono quindi dimostrare gli altri tre risultati del teorema.

(a) La formula segue dal fatto che si ha

$$\begin{aligned} \langle f - \mathcal{F}_L f, \mathcal{F}_L f \rangle_N &= \sum_{k=1}^d h\left(\frac{\deg(p_k)}{L}\right) \alpha_k \left\langle f - \sum_{\ell=1}^d h\left(\frac{\deg(p_\ell)}{L}\right) \alpha_\ell p_\ell, p_k \right\rangle_N \\ &= \sum_{k=1}^{d^-} h\left(\frac{\deg(p_k)}{L}\right) \alpha_k \left\langle f - \sum_{\ell=1}^{d^-} h\left(\frac{\deg(p_\ell)}{L}\right) \alpha_\ell p_\ell, p_k \right\rangle_N \end{aligned}$$

e

$$\begin{aligned} \left\langle f - \sum_{\ell=1}^{d^-} h\left(\frac{\deg(p_\ell)}{L}\right) \alpha_\ell p_\ell, p_k \right\rangle_N &= \langle f, p_k \rangle_N - \left\langle \sum_{\ell=1}^{d^-} h\left(\frac{\deg(p_\ell)}{L}\right) \alpha_\ell p_\ell, p_k \right\rangle_N \\ &= \alpha_k - h\left(\frac{\deg(p_k)}{L}\right) \alpha_k. \end{aligned}$$

Nel caso la chiusura del supporto di h sia $[0, a]$, non è difficile vedere che per ottenere tale risultato basta supporre che la formula di quadratura che descrive il prodotto scalare necessita grado di precisione almeno $2(\lceil aL \rceil - 1)$.

(b) Segue da (a) che

$$\langle \mathcal{F}_L f, \mathcal{F}_L f \rangle_N = \langle f, \mathcal{F}_L f \rangle_N - K(f)$$

e inoltre vale

$$\langle f - \mathcal{F}_L f, f - \mathcal{F}_L f \rangle_N = \langle f, f \rangle_N - 2\langle f, \mathcal{F}_L f \rangle_N + \langle \mathcal{F}_L f, \mathcal{F}_L f \rangle_N.$$

Sommando membro a membro le due relazioni si ricava l'uguaglianza dell'enunciato, visto che

$$\begin{aligned} \langle \mathcal{F}_L f, \mathcal{F}_L f \rangle_N + \langle f - \mathcal{F}_L f, f - \mathcal{F}_L f \rangle_N &= \\ &= \langle f, \mathcal{F}_L f \rangle_N - K(f) + \langle f, f \rangle_N - 2\langle f, \mathcal{F}_L f \rangle_N + \langle \mathcal{F}_L f, \mathcal{F}_L f \rangle_N \\ &= \langle f, f \rangle_N - K(f) - \langle f, \mathcal{F}_L f \rangle_N + \langle \mathcal{F}_L f, \mathcal{F}_L f \rangle_N \\ &= \langle f, f \rangle_N - K(f) - \langle f - \mathcal{F}_L f, \mathcal{F}_L f \rangle_N \\ &= \langle f, f \rangle_N - 2K(f) \end{aligned}$$

Osservando poi che $\langle g, g \rangle_N \geq 0$ per ogni $g \in C(\Omega)$, segue che $\langle f, f \rangle_N - 2K(f) \geq 0$, da cui si ottiene la limitazione su $K(f)$ cercata.

(c) La formula segue direttamente da (b) e dal fatto che $\langle f - \mathcal{F}_L f, f - \mathcal{F}_L f \rangle_N$ è una quantità positiva. \square

Sulla base del lemma appena dimostrato, si può enunciare il seguente teorema.

Teorema 3.1. *Sia un filtro $h : \mathbb{R}^+ \rightarrow [0, 1]$, avente supporto $\text{supp}(h) = [0, a]$ e si ponga $L^* := \lceil aL \rceil - 1$. Si supponga inoltre che la formula di quadratura utilizzata per il calcolo di $\mathcal{F}_L f$ abbia grado di precisione almeno $2L^*$. Allora esiste $\tau_1 \leq 1$, inversamente proporzionale con $K(f)$, tale che*

$$\|\mathcal{F}_L f\|_2 \leq \tau_1 V^{1/2} \|f\|_\infty, \quad (3.1)$$

ed esiste $\tau_2 \leq 1$, inversamente proporzionale con $K(f - p^*)$, tale che

$$\|\mathcal{F}_L f - f\|_2 \leq (1 + \tau_2) V^{1/2} \mathcal{E}_{L^*}(f) + \|p^* - \mathcal{F}_L p^*\|_2 \quad (3.2)$$

dove p^* è il polinomio di miglior approssimazione di f in \mathbb{P}_L .

Dimostrazione.

Per dimostrare la prima relazione si ha

$$\begin{aligned} \|\mathcal{F}_L f\|_2^2 &= \langle \mathcal{F}_L f, \mathcal{F}_L f \rangle = \langle \mathcal{F}_L f, \mathcal{F}_L f \rangle_N \leq \langle f, f \rangle_N - 2K(f) \\ &= \sum_{j=1}^N w_j f(\mathbf{x}_j)^2 - 2K(f) \leq \sum_{j=1}^N w_j \|f\|_\infty^2 - 2K(f) \\ &= V \|f\|_\infty^2 - 2K(f), \end{aligned}$$

dove nella prima uguaglianza è stato usato il fatto che in virtù del grado di esattezza della formula si ha $\langle \mathcal{F}_L f, \mathcal{F}_L f \rangle = \langle \mathcal{F}_L f, \mathcal{F}_L f \rangle_N$, mentre la disuguaglianza successiva segue dal punto (c) del Lemma 3.1. Per concludere, sfruttando l'ipotesi che l'immagine di h è

contenuta in $[0, 1]$, si ha che $K(f) \geq 0$, quindi esiste $\tau_1 \leq 1$ inversamente proporzionale con $K(f)$, tale che

$$\sqrt{V\|f\|_\infty^2 - 2K(f)} = \tau_1 V^{1/2} \|f\|_\infty.$$

Per dimostrare la seconda relazione si osserva che per ogni polinomio $p \in \mathbb{P}_L$, dalla linearità di \mathcal{F}_L ,

$$\begin{aligned} \|\mathcal{F}_L f - f\|_2 &= \|\mathcal{F}_L(f - p) - (f - p) - (p - \mathcal{F}_L p)\|_2 \\ &\leq \|\mathcal{F}_L(f - p)\|_2 + \|f - p\|_2 + \|p - \mathcal{F}_L p\|_2. \end{aligned}$$

Quindi poiché la scelta di $p \in \mathbb{P}_L$ è arbitraria, si può porre in particolare $p = p^*$, polinomio di miglior approssimazione di f in \mathbb{P}_L (relativamente alla norma 2). Allora esiste $\tau_2 < 1$ inversamente proporzionale con $K(f - p^*)$ tale che

$$\|\mathcal{F}_L f - f\|_2 \leq \tau_2 V^{1/2} \|f - p^*\|_\infty + V^{1/2} \|f - p^*\|_\infty + \|p^* - \mathcal{F}_L p^*\|_2,$$

dove il primo termine del membro di destra segue da (3.1), mentre il secondo termine segue dalla disuguaglianza di Cauchy-Schwarz per cui vale

$$\|f - p^*\|_2 = \sqrt{\langle f - p^*, f - p^* \rangle} \leq \|f - p^*\|_\infty \sqrt{\langle 1, 1 \rangle} = V^{1/2} \|f - p^*\|_\infty$$

. Quindi si ha la stima cercata. \square

Osservazione.

Si osservi che nelle ipotesi del teorema, il grado di precisione richiesto è almeno $2(\lceil aL \rceil - 1)$, con $a > 1/2$, che in generale è maggiore di quello richiesto dalla definizione di iperinterpolazione filtrata, ovvero $\lceil aL \rceil - 1 + \lfloor L/2 \rfloor$.

Osservazione.

Osservando sempre le ipotesi del teorema, se ne trova una riguardante una proprietà di h . Tale assunzione tuttavia non è restrittiva e verrà chiarita meglio nella sezione 4.2

Osservazione.

Confrontando questo risultato con quello del Teorema 2.4 si osserva che mentre il primo termine a secondo membro è almeno favorevole come quello ottenuto per il caso dell'iperinterpolazione classica, si introduce un ulteriore termine nella stima dell'errore rispetto a (2.3), che si può chiamare *errore di regolarizzazione*.

Dati soggetti a rumore

Per stimare l'errore di approssimazione nel caso di funzioni continue soggette a rumore, si ha a disposizione il seguente teorema

Teorema 3.2. *Si assumano le ipotesi del Teorema 3.1 e si considerino $f \in C(\Omega)$ e $f^\epsilon \in C(\Omega)$ una sua perturbazione continua tale per cui, sull'insieme di nodi $\{\mathbf{x}_j\}$, $j = 1, \dots, N$, vale $f^\epsilon(\mathbf{x}_j) = f(\mathbf{x}_j) + \epsilon_j$. Allora esiste $\tau_3 \leq 1$ inversamente proporzionale con $K(f^\epsilon - p^*)$, tale che*

$$\|\mathcal{F}_L f^\epsilon - f\|_2 \leq \tau_3 V^{1/2} \|f - f^\epsilon\|_\infty + (1 + \tau_3) V^{1/2} \mathcal{E}_L(f) + \|p^* - \mathcal{F}_L p^*\|_2, \quad (3.3)$$

dove p^* è il polinomio di miglior approssimazione di f in \mathbb{P}_L .

Dimostrazione.

Dal Teorema 3.1 si ha

$$\|\mathcal{F}_L f^\epsilon - f\|_2 \leq \tau_3 V^{1/2} \|f^\epsilon - p^*\|_\infty + V^{1/2} \|f - p^*\|_\infty + \|p^* - \mathcal{F}_L p^*\|_2,$$

dove $\tau_3 < 1$ è inversamente proporzionale con $K(f^\epsilon - p^*)$. Per concludere la dimostrazione si utilizza la stima $\|f^\epsilon - p^*\|_\infty \leq \|f^\epsilon - f\|_\infty + \|f - p^*\|_\infty$, da cui segue la disuguaglianza dell'enunciato. \square

Osservazione.

Si osserva che rispetto a (2.4), in (3.3) è minore la stima dell'errore dato dal rumore, ma compare il termine dovuto all'errore di regolarizzazione. Tuttavia, quando la funzione da approssimare è particolarmente soggetta a rumore, l'errore di regolarizzazione fornisce un contributo trascurabile alla stima dell'errore totale e quindi è preferibile ricorrere alla iperinterpolazione filtrata come metodo di approssimazione.

Capitolo 4

Problemi di approssimazione ai minimi quadrati

Per comprendere meglio il motivo per cui uno dei due metodi appena descritti approssimi in maniera migliore funzioni continue a seconda del fatto che i dati siano soggetti o meno a rumore (in termini di errore in norma L_2), si considera in questo capitolo la relazione tra iperinterpolazione filtrata e problema dei minimi quadrati regolarizzati (cf. [1, pp. 4-6]).

Si ricorda la notazione usata finora: dato $n \geq 1$, sia $\Omega \subset \mathbb{R}^n$ dominio compatto. $f \in C(\Omega)$ è la funzione che si intende approssimare nei dati $\{(\mathbf{x}_j, f(\mathbf{x}_j))\}_{j=1}^N$, $\{\mathbf{x}_j\}_{j=1}^N \subset \Omega$, $N \geq d$ (affinché il problema sia ben posto). \mathbb{P}_L è lo spazio dei polinomi di grado al più L , con $\dim(\mathbb{P}_L) = d$ e $\{p_1, \dots, p_d\}$ è una base di tale spazio. $\mathcal{L}_L f$ è l'iperinterpolante di f su \mathbb{P}_L (Definizione 2.1) e $\mathcal{F}_L f$ è la iperinterpolante filtrata di f su \mathbb{P}_{L-1} (Definizione 3.2).

4.1 Problema dei minimi quadrati "standard"

Si definisce problema di approssimazione ai *minimi quadrati* pesati

$$\min_{p \in \mathbb{P}_L} \left\{ \sum_{j=1}^N w_j (p(\mathbf{x}_j) - f(\mathbf{x}_j))^2 \right\} \quad (4.1)$$

dove $p(\mathbf{x}) = \sum_{\ell=1}^d \alpha_\ell p_\ell(\mathbf{x}) \in \mathbb{P}_L$ è il polinomio di grado L con cui si sceglie di approssimare f .

Sia ora $\mathbf{A} \in \mathbb{R}^{N \times d}$ la matrice di *Vandermonde* i cui elementi sono $[\mathbf{A}]_{ij} = p_i(\mathbf{x}_j)$, $j = 1, \dots, N$, $\ell = 1, \dots, d$, ($\mathbf{x}_j \in \mathbb{R}^n$), e sia \mathbf{W} una matrice diagonale in cui $W_{j,j} = w_j$, $j = 1, \dots, n$. Allora il problema di approssimazione (4.1) può essere riscritto nella forma equivalente

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^d} \|\mathbf{W}^{1/2}(\mathbf{A}\boldsymbol{\alpha} - \mathbf{f})\|_2^2 \quad (4.2)$$

dove $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_d]^T \in \mathbb{R}^d$ è il vettore con i coefficienti utilizzati per costruire p e $\mathbf{f} = [f(\mathbf{x}_1), \dots, f(\mathbf{x}_N)]^T \in \mathbb{R}^N$ è il vettore con i valori della funzione nei nodi di campionamento.

Osservando che il problema (4.2) è descritto da una funzione nella variabile α strettamente convessa, segue che il punto stazionario è soluzione unica. Si può calcolare quindi il gradiente rispetto ad α per ottenere la condizione (*equazioni normali*)

$$\mathbf{A}^T \mathbf{W} \mathbf{A} \boldsymbol{\alpha} - \mathbf{A}^T \mathbf{W} \mathbf{f} = 0. \quad (4.3)$$

Si ha dunque il seguente risultato.

Teorema 4.1. *Sotto le ipotesi del Lemma 2.1 si ha che $\mathcal{L}_L f$ è la soluzione unica del problema (4.1).*

Dimostrazione.

La dimostrazione si basa sulla condizione (4.3). Da un lato la matrice $\mathbf{A}^T \mathbf{W} \mathbf{A}$ è la matrice identica poiché essendo la base $\{p_k\}_{k=1,\dots,d}$ ortonormale rispetto al prodotto scalare di $L_2(\Omega)$ e il grado di precisione della formula di cubatura pari a $2L$

$$[\mathbf{A}^T \mathbf{W} \mathbf{A}]_{ik} = \sum_{j=1}^N w_j p_i(\mathbf{x}_j) p_k(\mathbf{x}_j) = \langle p_i, p_k \rangle_N = \delta_{ik}, \quad 1 \leq i, k \leq d,$$

D'altra parte si ha

$$[\mathbf{A}^T \mathbf{W} \mathbf{f}]_\ell = \sum_{j=1}^N w_j p_\ell(\mathbf{x}_j) f(\mathbf{x}_j) = \langle p_\ell, f \rangle_N, \quad \ell = 1, \dots, d.$$

Quindi la soluzione unica del problema (4.1) è il polinomio costruito con i coefficienti $\alpha_\ell = \langle p_\ell, f \rangle_N$, ovvero $\mathcal{L}_L f$. \square

Da questo importante teorema si può concludere, come già anticipato che in assenza di rumore l'iperinterpolazione approssima meglio in norma $L_2(\Omega)$ una funzione $f \in C(\Omega)$ nei dati $\{(\mathbf{x}_j, f(\mathbf{x}_j))\}_{j=1}^N$, in quanto minimizza l'errore in tale norma.

4.2 Problema dei minimi quadrati regolarizzati

Si definisce problema di approssimazione ai *minimi quadrati regolarizzati* pesati

$$\min_{p \in \mathbb{P}_L} \left\{ \sum_{j=1}^N w_j (p(\mathbf{x}_j) - f(\mathbf{x}_j))^2 + \sum_{j=1}^N w_j (\mathcal{R}_L p(\mathbf{x}_j))^2 \right\}, \quad (4.4)$$

dove $p(\mathbf{x}) = \sum_{\ell=1}^d \beta_\ell p_\ell(\mathbf{x}) \in \mathbb{P}_L$ è il polinomio di grado L con cui si sceglie di approssimare f , $\mathcal{R}_L : \mathbb{P}_L \rightarrow \mathbb{P}_L$ lineare è l'*operatore di regolarizzazione*, definito dalla mappa $p_k \mapsto \mu_k p_k$, dove i μ_k sono arbitrari.

Si noti che è stata scelta una diversa notazione per i coefficienti del polinomio approssimante rispetto al problema (4.1), per distinguerli da quelli dell'iperinterpolazione.

Il significato di costruire un problema di questo tipo è trovare un approssimante che minimizzi l'errore in norma L_2 quando si lavora con funzioni soggette a rumore, tramite un'opportuna scelta dell'operatore \mathcal{R}_L .

Con un procedimento analogo a quello utilizzato per passare da (4.1) a (4.2), si può riscrivere il problema (4.4)

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^d} \|\mathbf{W}^{1/2}(\mathbf{A}\boldsymbol{\beta} - \mathbf{f})\|_2^2 + \|\mathbf{W}^{1/2}\mathbf{R}\mathbf{A}\boldsymbol{\beta}\|_2^2, \quad (4.5)$$

dove $\mathbf{R} \in \mathbb{R}^{d \times d}$ è una matrice diagonale con entrate $\{\mu_\ell\}_{\ell=1}^d$ e $\boldsymbol{\beta} = [\beta_1, \dots, \beta_d]^T \in \mathbb{R}^d$.

Il punto stazionario di questo problema quadratico verifica la condizione

$$\mathbf{A}^T \mathbf{W} \mathbf{A} \boldsymbol{\beta} - \mathbf{A}^T \mathbf{W} \mathbf{f} + \mathbf{A}^T \mathbf{R}^T \mathbf{W} \mathbf{R} \mathbf{A} \boldsymbol{\beta} = 0. \quad (4.6)$$

Ora si osserva, similmente a quanto visto nel paragrafo (4.1), che

$$\begin{aligned} [\mathbf{A}^T \mathbf{R}^T \mathbf{W} \mathbf{R} \mathbf{A}]_{ik} &= \sum_{j=1}^N \mu_i^2 w_j p_i(\mathbf{x}_j) p_k(\mathbf{x}_j) \\ &= \mu_i^2 \langle p_i, p_j \rangle_N \\ &= \mu_i^2 \delta_{ik}, \quad 1 \leq i, k \leq d, \end{aligned} \quad (4.7)$$

Quindi, per (4.7) e per quanto provato nella dimostrazione al Teorema 2.1, segue che la condizione (4.6) è equivalente a

$$\beta_\ell - \alpha_\ell + \mu_\ell^2 \beta_\ell = 0, \quad (4.8)$$

da cui si ottiene che la soluzione al problema (4.4) è il polinomio costruito con i coefficienti $\beta_\ell = \frac{1}{1+\mu_\ell^2} \alpha_\ell$.

Si può osservare che tale soluzione coincide con l'iperinterpolante classica se $\mu_\ell = 0$, $\ell = 1, \dots, d$, vale a dire se non si introduce l'operatore di regolarizzazione e si considera un problema analogo a (4.1).

Si osserva inoltre che scegliendo $\mu_\ell = 0$ per $\ell = 1, \dots, \lfloor L/2 \rfloor$, l'operatore di regolarizzazione coincide con un particolare filtro e allo stesso modo, ogni filtro che abbia immagine in $[0, 1]$ coincide con un particolare operatore di regolarizzazione scelti opportunamente i μ_ℓ , tali che

$$h\left(\frac{\deg(p_\ell)}{L}\right) = \frac{1}{1 + \mu_\ell^2}, \quad \deg(p_\ell) < \frac{L}{2}, \quad (4.9)$$

Osservazione.

Dalla formula (4.9) segue che, ogni filtro costruito come operatore di regolarizzazione garantisce sia verificata l'ipotesi su h del Teorema 3.1.

Capitolo 5

Esempi di approssimazione su domini multivariati

In questo capitolo si vuole studiare la filtered hyperinterpolation applicata a vari domini quali l'intervallo, il cerchio unitario, la sfera e alcune regioni particolari su quest'ultima come i triangoli sferici. Per ciascun esempio si vuole dare una possibile formula di quadratura così che si possa determinare in forma esplicita una iperinterpolante filtrata (a meno della scelta del filtro) (cf. [1], [11]).

5.1 L'intervallo

Si consideri $\Omega = [-1, 1]$ e quale funzione peso quella di Gauss-Chebyshev $w(x) = 1/\sqrt{1-x^2}$ osservando comunque che in generale si può scegliere una classica funzione peso di tipo Gauss-Jacobi. Conseguentemente, nelle stime di errore sarà $V = \int_{-1}^1 w(x)dx = \pi$. In questo caso \mathbb{P}_L è lo spazio dei polinomi univariati di grado al più L su $[-1, 1]$ e quindi $\dim(\mathbb{P}_L) = L + 1$. Si assuma $\{p_\ell\}_{\ell=0, \dots, L}$ una base ortonormale di \mathbb{P}_L rispetto alla funzione peso w . Per $N \geq L + 1$ si consideri per esempio la formula gaussiana di quadratura, che si ricorda viene costruita scegliendo come nodi $\{x_j\}_{j=1, \dots, N}$ gli zeri del polinomio ortogonale p_N e come pesi i $\{w_j\}_{j=1, \dots, N}$ tali che $w_j = \int_{-1}^1 \ell_j(x) \frac{1}{2} dx$, dove $\ell_i(x)$ è il coefficiente dell' i -esimo polinomio di Lagrange. In questo modo, per via dell'esattezza delle formule gaussiane relativamente a tale funzione peso, in cui

$$x_i = \cos\left(\frac{2i-1}{2N}\pi\right), \quad w_i = \frac{\pi}{N}, \quad i = 1, \dots, N,$$

si ha

$$\sum_{j=1}^N w_j g(x_j) = \int_{-1}^1 \frac{1}{2} g(x) d\omega \quad \forall g \in \mathbb{P}_{2N-1}.$$

Quindi l'iperinterpolante filtrata, relativa alla formula di quadratura citata, risulta

$$\mathcal{F}_L f := \sum_{\ell=0}^L h\left(\frac{\ell}{L}\right) \left(\sum_{j=1}^N w_j f(x_j) p_\ell(x_j) \right) p_\ell$$

5.2 Il disco unitario

Si ponga ora $\Omega = \{\mathbf{x} \in \mathbb{R}^2 : \mathbf{x} = (x_1, x_2), \text{ t.c. } x_1^2 + x_2^2 \leq 1\}$, che è il disco unitario su \mathbb{R}^2 , con $d\omega = (1/\pi)dx$. In questo caso vale

$$V = \int_{\Omega} d\omega = \frac{1}{\pi} \int_0^1 \int_0^{2\pi} 1r d\theta dr = 1.$$

Si dimostra che $\mathbb{P}_L := \left\{ \sum_{j=0}^L \sum_{k=0}^j a_{jk} x_1^k x_2^{j-k} : a_{jk} \in \mathbb{R} \right\}$, ovvero lo spazio dei polinomi di grado al più L su Ω , ha $\dim(\mathbb{P}_L) = \binom{L+2}{2} = (L+2)(L+1)/2$. Una tipica base ortonormale di \mathbb{P}_L è la famiglia di *ridge polynomials* $\{\Lambda_\ell\}_{\ell=1, \dots, (L+2)(L+1)/2}$, detti di *Logan-Shepp* [6]. Un'alternativa molto apprezzata consiste nei *polinomi di Zernike* [5]. A questo punto, scrivendo in coordinate polari $\mathbf{x} = (\rho, \theta)$, dove ρ e θ sono rispettivamente il raggio e la direzione azimutale di \mathbf{x} , il prodotto interno discreto (1.4) può essere riscritto come segue

$$\begin{aligned} \langle u, v \rangle_N &:= \frac{1}{\pi} \sum_{j=1}^N \sum_{m=0}^{2N} u\left(\rho_j, \frac{2\pi m}{2N+1}\right) v\left(\rho_j, \frac{2\pi m}{2N+1}\right) w_j \frac{2\pi}{2N+1} \rho_j \\ &= \sum_{j=1}^N \sum_{m=0}^{2N} u\left(\rho_j, \frac{2\pi m}{2N+1}\right) v\left(\rho_j, \frac{2\pi m}{2N+1}\right) w_j \frac{2}{2N+1} \rho_j, \end{aligned} \quad (5.1)$$

dove è stato usato il metodo dei trapezi per approssimare l'integrale rispetto alla direzione azimutale e la formula di quadratura di Gauss-Legendre per l'integrale rispetto alla direzione radiale, con nodi e pesi $\{\rho_j\}_{j=0, \dots, N}$ e $\{w_j\}_{j=0, \dots, N}$ rispettivamente. Il prodotto interno (5.1) così definito è esatto per ogni coppia di polinomi in \mathbb{P}_N , vale a dire

$$\begin{aligned} \sum_{j=1}^N \sum_{m=0}^{2N} g\left(\rho_j, \frac{2\pi m}{2N+1}\right) w_j \frac{2}{2N+1} \rho_j &= \\ = \frac{1}{\pi} \int_0^1 \int_0^{2\pi} g(\rho, \theta) \rho d\theta d\rho &\quad \forall g \in \mathbb{P}_{2N}. \end{aligned}$$

Quindi la relativa iperinterpolante filtrata si scrive

$$\mathcal{F}_L f := \sum_{\ell=0}^L h\left(\frac{\ell}{L}\right) \left(\sum_{j=0}^N \sum_{m=0}^{2N} f\left(\rho_j, \frac{2\pi m}{2N+1}\right) \Lambda_\ell\left(\rho_j, \frac{2\pi m}{2N+1}\right) w_j \frac{2}{2N+1} \rho_j \right) \Lambda_\ell.$$

5.3 La sfera unitaria

Sia $\Omega = \mathbb{S}^2 = \{\mathbf{x} \in \mathbb{R}^3 : \mathbf{x} = (x_1, x_2, x_3), \text{ t.c. } x_1^2 + x_2^2 + x_3^2 = 1\}$, che è la sfera unitaria su \mathbb{R}^3 , con $d\omega = \omega(x)dx$, dove $\omega(x)$ è una misura di area su \mathbb{S}^2 . Si ha quindi che l'area della superficie di \mathbb{S}^2 è

$$V = \int_{\mathbb{S}^2} \omega(x) dx = 4\pi.$$

In questo caso $\mathbb{P}_L(\mathbb{S}^2)$ è lo spazio dei polinomi sferici di grado al più L . Si scelga come base di tale spazio un insieme ortonormale di armoniche sferiche (cf. [7]) $\{Y_{\ell, k} : \ell = 0, \dots, L; k = 1, \dots, 2\ell + 1\}$, da cui si deduce che la dimensione di $\mathbb{P}_L(\mathbb{S}^2)$ è $\dim(\mathbb{P}_L(\mathbb{S}^2)) = d = (L+1)^2$.

Anche in questo caso si può ricorrere a un sistema di coordinate polari tali per cui si scrive $\Omega = \{\mathbf{x} \in \mathbb{R}^2 : \mathbf{x} = (x, \Phi), 0 \leq x \leq \pi, 0 \leq \Phi \leq 2\pi\}$. Quindi servono due formule di quadratura per calcolare gli integrale rispetto alle due coordinate.

A tal proposito, si ricorda che ne esistono di molti tipi, come ad esempio gli *spherical t-designs*. Molte di queste formule sono descritte in [4].

Per comprendere l'applicazione all'iperinterpolazione filtrata ci si riferisce a una semplice formula, descritta in [9] e spesso utilizzata, che però soffre del fatto di accumulare i punti ai poli, proprietà ritenuta negativa.

Per calcolare l'integrale rispetto a Φ si consideri la regola dei rettangoli

$$\frac{\pi}{N+1} \sum_{k=0}^{2N+1} g\left(\frac{k\pi}{N+1}\right) = \int_0^{2\pi} g(\Phi) d\Phi$$

che è esatta per tutti i polinomi trigonometrici di grado $\leq 2N+1$.

Per calcolare invece l'integrale rispetto a x , si scelga per esempio la regola di Clenshaw-Curtis [2] avente nodi $\{x_j = \cos\left(\frac{j\pi}{2N}\right)\}_{j=0,\dots,2N}$ e pesi

$$w_0 = w_{2N} = \frac{1}{n} \sum_{k=0}^N \text{"} \frac{1}{1-4k^2},$$

$$w_j = w_{2N-j} = \frac{2}{n} \sum_{k=0}^N \text{"} \frac{1}{1-4k^2} \cos\left(\frac{kj\pi}{n}\right), \quad j = 1, \dots, N,$$

dove l'apice " sulle sommatorie indica che il primo e l'ultimo termine vanno dimezzati.

In questo caso una iperinterpolante filtrata diventa

$$\mathcal{F}_L f := \sum_{\ell=0}^L \sum_{m=0}^{2N+1} h\left(\frac{\ell}{L}\right) \alpha_{\ell,m} Y_{\ell,m}.$$

dove

$$\alpha_{\ell,m} = \sum_{j=0}^{2N} f\left(\cos\left(\frac{j\pi}{2N}\right), \frac{m\pi}{N+1}\right) Y_{\ell,m}\left(\cos\left(\frac{j\pi}{2N}\right), \frac{m\pi}{N+1}\right) w_j \frac{\pi}{2N+1}.$$

5.4 Triangoli sferici

Si consideri in questa sezione come insieme Ω , un triangolo sferico $\mathcal{T} = \widehat{abc} \subset \mathbb{S}^2$, $d\omega = \omega(x)dx$ una misura di superficie su \mathbb{S}^2 e $\mathbb{P}_L(\mathcal{T})$ lo spazio dei polinomi sferici ristretti al triangolo sferico \mathcal{T} , avente vertici \mathbf{a} , \mathbf{b} , \mathbf{c} .

Si ricorda che \mathcal{T} è la porzione di sfera delimitata da 3 *grandi cerchi*.

Senza perdita di generalità, si può considerare, a meno di rotazioni, un triangolo sferico con centro $(\mathbf{a} + \mathbf{b} + \mathbf{c}) / \|\mathbf{a} + \mathbf{b} + \mathbf{c}\|_2$ nel polo nord e si richiede che tale triangolo sia interamente contenuto nell'emisfero nord.

Nell'ambito dell'iperinterpolazione, risulta fondamentale determinare una formula di quadratura con pesi positivi e grado di esattezza M su \mathcal{T} . Un possibile metodo è descritto in [11]. Di seguito viene richiamato solo il procedimento di base.

Per iniziare, si proietta il triangolo sferico sul piano XY , ottenendo un *triangolo ellittico* che può essere suddiviso in tre settori ellittici (congiungendo i vertici con l'origine) S_1, S_2, S_3 (si veda Figura 5.1).

Data quindi una funzione $f \in C(\mathcal{T})$, si può scrivere

$$\begin{aligned} \int_{\mathcal{T}} f(x, y, z) d\omega &= \int_{\mathbb{T}} f(x, y, g(x, y)) \frac{1}{g(x, y)} dx dy \\ &= \sum_{i=1}^3 \int_{S_i} f(x, y, g(x, y)) \frac{1}{g(x, y)} dx dy, \end{aligned} \quad (5.2)$$

dove $g(x, y) = \sqrt{1 - x^2 - y^2}$ e \mathbb{T} è la proiezione di \mathcal{T} sul piano equatoriale.

Si vuole una formula che risulti esatta per $f(x, y, g(x, y)) = x^\alpha y^\beta g(x, y)^{\gamma-1}$, con $0 \leq \alpha + \beta + \gamma \leq M$, cioè quando f è un polinomio sferico di grado al più M .

A tal proposito, una strategia è di approssimare la funzione $1/g$ alla precisione di macchina con un polinomio, problema che è a sua volta risolvibile determinando il grado m per cui esista un polinomio univariato $p_m \in \mathbb{P}_m([0, r])$ con $r = \max\{\|\mathbf{a}\|_2^2 + \|\mathbf{b}\|_2^2 + \|\mathbf{c}\|_2^2\}$ che approssimi circa alla precisione di macchina funzione $1/\sqrt{1-t}$ in $[0, r]$. In questo modo infatti, si può considerare il polinomio di miglior approssimazione di $1/g$ e quindi calcolare l'ultimo integrale in (5.2) con un'opportuna formula di quadratura su ogni singolo settore, il cui grado dipende da M e m (si veda [11] per dettagli).

Il calcolo dei singoli integrali sui settori circolari $S_i, i = 1, 2, 3$, è dato da una applicazione delle formule di quadratura sub-gaussiane (cf. [3]). Supponendo siano $\{x_j^{(i)}, y_j^{(i)}\}_j$ e $\{w_j^{(i)}\}_j$ rispettivamente i nodi e i pesi positivi di tali formule su $S_i, i = 1, 2, 3$, la formula di quadratura su \mathcal{T} avrà nell'ordine come nodi $\cup_{i=1}^3 \{x_j^{(i)}, y_j^{(i)}, g(x_j^{(i)}, y_j^{(i)})\}_j$ e corrispettivi pesi $\cup_{i=1}^3 \{w_j^{(i)}/g(x_j^{(i)}, y_j^{(i)})\}_j$.

Si osserva che $f(x, y, g(x, y))/g(x, y)$ non è sempre un polinomio rispetto alle variabili (x, y) (questa situazione si verifica solo se γ è dispari), quindi una formula di quadratura utilizzata per approssimare l'ultimo integrale in (5.2) sarà quasi esatta in $\mathbb{P}_L(\mathcal{T})$, nel senso che l'errore rimarrà vicino alla precisione di macchina.

Osservazione.

L'implementazione del metodo appena descritto consente di essere applicato a contesti più generali.

Infatti si osserva che nel caso in cui il triangolo sferico non garantisca le condizioni iniziali, questo può essere suddiviso in tre triangoli sferici più piccoli (a loro volta suddivisibili), che soddisfino le ipotesi richieste.

Più in generale, un qualsiasi poligono sferico con n lati, può essere scomposto in n triangoli sferici sui quali è possibile costruire formule aventi il grado di precisione richiesto.

Non viene scritta in questo caso una forma esplicita dell'iperinterpolante filtrata per via della notazione che diventerebbe troppo impegnativa.

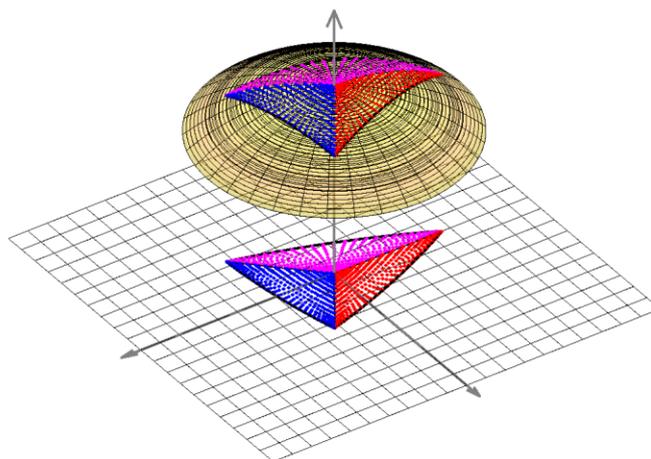


Figura 5.1: Nodi di quadratura su un triangolo ellittico ruotato con centro al polo nord, sollevato rispetto al triangolo ellittico proiettato sul piano XY

Capitolo 6

Risultati numerici

In questo ultimo capitolo si intende verificare operativamente, mediante alcuni esperimenti numerici, quanto discusso finora.

Nell'articolo [1], sono già stati condotti esperimenti simili su intervallo, disco e sfera, di seguito l'attenzione verterà sui triangoli sferici.

Ciò che ci si aspetta di osservare è che l'iperinterpolazione classica sia preferibile rispetto alla sua variante filtrata quando si intende approssimare funzioni prive di rumore, ma non nel caso in cui i dati siano soggetti a perturbazioni.

Nello specifico, ciò che si vuole verificare è il teorema di proiezione per iperinterpolazione classica e filtrata e poi testare la performance dei due metodi per funzioni di *natura* esponenziale su un generico triangolo sferico.

Prima di riportare i dati, si richiamano alcuni passaggi chiave dell'algoritmo utilizzato.

Come visto nei capitoli precedenti, un punto fondamentale dell'iperinterpolazione consiste nell'avere a disposizione una base ortonormale (anche triangolare nel caso di iperinterpolazione filtrata), tuttavia una conoscenza esplicita di tale base non è garantita su un generico triangolo sferico, quindi va costruita numericamente.

Questo punto rende il dominio in questione, ovvero un triangolo sferico, diverso da quelli analizzati in [1], in quanto su intervalli, dischi, sfere, cubi, sono note basi ortonormali.

Per risolvere questo problema si costruisce la base cercata a partire da una base ortonormale sulla sfera unitaria, ad esempio l'insieme delle armoniche sferiche, mediante un procedimento di fattorizzazione QR. Tale procedura viene eseguita dalla routine `dHYPERFIT2`, che calcola anche i prodotti scalari tra gli elementi della base, vale a dire i coefficienti dell'iperinterpolante.

A questo punto l'algoritmo proposto permette di decidere diversi elementi per costruire un esperimento, quali:

1. la funzione da approssimare, scegliendo tra polinomio di grado n o $\lfloor n/2 \rfloor$ e funzione esponenziale;
2. la presenza o meno di rumore, che può essere di tipo impulsivo (in questo caso è possibile scegliere l'intervallo entro il quale il rumore assume valori casuali) o di tipo gaussiano (di cui si può scegliere la varianza); il grado di approssimazione (negli esperimenti riportati di seguito va da 1 a 10);
3. il grado algebrico di precisione della formula, che nei test effettuati è $2n$ per $n = 1, \dots, 10$.

Una volta assegnati questi elementi, l'algoritmo ripete 100 volte (numero di test modificabile) nelle stesse condizioni l'esperimento, variando in maniera casuale nell'intervallo $(0, 1)$ i coefficienti della funzione scelta e calcolando poi una media degli errori commessi, come differenza in norma 2 tra funzione esatta f e approssimante della funzione (anche soggetta a rumore). In particolare vengono calcolati approssimativamente sia l'errore in norma infinito, sia in norma 2, quest'ultimo rispetto ai pesi della formula di cubatura di grado 30 calcolata sul triangolo sferico dalla routine `cub_sphtri`.

Per quanto riguarda il filtro utilizzato negli esperimenti, è stato scelto il polinomio trigonometrico descritto nel capitolo 3, ma può essere cambiato nella routine `hyperfilter`.

Si ricorda, infine, che ripetendo tali esperimenti nelle medesime condizioni, i valori osservati potrebbero risultare diversi. Questo fenomeno è dovuto in primo luogo al fatto che negli algoritmi proposti, più volte sono stati introdotti processi casuali che rendono variabili i risultati.

Oltre a questo, a fare la differenza potrebbe essere la versione di Matlab utilizzata e la macchina su cui è installato il programma, in questo caso rispettivamente *'Matlab R2021b'* e Laptop *'HP'* con processore *'Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80 GHz'* e RAM da 12,0 GB.

Si procede ora con la descrizione dei dati osservati (nelle tabelle che seguiranno, i simboli $\|\cdot\|_\infty$ e $\|\cdot\|_2$, indicano l'errore di approssimazione calcolato rispettivamente rispetto alla norma infinito e alla norma due).

Sono riportate di seguito le immagini dei due domini utilizzati negli esperimenti.

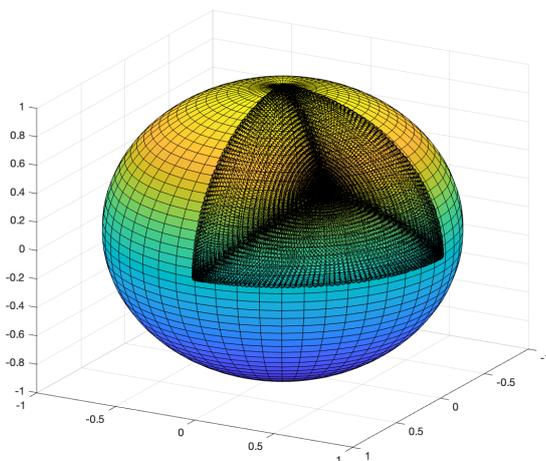


Figura 6.1: Triangolo sferico \widehat{ABC} di vertici $A = (1, 0, 0)$, $B = (0, 1, 0)$, $C = (0, 0, 1)$.

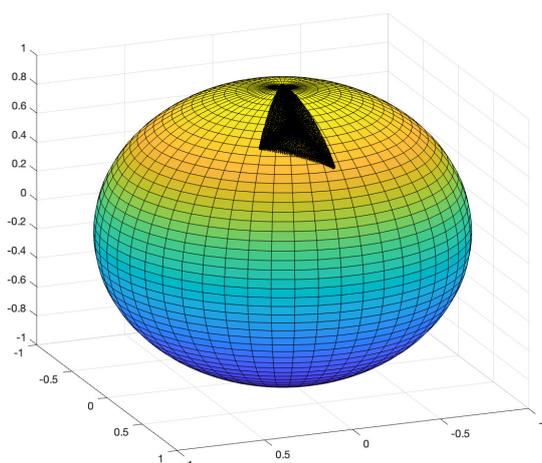


Figura 6.2: Triangolo sferico $\widehat{A'B'C'}$ di vertici $A' = (0, 0, 1)$, $B' = (0, \sqrt{a}, \sqrt{1-a})$, $C' = (\sqrt{b}, \sqrt{a/2}, \sqrt{1-a/2-b})$, dove $a = 0.5$, $b = 0.1$.

Scopo del primo esperimento è verificare il teorema di proiezione (T2.3). Come dominio è stato scelto il triangolo sferico di vertici $A = (1, 0, 0)$, $B = (0, 1, 0)$, $C = (0, 0, 1)$ (figura 6.1) e la funzione da approssimare, in assenza di rumore, viene fatta variare all'interno della famiglia di polinomi di grado n tramite il comando

```
1 nexp=n;
  c0=rand(1); c1=rand(1); c2=rand(1); c3=rand(1);
3 g=@(x,y,z) (c0+c1*x+c2*y+c3*z).^nexp;
```

	IPERINTERPOLAZIONE		IPERINTERPOLAZIONE FILTRATA	
Grado	$\ \cdot\ _\infty$	$\ \cdot\ _2$	$\ \cdot\ _\infty$	$\ \cdot\ _2$
1	2.7323e-14	5.741e-16	0.51428	0.18306
2	4.3063e-14	7.3336e-16	0.19553	0.040127
3	1.2787e-13	1.9358e-15	0.11326	0.027941
4	1.4792e-13	2.6172e-15	0.059775	0.007813
5	3.9307e-13	6.6384e-15	0.043868	0.013025
6	8.1881e-13	3.0017e-14	0.081724	0.0077351
7	2.1105e-13	5.8508e-15	0.0060982	0.0023438
8	2.0839e-12	9.7684e-14	0.037861	0.006115
9	1.6038e-13	1.8331e-14	0.0028953	0.00086126
10	4.0711e-12	1.5323e-13	0.011523	0.001565

Come previsto, i dati mostrano che l'iperinterpolazione classica risulta esatta per i polinomi di grado al più n , mentre ciò non risulta vero nel caso dell'iperinterpolazione filtrata. Si nota un certo degradamento dell'errore in norma al crescere del grado pur rimanendo particolarmente piccolo.

Con il prossimo esperimento, si vuole mostrare lo stesso risultato precedente, nel caso dell'iperinterpolazione filtrata. Come sopra, il dominio scelto è il triangolo \widehat{ABC} , i dati

non sono soggetti a rumore, ma in questo caso, la funzione da approssimare appartiene alla famiglia dei polinomi di grado $\lfloor n/2 \rfloor$ ed è definita dal seguente comando

```
1 nexp=floor(n/2);
  c0=rand(1); c1=rand(1); c2=rand(1); c3=rand(1);
3 g=@(x,y,z) (c0+c1*x+c2*y+c3*z).^nexp;
```

	IPERINTERPOLAZIONE		IPERINTERPOLAZIONE FILTRATA	
Grado	$\ \cdot\ _\infty$	$\ \cdot\ _2$	$\ \cdot\ _\infty$	$\ \cdot\ _2$
1	6.6613e-16	4.3824e-16	3.3307e-16	4.1744e-16
2	3.002e-14	4.6993e-16	2.9965e-14	4.4185e-16
3	3.6471e-14	8.9825e-16	3.6871e-14	8.7664e-16
4	5.9419e-14	1.0302e-15	5.8281e-14	1.0042e-15
5	6.1711e-14	1.1971e-15	6.269e-14	1.0849e-15
6	1.0294e-13	1.8812e-15	1.142e-13	1.6999e-15
7	8.6148e-14	1.6283e-15	8.9193e-14	1.5491e-15
8	1.1167e-13	2.2173e-15	1.2178e-13	2.0735e-15
9	1.4896e-13	4.2587e-15	1.7203e-13	4.0502e-15
10	1.4551e-13	3.5539e-15	1.3579e-13	3.4518e-15

Anche in questo caso gli esperimenti confermano quanto descritto dalla teoria relativamente all'esattezza dell'iperinterpolazione filtrata per polinomi di grado al più $\lfloor n/2 \rfloor$.

Nel prossimo esperimento, continuando a lavorare in assenza di rumore, si prende in considerazione una funzione appartenente a una famiglia di funzioni esponenziali definita dal seguente comando

```
1 c0=rand(1); c1=rand(1); c2=rand(1); c3=rand(1);
  g=@(x,y,z) exp(-(c0*x.^2+c1*y.^2+c2*z.^2+c3));
```

Il dominio considerato è sempre il triangolo sferico \widehat{ABC} .

	IPERINTERPOLAZIONE		IPERINTERPOLAZIONE FILTRATA	
Grado	$\ \cdot\ _\infty$	$\ \cdot\ _2$	$\ \cdot\ _\infty$	$\ \cdot\ _2$
1	0.028936	0.011521	0.11487	0.072246
2	0.0052997	0.0015745	0.030422	0.013532
3	0.00036378	9.3856e-05	0.0063678	0.0023178
4	8.3232e-05	2.0685e-05	0.0021506	0.00069395
5	8.6461e-06	1.9234e-06	0.0005218	0.00016861
6	7.7749e-07	1.4836e-07	0.00012831	3.2575e-05
7	1.9468e-08	3.3707e-09	1.5977e-05	3.9334e-06
8	2.1328e-08	3.6325e-09	1.7897e-05	4.395e-06
9	1.5124e-09	2.3406e-10	3.5393e-06	8.196e-07
10	1.3399e-10	1.9103e-11	1.12e-06	2.2648e-07

Osservando i dati si può concludere, in accordo con la teoria, che l'iperinterpolazione classica fornisce una migliore approssimazione rispetto alla versione filtrata in assenza di rumo-

re. Tuttavia si osserva anche che già al grado 10, l'errore in norma 2 dell'iperinterpolazione filtrata è dell'ordine di 10^{-7} , quindi si tratta ugualmente di una buona approssimazione.

A questo punto si introduce del rumore. Nello specifico i dati sono soggetti ad una perturbazione casuale la cui funzione di distribuzione coincide con quella di una variabile aleatoria gaussiana con media $\mu = 0$ e varianza σ^2 , con $\sigma = 0.5$. Il comando che introduce tale rumore è il seguente

```
pert_gauss=0;
2 if sigma > 0
  var=sigma^2;
4 pert_gauss=sqrt(var)*randn(size(g_pts));
  while norm(pert_gauss) == 0
6 pert_gauss=sqrt(var)*randn(size(g_pts));
  end
8 end
```

Dominio e funzione da approssimare scelti sono gli stessi del caso precedente.

	IPERINTERPOLAZIONE		IPERINTERPOLAZIONE FILTRATA	
Grado	$\ \cdot\ _\infty$	$\ \cdot\ _2$	$\ \cdot\ _\infty$	$\ \cdot\ _2$
1	0.087509	0.037355	0.13335	0.081599
2	0.11693	0.04387	0.067896	0.03371
3	0.21896	0.059758	0.09347	0.036807
4	0.32367	0.074879	0.16455	0.049334
5	0.43479	0.086656	0.20344	0.05679
6	0.50888	0.097285	0.23893	0.070067
7	0.49688	0.099363	0.24572	0.067702
8	0.57186	0.10601	0.25996	0.072976
9	0.77238	0.1159	0.3667	0.080386
10	0.7635	0.12073	0.3145	0.081019

Da questi valori si osserva chiaramente che l'iperinterpolazione filtrata risulta migliore della versione classica per approssimare funzioni soggette a rumore, in particolare risulta sperimentalmente che l'errore in norma 2 viene ridotto circa di un fattore 1.5.

L'esperimento successivo ripropone la stessa famiglia di funzioni esponenziali, soggette al rumore gaussiano del precedente esperimento. In questo caso però, il dominio scelto è il triangolo sferico di vertici $A' = (0, 0, 1)$, $B' = (0, \sqrt{a}, \sqrt{1-a})$, $C' = (\sqrt{b}, \sqrt{a/2}, \sqrt{1-a/2-b})$, dove $a = 0.5$, $b = 0.1$ (figura 6.2).

	IPERINTERPOLAZIONE		IPERINTERPOLAZIONE FILTRATA	
Grado	$\ \cdot\ _\infty$	$\ \cdot\ _2$	$\ \cdot\ _\infty$	$\ \cdot\ _2$
1	0.1066	0.015195	0.043034	0.0091038
2	0.29734	0.026795	0.12415	0.017106
3	0.44523	0.03105	0.18203	0.020612
4	0.4942	0.038818	0.27215	0.027608
5	0.68768	0.039235	0.30729	0.025942
6	0.72552	0.044215	0.36305	0.028147
7	0.96914	0.044461	0.40316	0.029552
8	0.85032	0.043523	0.40474	0.029586
9	0.8472	0.050661	0.42357	0.034742
10	1.0235	0.051211	0.43651	0.034507

Il risultato che si osserva è che la scelta del dominio interviene nella stima dell'errore in norma 2, per via del termine $V^{1/2}$ che compare nei Teoremi T2.4, T2.5, T3.1, T3.2. In particolare, in questo secondo caso, l'errore risulta inferiore rispetto al precedente in quanto l'area del triangolo sferico $\widehat{A'B'C'}$ è minore dell'area di \widehat{ABC} .

Non cambia invece il fatto che l'iperinterpolazione filtrata risulta più efficace della versione senza filtro e il fattore di miglioramento è sempre circa 1.5.

Come ulteriore esperimento, si propone di mantenere fisse la famiglia di funzioni da approssimare, cioè quella esponenziale, il grado di approssimazione, ovvero 10 e far variare invece la varianza σ del rumore gaussiano. Il dominio di riferimento è il triangolo sferico $\widehat{A'B'C'}$.

	IPERINTERPOLAZIONE		IPERINTERPOLAZIONE FILTRATA	
σ	$\ \cdot\ _\infty$	$\ \cdot\ _2$	$\ \cdot\ _\infty$	$\ \cdot\ _2$
0.1	0.17358	0.010234	0.10298	0.0069425
0.2	0.39213	0.020851	0.19131	0.014375
0.3	0.58235	0.031577	0.30883	0.020796
0.4	0.87456	0.040887	0.40569	0.026798
0.5	1.0235	0.051211	0.43651	0.034507

Come ci si poteva aspettare, gli errori assoluti sia in norma 2 che in norma infinito, crescono all'aumentare di σ , ma non cambia il rapporto tra l'errore dato dall'iperinterpolazione classica e da quella filtrata.

Quale ultimo esperimento si prende in considerazione la possibilità di introdurre sui dati un rumore di tipo impulsivo, vale a dire che la perturbazione assume valori casuali nell'intervallo $[-a, a]$ con probabilità 1/2. Si definisce tramite il comando

```

pert_impulse=0;
2 if a > 0
  pert_impulse=a*(1-2*rand(length(g_pts),1))*binornd(1,0.5);
4 while norm(pert_impulse) == 0
  pert_impulse=a*(1-2*rand(length(g_pts),1))*binornd(1,0.5);

```

6 end
end

In questo caso, si pone $a = 2.5$, il dominio è il triangolo sferico $\widehat{A'B'C'}$ e la famiglia di funzioni da approssimare è sempre quella esponenziale.

	IPERINTERPOLAZIONE		IPERINTERPOLAZIONE FILTRATA	
Grado	$\ \cdot\ _\infty$	$\ \cdot\ _2$	$\ \cdot\ _\infty$	$\ \cdot\ _2$
1	0.45252	0.058991	0.095811	0.025964
2	0.85433	0.067136	0.27314	0.034382
3	0.96448	0.082438	0.61785	0.05846
4	1.313	0.096167	0.67559	0.063903
5	1.6936	0.1143	0.87411	0.078531
6	2.0015	0.1147	0.92512	0.075805
7	2.6449	0.12882	1.1377	0.085494
8	2.4356	0.13305	1.246	0.092132
9	2.6947	0.14086	1.2723	0.094426
10	2.5547	0.15234	1.423	0.098423

I dati mostrano che l'errore in norma infinito risulta in questo caso elevato, per via del fatto che il rumore impulsivo può perturbare in maniera significativa la funzione su alcuni nodi. Ciò che conta maggiormente è osservare che l'errore in norma due assume ancora valori piccoli, in particolar modo per l'iperinterpolazione filtrata, confermando l'efficacia di approssimazione di tale metodo.

Appendice A

Codici

In questa sezione vengono riportate le principali routines utilizzate in questo lavoro, con i relativi commenti tradotti. Gli stessi codici, nella versione originale, insieme ad eventuali algoritmi di supporto, sono disponibili open-source al link: https://github.com/enricocaregnato/Codici_tesi.

Il primo codice "demo_sphtri" è quello principale che calcola iperinterpolante classica e filtrata con i relativi errori.

```
1 function demo_sphtri
  %-----
3 % Descrizione:
  % Demo per l'iperinterpolazione in un dominio fissato su
5 % triangoli sferici sulla sfera unitaria S2.
  % Calcola: iperinterpolazione, iperint. filtrata.
7 %
  % Autori: E. Caregnato e A. Sommariva
9 %
  % Data: 26 Nov 2022
11 %-----
  clear; clf;
13 domain_example=1;

15 % Grado di precisione in esperimenti numerici:
  % possibilmente definito come un vettore
17 nV=1:10;

19 % Parametri per il rumore.
  noise=0;           % 0: no rumore, 1: rumore
21 a=0.5;             % definisce il rumore impulsivo
  sigma=0.5; %0.02;  % definisce il rumore gaussiano
23 domain_structure.domain='spherical-triangle';
  %-----
25 % ..... main code .....
  % Impostazioni speciali
27
  % Approximation type parameter "pts_type".
29 % case 1, pts_type='Hyper su insieme completo';
```

```

% case 2, pts_type='Hyper su insieme compresso';
31 %
% Nota: il caso "2" dovrebbe essere utilizzato per
33 % piccoli valori di "n", al massimo 15.

35 pts_type=1;

37 % grafico di dominio e nodi:
% do_plot=1 (yes), do_plot=0 (no).
39
do_plot=0;
41
% Numero di test per ricostruire i polinomi.
43 ntests=100;

45 % funzioni test:
% 1. polinomio di grado n,
47 % 2. polinomio di grado floor(n/2)-1
% 3. esponenziale di tipo gaussiana
49 funct_example=2;

51 % Applicazione delle ipostazioni
% Dominio
53 vertices=define_domain(domain_example);

55 % Insieme dei punti
% [XYZWR, dbox]=define_cub_rule(domain_structure,30);
57 P1=vertices(1,:); P2=vertices(2,:); P3=vertices(3,:);
XYZWR = cub_sphtri(30,P1',P2',P3',pos);
59 XR=XYZWR(:,1:end-1); WR=XYZWR(:,end);

61 % Approssimazione numerica, variando il grado in "nV"
fprintf('\n \t ');
63 % vettori usati per le statistiche
AEinfMV=[]; AE2MV=[]; beta0MV=[];
65 for k=1:length(nV)
n=nV(k);
67 dimpoly=(n+1)^2;

69 % Estrazione dei nodi di iperinterpolazione
if pts_type == 2 % insieme compresso
71 XYZW = cub_sphtri(2*n,P1',P2',P3',pos);
[pts,weights,momerr,dbox] = ...
73 dCATCH(2*n,XYZW(:,1:3),XYZW(:,4));
else % insieme completo
75 XYZW = cub_sphtri(2*n,P1',P2',P3',pos);
pts=XYZW(:,1:end-1); weights=XYZW(:,end);
77 end

79 % Calcolo dell'errore dell'iperinterpolazione

```

```

    % per ogni "f" al grado "deg"
81 poly_coeffs=[];
    for j=1:ntests
83 switch funct_example

85 case 1 % testa l'esattezza dell'iperinterpolazione
    nexp=n;
87 c0=rand(1); c1=rand(1); c2=rand(1); c3=rand(1);
    g=@(x,y,z) (c0+c1*x+c2*y+c3*z).^nexp;
89
    case 2
91 % testa l'esattezza dell'iperinterpolazione filtrata
    nexp=ceil(n/2)-1;
93 c0=rand(1); c1=rand(1); c2=rand(1); c3=rand(1);
    g=@(x,y,z) (c0+c1*x+c2*y+c3*z).^nexp;
95
    case 3 % esponenziale
97 c0=rand(1); c1=rand(1); c2=rand(1); c3=rand(1);
    g=@(x,y,z) exp(-(c0*x.^2+c1*y.^2+c2*z.^2+c3));
99
    end
101
    % valutazione della funzione da approssimare
103 g_pts=feval(g,pts(:,1),pts(:,2),pts(:,3));

105 % aggiunta del rumore (se presente)
    if noise
107 % aggiunta del rumore impulsivo
        pert_impulse=0;
109 if a > 0
            pert_impulse=...
111 a*(1-2*rand(length(g_pts),1))*binornd(1,0.5);
            while norm(pert_impulse) == 0
113 pert_impulse=...
                a*(1-2*rand(length(g_pts),1))*binornd(1,0.5);
115 end
            end
117
            % aggiunta del rumore gaussiano
119 pert_gauss=0;
            if sigma > 0
121 var=sigma^2;
                pert_gauss=sqrt(var)*randn(size(g_pts));
123 while norm(pert_gauss) == 0
                    pert_gauss=sqrt(var)*randn(size(g_pts));
125 end
                end
127
            % somma dei rumori
129 pert=pert_impulse+pert_gauss;

```

```

131 % calcolo valori perturbati
    g_pts_pert=g_pts+pert;
133 else
    g_pts_pert=g_pts;
135 end

137 % calcolo del polinomio interpolante
    [coeff0,R,jvec,dbox,degs] = dHYPERFIT2(n,pts,weights,...
139 g_pts_pert,[],[],domain_structure,dimpoly);
    if iscell(jvec), degs=degs(jvec{1});
141 else, degs=degs(jvec);
    end
143
    % test dell'iperinterpolante con o senza filtri
145 for ktest=1:2
    switch ktest
147 case 1
        hypermode='hyperinterpolation';
149 parms.lambda=[]; parms.mu=[];
        coeff=coeff0;
151 case 2
        hypermode='filtered';
153 parms.lambda=[]; parms.mu=[];
        coeff=hyperfilter(hypermode,coeff0,degs,parms);
155 end

157 % valutazione dell'iperinter sull'insieme iniziale
    p_XR=...
159 dPOLYVAL2(n,coeff,XR,R,jvec,dbox,domain_structure,dimpoly);

161 % stima dell'errore
    g_XR=feval(g,XR(:,1),XR(:,2),XR(:,3));
163 tolAE=10^(-12)*ones(size(p_XR));
    % errore assoluto in norma infinito
165 AEinfV(ktest,j)=norm(g_XR-p_XR,inf);
    % errore assoluto in norma 2
167 AE2V(ktest,j)=sqrt(WR'*((g_XR-p_XR).^2));
    end
169 end

171 % medie degli errori (vettori 2 x 1)
    AEinfM=mean(AEinfV,2);
173 AE2M=mean(AE2V,2);
    beta0M=mean(beta0V,2);
175
    fprintf('\n.... table at degree: %2.0f .... \n \n ',n);
177 HypType=categorical({'hyperint.'; 'filtered'});
    T = table(HypType,AEinfM,AE2M,beta0M); disp(T)
179 AEinfMV=[AEinfMV AEinfM];

```

```

AE2MV=[AE2MV AE2M];
181
% parametri per le statistiche
183 card_pts(k)=length(weights);
card_polyspace(k)=(n+1)^2;
185 end

187 % ..... Grafico dei nodi .....
domain_parms=vertices;
189 plot_s2(domain_example, domain_parms, pts, [], '', []);

191 % ..... Statistiche generali .....

193 fprintf('\n \t .....');
fprintf('\n \t ** Domain: %2.0f'); disp(domain_example);
195 if pts_type == 1
fprintf('\n \t ** Hyper on full pointset \n \t');
197 else
fprintf('\n \t ** Hyper on compressed pointset \n \t');
199 end

201 fprintf('\n \t .....');
fprintf('\n \t | n | hypset | dimpoly | ');
203 fprintf('\n \t .....');
for k=1:length(nV)
205 fprintf('\n \t | %3.0f | %5.0f | %5.0f |', ...
nV(k), card_pts(k), card_polyspace(k));
207 end
fprintf('\n \t .....');
209 fprintf('\n \t * SPECIFIC PARAMETERS');
fprintf('\n \t .....');
211 fprintf('\n \t impulse noise parameter, a: %1.15g', a);
fprintf('\n \t gaussian noise par, sigma: %1.15g', sigma);
213 fprintf('\n \t averages on # tests: %3.0f', ntests);
fprintf('\n \t .....');
215 fprintf('\n \n');
fprintf('\n \n');
217 %-----
%-----

219 function vertices=define_domain(example)
switch example
221
case 0 % dominio esteso
223 vertices=[ 1 0 0; 0 1 0; 0 0 1];

225 case 1 % dominio medio-esteso
a=0.5; b=0.1;
227 vertices=[0 0 1; 0 sqrt(a) sqrt(1-a);
sqrt(b) sqrt(a/2) sqrt(1-a/2-b)];

```

Il prossimo algoritmo "cub_sphtri" viene richiamato dal precedente per calcolare una formula di cubatura di grado n sul triangolo sferico scelto come dominio.

```

function xyzw = cub_sphtri(n,P1,P2,P3,pos)
2  %-----
   % Descrizione:
4  % La seguente routine calcola una formula di cubatura
   % "xyzw", con grado di precisione quasi algebrico "n" su
6  % un triangolo sferico di vertici "P1","P2","P3",
   % mediante una formula 2D con grado di precisione "m+n"
8  % sulla proiezione nel piano XY del triangolo sferico,
   % precedentemente ruotato in modo da far coincidere il
10 % suo baricentro con il polo nord.
   %
12 % Il parametro "m" dipende dalle dimensioni del raggio
   % della circonferenza circoscritta al triangolo
14 % proiettato: vale "2" per 'piccoli' "r" e cresce
   % per "r" che tende a "1".
16 %
   % IMPORTANTE: Questa routine richiede Chebfun
18 %-----
   % Input:
20 % n: grado algebrico di precisione della formula;
   % P1,P2,P3: vettori colonna con le coordinate dei
22 % vertici del triangolo sferico
   % pos: pesi positivi per "pos=1",
24 % possibili pesi negativi (piu' veloce) per "pos=0"
   %-----
26 % Output:
   % xyzw : 4 vettori colonna con le coordinate cartesiane
28 % dei nodi e i pesi rispettivamente
   %-----
30 % Routines chiamate:
   % 1. compute_m (interna)
32 % 2. cub_circsect (esterna)
   % 3. dCATCH (esterna)
34 %-----
   % Data:
36 % Routine originale scritta da M. Vianello nel Maggio 2019.
   % Ultimo aggiornamento: 01/01/2021 da A. Sommariva.
38 %-----
   %% Copyright (C) 2021
40 %% Alwise Sommariva, Marco Vianello.
   %%
42 %% This program is free software; you can redistribute
   %% it and/or modify it under the terms of the GNU
44 %% General Public License as published by the Free
   %% Software Foundation; either version 2 of the License,
46 %% or (at your option) any later version.
   %%

```

```

48  %% This program is distributed in the hope that it will
    %% be useful, but WITHOUT ANY WARRANTY; without even the
50  %% implied warranty of MERCHANTABILITY or FITNESS FOR A
    %% PARTICULAR PURPOSE. See the GNU General Public
52  %% License for more details.
    %%
54  %% You should have received a copy of the GNU General
    %% Public License along with this program; if not, write
56  %% to the Free Software Foundation, Inc., 59 Temple
    %% Place, Suite 330, Boston, MA 02111-1307 USA
58  %%
    %% Autori: Alvisè Sommariva, Marco Vianello.
60  %%
    %% Data: January 01, 2021
62  %-----
    % ..... risoluzione problemi .....
64
    % lavorando con vertici come vettori colonna
66  if size(P1,1) == 1, P1=P1'; end
    if size(P2,1) == 1, P2=P2'; end
68  if size(P3,1) == 1, P3=P3'; end
    %-----
70  % ..... main code .....
    % Il problema viene scalato nella sfera unitaria con
72  % centro [0 0 0].
    % In altre parole, si costruisce una formula di cubatura
74  % sulla sfera unitaria e mappando sulla sfera iniziale
    % di raggio R la formula di cubatura così ottenuta.
76
    % ... baricentro ...
78  RR=norm(P1);
    vert0=[P1 P2 P3]/RR;
80  CC=1/3*(P1+P2+P3); CC=CC/norm(CC);

82  % ..... matrice di rotazione al polo nord .....
    [az,el,r] = cart2sph(CC(1),CC(2),CC(3));
84  phi=az; theta=pi/2-el;
    cp=cos(phi); sp=sin(phi); ct=cos(theta); st=sin(theta);
86  R1=[ct 0 -st; 0 1 0; st 0 ct];
    R2=[cp sp 0; -sp cp 0; 0 0 1];
88  R=R1*R2; invR=R';

90  % ..... vertici del triangolo al polo nord .....
    vert1=R*vert0;
92
    % calcolo di "m" per determinare il grado di precisione
94  m=compute_m(vert1');

96  % ..... determinazione della regola di quadratura .....
    xyzw=[]; % nodi sulla sfera

```

```

98  vert1=[vert1 vert1(:,1)];
    for i=1:3
100      % matrice di trasformazione affine
102  P=vert1(:,i); Q=vert1(:,i+1);
    om=acos(P'*Q);
104  xi=cos(om/2); eta=sin(om/2);
    M=[xi eta 0 0; 0 0 xi eta; xi -eta 0 0; 0 0 xi -eta];
106  h=[P(1); P(2); Q(1); Q(2)];
    u=M\h;
108  T=[u(1) u(2); u(3) u(4)];

110  % nodi e pesi per la proiezione su XY del triangolo
    % sferico ruotato
112  nw=cub_circsect(m+n,om/2,0,1); nod2=T*nw(:,1:2)';

114  % nodi e pesi per il triangolo sferico centrato nel
    % polo nord
116  x=nod2(1,:); y=nod2(2,:); z=sqrt(1-x.^2-y.^2);
    weights=abs(det(T))*nw(:,3)./z';
118
    % rotazione inversa per il triangolo sferico iniziale
120  nodes=invR*[x; y; z];

122  % formula di cubatura
    xyzw=[xyzw;[nodes' weights]];
124
    end
126  %-----
    % Estensione del risultato ad una sfera di raggio "R"
128  X=RR*xyzw(:,1);
    Y=RR*xyzw(:,2);
130  Z=RR*xyzw(:,3);
    W=RR*xyzw(:,4);
132  xyzw=[X Y Z W];
    %-----
134  %-----
    function m=compute_m(vert)
136  %-----
    % Descrizione:
138  % Calcola il numero intero "m", che dipende dal
    % triangolo sferico di vertici "vert", tale per cui una
140  % formula di cubatura di grado "n", sul triangolo
    % sferico, si ottiene mediante una formula di
142  % quadratura di grado "n+m" sulla proiezione del
    % triangolo sul piano XY.
144  %-----
    % Input:
146  % vert: sono le coordinate dei punti che descrivono i
    % vertici del triangolo sferico (il k-esimo punto viene

```

```

148 % descritto dalla k-esima riga).
%-----
150 % Output:
% m: numero intero positivo che dipende dal
152 % triangolo sferico di vertici "vert", tale per cui una
% formula di cubatura di grado "n", sul triangolo
154 % sferico, si ottiene mediante una formula di
% quadratura di grado "n+m" sulla proiezione del
156 % triangolo sul piano XY.
%-----
158 % Data:
% First version: 13/01/2021
160 % by A. Sommariva and M. Vianello.
%-----
162
r=sqrt( (vert(:,1)).^2 + (vert(:,2)).^2 );
164 r0=max(r);
intvf=[0,r0]; F=@(r) sqrt(1-r); f=chebfun(F,intvf);
166 m=max(1,(length(f)-1));

```

L'ultimo algoritmo riportato "dHYPERFIT2" fornisce i coefficienti dell'iperinterpolante di grado fissato, sul triangolo sferico scelto, dopo aver calcolato su quest'ultimo una base ortonormale.

```

function [coeff,R,jvec,dbox,degs] = dHYPERFIT2(...
2 deg,nodes,w,f,jvec,dbox,domain_structure,dimpoly)
%-----
4 % Descrizione:
% Questo algoritmo calcola i coefficienti
6 % dell'iperinterpolante di grado totale "deg", rispetto
% a una base "w"-ortonormale su un insieme di punti di
8 % dimensione d.
%-----
10 % Input:
% deg: grado totale dell'iperinterpolante;
12 % nodes: vettore colonna con le coordinate dei nodi;
% w: vettore colonna di pesi non negativi, o scalare non
14 % nel caso di pesi uguali;
% f: vettore colonna con i valori della funzione da
16 % approssimare nei nodi;
% * jvec: vettore di indici che selezione una base
18 % polinomiale;
% * dbox: variabile che definisce un "iperrettangolo"
20 % con i lati paralleli agli assi, contenente
% il dominio o l'insieme dei nodi.
22 % Se "dbox" non viene dato, si considera il
% minimo "iperrettangolo" con i lati paralleli
24 % agli assi, contenente il dominio o l'insieme
% dei nodi.
26 % Viene definito da una matrice "2 x d", dove

```

```

%           "d" coincide con la dimensione dello spazio in
28 %           cui si immerge il dominio.
% * domain_structure: struttura che definisce il dominio,
30 % domain_struct.domain: stringa con il nome del dominio.
% domain_struct.parms: parametri del dominio.
32 %
% Nota: le variabile contrassegnate da asterisco "*"
34 %       sono facoltative e possono essere poste come
%       matrici vuote.
36 %-----
% Output:
38 % coeff: vettore colonna con i coefficienti
%       dell'iperinterpolante;
40 % R: matrice triangolare nella fattorizzazione QR
%       diag(sqrt(w))*C(:,jvec)=Q*R
42 % dove "C=dCHEBVAND(n,Y)" (o una base particolare se
$       disponibile per il dominio considerato);
44 % jvec: vettore colonna di indici che seleziona una base
%       polinomiale;
46 % dbox: variabile che definisce l'insieme d-dimensionale
%       su cui adattare la base ortonormale.
48 %-----
% Data: Scritto il 26/07/2020
50 % da M. Dessolet, F. Marcuzzi, M. Vianello.
%
52 % Modificato da:
% 29/10/2020: M. Vianello.
54 % 03/11/2020: A. Sommariva.
% 04/11/2020: M. Dessolet and M. Vianello.
56 % 05/11/2020: A. Sommariva.
%-----
58 % ..... risoluzione problemi .....

60 if nargin < 8, dimpoly=[]; end
   if nargin < 7, domain_structure.domain='generic'; end
62 if isempty(domain_structure),
   domain_structure.domain='generic'; end
64 if nargin < 6, dbox=[]; end
   if nargin < 5, jvec=[]; end
66 if nargin < 4, w=[]; end
%-----
68 % ..... main code .....

70 domain=domain_structure.domain;
   switch domain
72
% domini con basi speciali disponibili
74 case {'sphere','circle','unit-simplex','unit-disk'}
[V,dbox,degs] = vandermonde_matrix(deg,...
76 nodes,dbox,domain_structure);

```

```

    coeff=(f.*w) '*V; coeff=coeff';
78 R=[]; jvec=1:length(degs);

80 case {'spherical-triangle','spherical-polygon',...
        'spherical-rectangle'}
82 dimpoly=(deg+1)^2;
    reorths=1;
84 [V,degs]=vandermonde_sphharm(deg,nodes);
    [~,jvec,Q,R,dbox]=dORTHVAND2(deg,nodes,w,jvec,V,dbox,...
86 dimpoly,reorths);
    coeff=Q'*(sqrt(w).*f);
88
    otherwise % caso generico
90 [~,jvec,Q,R,dbox,degs]=dORTHVAND2(deg,...
    nodes,w,jvec,[],dbox,dimpoly);
92 coeff=Q'*(sqrt(w).*f);

94 end

```


Bibliografia

- [1] C. An, H.N. Wu, *Lasso hyperinterpolation over general regions*, SIAM J Sci Comput 43(6), 2021, DOI: <http://dx.doi.org/10.1137/20M137793X>.
- [2] C. W. Clenshaw and A. R. Curtis, *A method for numerical integration on an automatic computer*, Numer. Math. 2, 1960, pp. 197-205, DOI: <https://doi.org/10.1007/BF01386223>.
- [3] G. Da Fies, A. Sommariva and M. Vianello, *Algebraic cubature by linear blending of elliptical arcs*, Appl. Numer. Math. 74, 2013, pp. 49–61, DOI: <http://dx.doi.org/10.1016/j.apnum.2013.08.003>.
- [4] K. Hesse, I.H. Sloan and R.S. Womersley, Numerical Integration on the Sphere, Handbook of Geomathematics, 2010, pp 1185–1219, DOI: [10.1007/978-3-642-01546-5_40](https://doi.org/10.1007/978-3-642-01546-5_40).
- [5] V. Lakshminarayanan and A. Fleck, *Zernike polynomials: a guide*, Journal of Modern Optics 58(7), 2011, DOI: <https://doi.org/10.1080/09500340.2011.554896>.
- [6] B. F. Logan and L. A. Shepp, *Optimal reconstruction of a function from its projections*, Duke Math. J., 42, 1975, pp. 645–659, DOI: <https://doi.org/10.1215/S0012-7094-75-04256-8>.
- [7] C. Müller, Spherical Harmonics, 1st edn. Springer-Verlag, Berlin, 1996, DOI: [10.1007/BFb0094775](https://doi.org/10.1007/BFb0094775).
- [8] I.H. Sloan, *Polynomial interpolation and hyperinterpolation over general regions*, J Approx Theory 83(2), 1995, pp.238–254, DOI: <https://doi.org/10.1006/jath.1995.1119>.
- [9] I. H. Sloan, R. S. Womersley, *Constructive Polynomial Approximation on the Sphere*, Journal of Approximation Theory, 103(1), 2000, pp. 91-118, DOI: <https://doi.org/10.1006/jath.1999.3426>.
- [10] I.H. Sloan and R.S. Womersley, *Filtered hyperinterpolation: a constructive polynomial approximation on the sphere*, Int J Geomath 3(1), 2012, pp.95-117, DOI: [10.1007/s13137-011-0029-7](https://doi.org/10.1007/s13137-011-0029-7).
- [11] A. Sommariva, M. Vianello, *Numerical hyperinterpolation over spherical triangles*, Math Comput Simul 190, 2021, pp.15–22, DOI: <https://doi.org/10.1016/j.matcom.2021.05.003>.