

# Indice

<b>Introduzione</b>	<b>3</b>
<b>1 Metodi Monte Carlo</b>	<b>4</b>
1.1 Introduzione . . . . .	4
1.2 Formule di cubatura . . . . .	5
1.3 Problemi di costruzione della cubatura . . . . .	5
1.4 Integrazione di Monte Carlo . . . . .	7
1.4.1 Errore dell'integrazione Monte Carlo . . . . .	8
<b>2 Metodi quasi-Monte Carlo</b>	<b>10</b>
2.1 Introduzione . . . . .	10
2.2 Successioni uniformemente distribuite . . . . .	11
2.3 Discrepanza . . . . .	11
2.4 Disuguaglianza di Koksma-Hlawka . . . . .	12
<b>3 Sperimentazione MC e QMC</b>	<b>15</b>
3.1 Integrazione Monte Carlo . . . . .	15
3.1.1 Esercizio 1 . . . . .	15
3.1.2 Esercizio 2 . . . . .	18
3.2 Integrazione quasi-Monte Carlo . . . . .	20
3.2.1 Successione di Halton . . . . .	20

3.2.2	Successione di Sobol . . . . .	21
3.2.3	Esercizio 3 . . . . .	23
<b>4</b>	<b>Lattice rules e worst-case error</b>	<b>25</b>
4.1	Lattice rules . . . . .	25
4.2	Worst-case error . . . . .	27
4.2.1	Reproducing kernel Hilbert space . . . . .	29
4.3	Integrazione nel contesto del worst-case error . . . . .	30
4.4	Reproducing kernel Hilbert spaces con pesi generici . . . . .	32
4.5	Spazi di Korobov . . . . .	33
4.6	Spazi di Sobolev . . . . .	34
4.7	Costruzione Component-by-Component . . . . .	36
4.7.1	La struttura di $\Omega_n$ . . . . .	37
<b>5</b>	<b>Sperimentazione sul worst-case error</b>	<b>40</b>
5.1	Esercizio 1 . . . . .	41
5.2	Confronto tra il caso pesato e il caso non pesato in spazi di Sobolev . . . . .	43
5.3	Esercizio 2 . . . . .	44
5.4	Esercizio 3 . . . . .	45
	<b>Conclusioni</b>	<b>47</b>

# Introduzione

Il seguente progetto tratta dei metodi numerici per l'integrazione multi-dimensionale. Si introdurranno, quindi, i Metodi Monte Carlo per la risoluzione del problema della cubatura. In seguito si descriveranno i Metodi quasi-Monte Carlo, introdotti per risolvere il problema della non limitatezza dell'errore per i Metodi Monte Carlo. I quasi-Monte Carlo inoltre fanno parte di una classe di metodi più ampia chiamata rank 1 lattice rules, in seguito verranno esposti e verrà descritto il worst-case error che indicherà la "bontà" di una rank-1 lattice rule.

Nel progetto in aggiunta verranno discussi concetti matematici fondamentali per la trattazione dei problemi presentati e per la loro sperimentazione.

Inoltre i capitoli intermedi saranno dedicati alla parte sperimentale, implementata attraverso Matlab e i cui codici saranno presentati negli allegati del progetto. Per la sperimentazione si sono implementati i codici per l'integrazione Monte Carlo e quasi-Monte Carlo. Nell'ultimo capitolo invece si è descritto e attuato un algoritmo per il worst-case error quadratico di una rank 1 lattice rule in uno spazio di Sobolev.

# Capitolo 1

## Metodi Monte Carlo

### 1.1 Introduzione

Nell'integrazione numerica, quando le funzioni integrande possono essere approssimate con funzioni note come polinomi, i metodi di cubatura sono efficaci. Tuttavia la costruzione delle formule di cubatura non è conosciuta in generale e, negli esempi noti, gli integrali sono estesi a domini particolari come ipercubi o sfere. In realtà, le formule di cubatura possono essere costruite a partire da misure di probabilità purchè si abbia un campione i.i.d. (cioè le cui osservazioni sono indipendenti e identicamente distribuite).

Sia  $X$  una variabile aleatoria a valori in un certo spazio astratto, allora il problema della cubatura consiste nel cercare una buona approssimazione del valore atteso  $E[f(X)]$ , per ogni funzione integranda. Uno dei metodi che consente di risolvere questo problema è il *metodo Monte Carlo*. In questo paragrafo verrà descritto il metodo con relativa sperimentazione.

## 1.2 Formule di cubatura

Si consideri il seguente problema d'integrazione

$$\int_{\Omega} f(x) dx$$

dove  $f : \Omega \rightarrow \mathbb{R}$ ,  $\Omega \subseteq \mathbb{R}^d$  insieme di Borel e si consideri una misura di probabilità  $\mu$  su  $\Omega$  (dotata della  $\sigma$ -algebra di Borel). Dati  $x_1, \dots, x_n \in \Omega$  nodi e  $w_1, \dots, w_n$  pesi positivi, una *formula di cubatura di grado  $t$*  è data da

$$\int_{\Omega} f(x) d\mu(x) = \sum_{i=1}^n w_i f(x_i)$$

per ogni  $f \in \mathcal{P}_t(\Omega)$ , insieme di tutti i polinomi reali su  $\Omega$  di grado al più  $t$ . Se l'aspettazione  $E[|f_i(X)|] < \infty$  per ogni  $i = 1, \dots, d$ , una formula di cubatura può essere descritta nel seguente modo

$$E[f_i(X)] = \sum_{i=1}^n w_i f_i(x_i) \quad (1.1)$$

La precedente formula può essere riscritta anche nel caso vettoriale.

## 1.3 Problemi di costruzione della cubatura

In questo paragrafo saranno analizzati due problemi diversi. In entrambi verranno considerati come dati iniziali un campione casuale  $(X_1, \dots, X_n)$  i.i.d con genitrice  $X$  a valori in  $\Omega$  e una funzione vettoriale  $f : \Omega \rightarrow \mathbb{R}^d$  con la condizione  $E[||f(X)||] < \infty$ .

1. **Problema di cubatura esatta:** si può assumere di poter calcolare il valore esatto di  $E[f(X)]$  trovando  $n \leq d$  punti  $x_1, \dots, x_n \in \Omega$  e pesi  $w_1, \dots, w_n > 0$  tali che

$$f(x_1), \dots, f(x_n) \in \text{supp} P_{f(X)} \quad E[f(x)] = \sum_{i=1}^n w_i f(x_i)$$

dove  $P_{f(X)}$  è la distribuzione di  $f(X)$  su  $\mathbb{R}^d$ . Quindi il supporto sarà

$$\text{supp}P_{f(X)} = \{x \in \mathbb{R}^d | P(f(X) \in A) > 0 \text{ per un qualsiasi intorno } A \text{ di } x\}$$

**2. Problema di cubatura approssimata:** senza alcuna conoscenza del valore esatto di  $E[f(X)]$  (ad eccezione di  $E[f_1(X)] = 1$  per semplicità), si trovano  $n \leq d$  punti  $x_1, \dots, x_n \in \Omega$  e pesi  $w_1, \dots, w_n > 0$  tali che

$$f(x_1), \dots, f(x_n) \in \text{supp}P_{f(X)}, \quad E[f(x)] \approx \sum_{i=1}^n w_i f(x_i)$$

Nel problema 2), dato che non si conosce  $E[f(X)]$ , è quasi impossibile trovare una formula di cubatura esatta. Tuttavia, come successivamente verrà mostrato, l'integrazione Monte Carlo, come quella quasi Monte Carlo, approssima bene l'aspettazione  $E[f(X)]$  pur non conoscendo a priori i suoi valori.

Dato  $S \subset \mathbb{R}^d$ , si definisce *l'involuppo convesso di S* come

$$\text{conv}S := \left\{ \sum_{i=1}^m w_i x_i : m \geq 1, w_i > 0, x_i \in S, \sum_{i=1}^m w_i = 1 \right\}$$

Per risolvere il problema 1., partendo da un campione casuale  $(X_1, \dots, X_n)$  i.i.d. come set di punti da usare nella formula di cubatura, se si ha

$$E[f(X)] \in \text{conv}\{f(X_1), \dots, f(X_n)\} \quad (1.2)$$

allora si può costruire una formula di cubatura utilizzando il *metodo di Sottocampionamento di Carathéodory–Tchakaloff*, il quale, a partire da una misura discreta  $\mu$  (la quale può essere già approssimazione di una misura non discreta), ottiene una misura  $\mu_{CT}$  discreta, che è uguale a  $\mu$  nell'integrazione delle funzioni test  $f_1, \dots, f_n$ , quindi può essere considerata come una compressione di  $\mu$ . Nei recenti sviluppi, la misura  $\mu_{CT}$  è stata anche ottenuta come soluzione ammissibile di base di un problema di Programmazione Lineare.

Come già è stato detto, è possibile applicare il metodo se si verifica l'ipotesi (1.2). Il seguente teorema ci assicura l'esistenza di un  $n$  tale che (1.2) risulti verificata.

**Teorema 1.1.** *Sia  $(X_1, \dots, X_n)$  un campione casuale i.i.d. con genitrice  $X$ . Allora esiste un intero positivo  $n$  tale che  $E[f(X)]$  è contenuto in  $\text{conv}\{f(X_1), \dots, f(X_n)\}$  quasi certamente.*

Per risolvere il problema 2. bisogna distinguere due casi:

- a) Si conosce la distribuzione di  $X$ , ma non il valore esatto di  $E[f(X)]$ .
- b) Non si conosce la distribuzione di  $X$ .

Nel caso a) conoscendo la distribuzione  $P_X$ , essa viene approssimata da una  $\mu$  discreta e si applica il metodo di sottocampionamento di Carathéodory–Tchakaloff.

Nel caso b) bisogna ricorrere all'integrazione di Monte Carlo, che verrà esposta nella sezione successiva. Tuttavia se si dispone di un campione casuale  $(X_1, \dots, X_n)$  i.i.d. si può costruire  $\mu_{CT}$ . Ad ogni modo l'integrazione Monte Carlo resta quella più efficace in quanto si usano  $n$  punti di valutazione di ogni integranda, invece di  $d$  punti.

## 1.4 Integrazione di Monte Carlo

Si consideri un campione casuale  $(X_1, \dots, X_n)$  i.i.d. con genitrice  $X$  con realizzazioni in  $\Omega$ , osservando che la media campionaria  $\frac{1}{n}(f(X_1) + \dots + f(X_n))$  è uno stimatore esatto per  $E[f(X)]$ , si ricava che

$$E(f) \approx E_n(f) = \frac{1}{n} \sum_{i=1}^n f(x_i) \quad (1.3)$$

Questo risultato segue dalla Legge Forte dei Grandi Numeri; infatti

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n f(x_i) = E(f)$$

Applicando la stima (1.1) alla formula (1.3) si ottiene la *stima di Monte Carlo*

$$I(f) = \int_{\Omega} f(x) dx \approx \frac{\lambda_d(\Omega)}{n} \sum_{i=1}^n f(x_i) := I_n(f) \quad (1.4)$$

dove  $\lambda_d(\Omega)$  è la misura di Lebesgue  $d$ -dimensionale finita.

### 1.4.1 Errore dell'integrazione Monte Carlo

Si definisce l'errore dell'integrazione Monte Carlo come

$$\epsilon_n := I(f) - I_n(f) \quad (1.5)$$

La varianza di  $f$  è definita come

$$\sigma^2(f) = \int_{\Omega} (f(x) - E(f))^2 d\mu(x)$$

Il seguente teorema è importante ai fini della stima dell'errore;

**Teorema 1.2.** *Se la varianza di  $f$  è finita, allora per ogni  $n \geq 1$  si ha*

$$\int_{\Omega} \dots \int_{\Omega} \left( \frac{1}{n} \sum_{i=1}^n f(x_i) - E(f) \right)^2 d\mu(x_1) \dots d\mu(x_n) = \frac{\sigma^2(f)}{n}$$

Questo risultato mostra che l'errore commesso nell'approssimare il valore atteso (1.3) è in media  $\sigma(f)n^{-1/2}$ , dove  $\sigma(f)$  è la deviazione standard di  $f$ .

**Corollario 1.1.** *Sotto le ipotesi del precedente teorema, si ha la seguente stima dell'errore dell'integrazione Monte Carlo*

$$[E(\epsilon_n^2)]^{1/2} = \lambda_d(\Omega) \sigma(f) n^{-1/2}$$



**Teorema 1.3** (Teorema del Limite Centrale). *Sia  $\sigma^2(f)$  la varianza di  $f$  tale che essa sia finita. Allora esiste una variabile aleatoria normale standard  $Z \sim N(0, 1)$  tale che*

$$\lim_{n \rightarrow \infty} \epsilon_n(f) = \lambda_d(\Omega) \sigma(f) n^{-1/2} Z$$

Ciò significa che l'errore dell'integrazione Monte Carlo è dell'ordine  $O(n^{-1/2})$  con una costante moltiplicativa che è la radice quadrata della varianza della funzione integranda, moltiplicata per la misura del dominio. Inoltre, grazie a questo teorema si ottiene che la distribuzione dell'errore è approssimativamente quella di una variabile aleatoria normale. La particolarità di questo risultato si riscontra nel fatto che anziché avere un upper bound, si ottiene una stima probabilistica della grandezza dell'errore e della sua distribuzione. Il vantaggio è che questa stima risulta indipendente dalla dimensione  $d$  dello spazio; nello scorso paragrafo infatti è già stato accennato come il metodo di integrazione Monte Carlo risulti più efficiente per questo motivo.

## Capitolo 2

# Metodi quasi-Monte Carlo

### 2.1 Introduzione

Come spiegato alla fine del capitolo precedente, l'integrazione Monte Carlo gode di alcuni vantaggi. Nonostante ciò nella costruzione della stima dell'errore si è ottenuta una stima probabilistica di esso, non un bound, quindi l'accuratezza richiesta potrebbe non essere raggiunta realmente. Inoltre tale stima risulta essere dipendente dalla varianza di  $f$ , quindi dall'ipotesi che  $f \in L^2(\mu)$ , e ciò risulta problematico nell'ambito dell'analisi numerica nel quale si vuole ottenere la stima dell'integrale di  $f$  senza preoccuparsi troppo delle sue proprietà. Infine, per applicare il Metodo Monte Carlo, è richiesta la generazione di un campione casuale, la quale potrebbe non essere semplice. L'idea di base per i metodi quasi-Monte Carlo è quella di sostituire i campioni casuali del metodo Monte Carlo con punti deterministici opportunamente scelti. Nel caso dell'integrazione numerica, il criterio è quello di cercare successioni uniformemente distribuite o a bassa discrepanza. Inoltre si otterrà un bound per l'errore dato dalla disuguaglianza di Koksma-Hlawka.

## 2.2 Successioni uniformemente distribuite

Supponiamo di avere come dominio di integrazione  $\Omega$  il cubo unitario chiuso  $[0, 1]^s$ . Data la funzione integranda  $f : [0, 1]^s \longrightarrow \mathbb{R}$  si consideri l'approssimazione quasi-Monte Carlo

$$\int_{[0,1]^s} f(x)dx \approx \frac{1}{n} \sum_{i=1}^n f(x_i) \quad (2.1)$$

dove  $x_1, \dots, x_n \in [0, 1]^s$ . Si può supporre di avere una successione infinita  $\{x_i\}_{i \in \mathbb{N}} \subset [0, 1]^s$  invece della successione finita. Bisogna dunque richiedere un metodo di convergenza, cioè che

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n f(x_i) = \int_{[0,1]^s} f(x)dx \quad (2.2)$$

La validità di questa equazione equivale a chiedere che la successione  $\{x_i\}_{i \in \mathbb{N}}$  sia uniformemente distribuita su  $[0, 1]^s$ .

**Definizione 2.1.** *Dato  $E$  uno spazio di Hausdorff compatto e  $\mu$  una misura positiva regolare normata di Borel. La successione  $\{x_i\}_{i \in \mathbb{N}} \subset E$  è detta  $\mu$ -uniformemente distribuita in  $E$  se per ogni sottinsieme  $B \subset E$  vale*

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n \chi_B(x_i) = \mu(B)$$

dove  $\chi_B$  è la funzione caratteristica di  $B$ .

## 2.3 Discrepanza

La discrepanza di un insieme di punti misura quanto essi si discostano da una distribuzione uniforme.

**Definizione 2.2** (Discrepanza locale). *Sia  $P_n$  un insieme di  $n$  punti e sia  $J$  un arbitrario sottinsieme di un cubo unitario  $s$ -dimensionale  $[1, 0]^s$ . Si*

definisce la discrepanza locale come

$$discr(J, P_n) := \frac{|P_n \cap J|}{n} - vol(J)$$

Essa misura la mancata corrispondenza tra il numero effettivo di punti di  $J$  e il numero atteso di punti di  $J$ .

**Definizione 2.3** (Discrepanza estrema). *La discrepanza estrema considera tutti i rettangoli regolari*

$$D_\infty(P_n) := \sup_{\mathbf{x}, \mathbf{y} \in [0,1]^s} |discr([\mathbf{x}, \mathbf{y}), P_n)|$$

dove  $[\mathbf{x}, \mathbf{y}) = [x_1, y_1) \times \dots \times [x_s, y_s)$ .

**Definizione 2.4** (Discrepanza star). *La discrepanza star considera tutti gli intervalli rettangolari ancorati all'origine*

$$D_\infty^*(P_n) := \sup_{\mathbf{x} \in [0,1]^s} |discr([0, \mathbf{x}), P_n)|$$

Nelle due definizioni appena enunciate si usa la norma  $L_\infty$ ; nonostante ciò sono comuni anche varianti con la norma  $L_p$  o la norma  $L_2$ . Quando la discrepanza di un insieme di punti è  $O(n^{-1}(\log(n))^s)$  allora esso è comunemente chiamato **insieme a bassa discrepanza**.

## 2.4 Disuguaglianza di Koksma-Hlawka

Come è stato precedentemente spiegato, l'obiettivo è quello di ottenere un bound superiore per l'errore; a tal proposito è necessario introdurre la nozione di variazione limitata.

**Definizione 2.5.** *La variazione di una funzione unidimensionale  $f$  è definita come*

$$V(f) := \sup_{\mathcal{P}} \sum_{x_i, x_{i+1} \in \mathcal{P}} |f(x_{i+1}) - f(x_i)|$$

dove l'estremo superiore varia su tutte le partizioni dell'intervallo  $[0, 1]$

$$\mathcal{P} = \{0 = x_0 \leq x_1 \leq \dots \leq x_{n-1} \leq x_n = 1\}$$

La diretta generalizzazione per una funzione a più variabili è chiamata variazione nel senso di Vitali.

**Definizione 2.6** (Variazione nel senso di Vitali). *Sia  $\mathcal{D}_s = \{1, \dots, s\}$ . La variazione nel senso di Vitali di una funzione  $f$  è definita come*

$$V_{1, \mathcal{D}_s}(f) := \sup_{\mathcal{P}} \sum_{J \in \mathcal{P}} |\Delta(f, J)|$$

dove  $\mathcal{P}$  è l'insieme di tutte le partizioni di  $[0, 1]^s$  e  $\Delta(f, J)$  è l'operatore differenza  $s$ -dimensionale.

Se le derivate parziali esistono e sono continue nel cubo, allora la variazione nel senso di Vitali può essere scritta come

$$V_{1, \mathcal{D}_s}(f) = \int_{[0, 1]^s} \left| \frac{\partial^s f(x)}{\partial x_1 \dots \partial x_s} \right| dx$$

**Definizione 2.7** (Variazione nel senso di Hardy e Krause). *Si definisce la variazione nel senso di Hardy e Krause come*

$$V_1(f) := \sum_{\emptyset \neq u \subseteq \mathcal{D}_s} V_{1, u}(f(x_u, 1))$$

dove  $f(x_u, 1)$  è la restrizione di  $f$  che vale 1 quando la coordinata non appartiene all'insieme  $u$ .

**Teorema 2.1** (Disuguaglianza di Koksma-Hlawka). *Se la funzione  $f$  ha variazione limitata nel senso di Hardy e Krause, allora per ogni insieme di punti  $P_n$*

$$|Q(f; P_n) - I(f)| \leq D_{\infty}^*(P_n) V_1(f)$$

Da questa disuguaglianza segue che l'errore per l'integrazione numerica può essere scritto in termini di due fattori indipendenti: uno che coinvolge solo la funzione integranda, l'altro che coinvolge solo l'insieme di punti utilizzato. E' importante riconoscere che la variazione  $V_1(f)$  misura l'irregolarità della funzione  $f$  rispetto all'integrazione quasi-Monte Carlo.

## Capitolo 3

# Sperimentazione MC e QMC

### 3.1 Integrazione Monte Carlo

In questa sezione saranno esposti risultati per l'integrazione numerica ottenuti tramite integrazione Monte Carlo. Lo scopo dell'algoritmo implementato è quello di calcolare l'integrale partendo da un set di punti casuali, confrontarlo con il risultato di `integral` di Matlab e visualizzare la distribuzione dell'errore osservandone il comportamento. Il vettore di punti casuali è stato generato attraverso la function `rand`, la quale restituisce un vettore di lunghezza  $n$  di numeri casuali distribuiti uniformemente nell'intervallo  $[0, 1]$ ; nel codice è stato anche implementato uno shift per permettere l'integrazione in qualsiasi intervallo  $[a, b]$ .

#### 3.1.1 Esercizio 1

Per il primo esempio sono stati forniti i seguenti dati:

$$f = e^x x^2 \sqrt{e^x}; \quad x \in [2, 4]; \quad N = 10000$$

Applicando l'algoritmo implementato risulta:

Integrale Montecarlo = 3.133964e+03

Integrale Matlab = 3.078140e+03

Con errore = 6.003015e-02

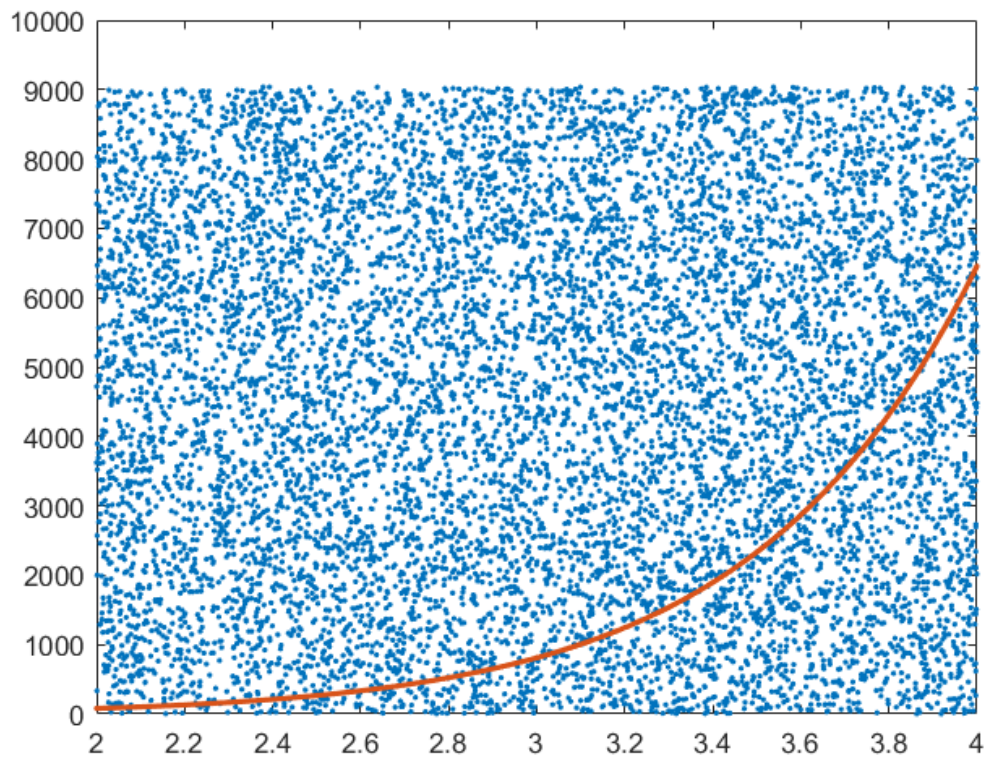


Figura 3.1: Integranda rappresentata attraverso la distribuzione dei punti random

Nella figura (3.1) sono rappresentati in blu i punti generati dalla function `rand` e in rosso il grafico della funzione integranda. Ovviamente, come è noto dalla teoria, non tutti i punti generati danno un contributo effettivo nel calcolo dell'integrale, sono rilevanti infatti esclusivamente i punti che si trovano al di sotto del grafico della funzione. Per questo motivo nel codice



implementato è stata fatta una sorta di selezione andando a controllare la posizione occupata dai punti casuali rispetto al grafico della funzione in esame. Per quanto riguarda l'errore, si fa invece riferimento alla figura (3.2) nella quale sono messe a confronto la propagazione dell'errore e la distribuzione di una normale standard.

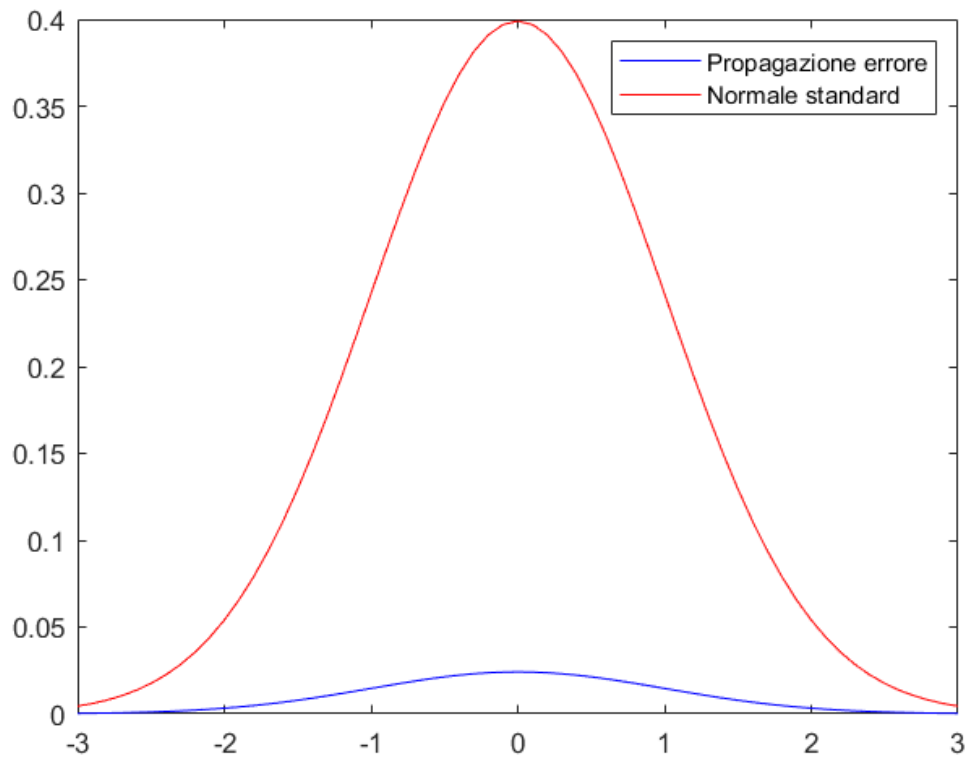


Figura 3.2: Distribuzione dell'errore

In questo caso particolare dato che la funzione integranda gode della proprietà di essere  $L^2$ , come visto nella teoria, la propagazione dell'errore si mantiene al di sotto della gaussiana pur non avendo un vero e proprio bound.

### 3.1.2 Esercizio 2

Per il secondo esempio sono stati forniti i seguenti dati:

$$f = 10 + 5\sin(5x); \quad x \in [-1, 3]; \quad N = 10000$$

Applicando l'algoritmo implementato si ottiene:

Integrale Montecarlo = 4.151725e+01

Integrale Matlab = 4.104335e+01

Con errore = 2.779190e+01

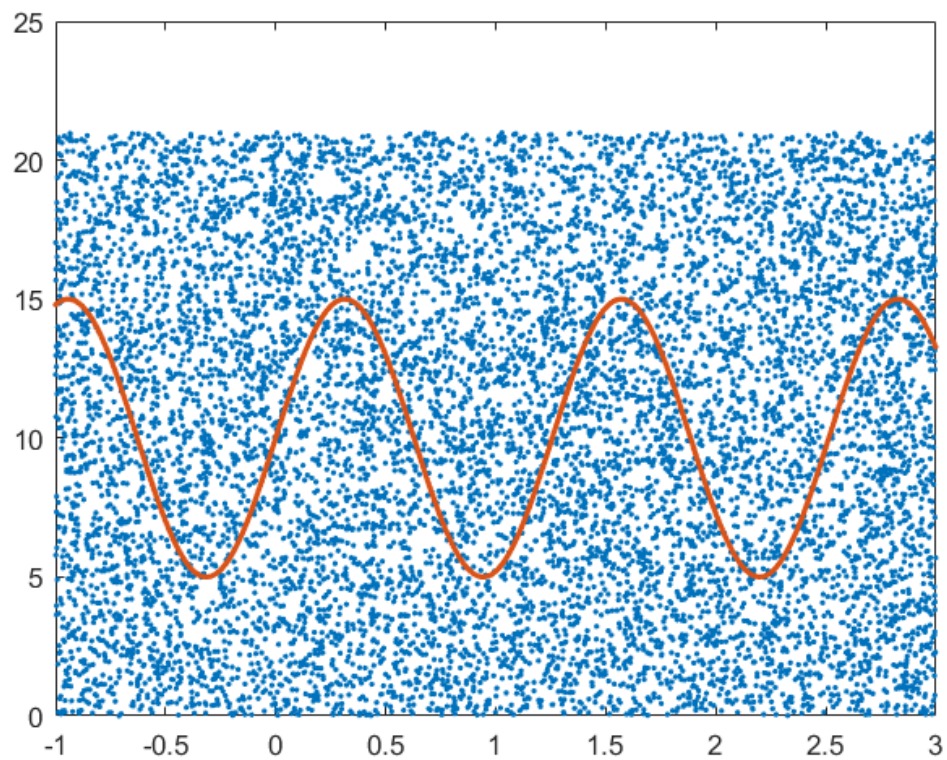


Figura 3.3: Integranda rappresentata attraverso la distribuzione dei punti random

Nella figura (3.3) sono rappresentati in blu i punti generati dalla function `rand` e in rosso il grafico della funzione integranda.

Per quanto riguarda l'errore questa volta la situazione cambia;

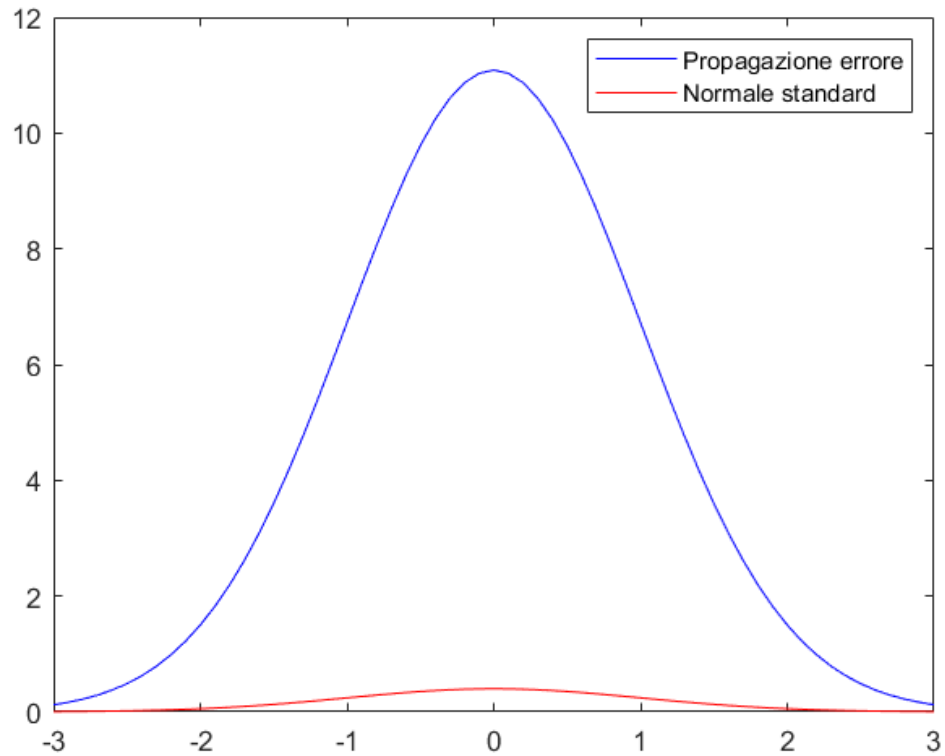


Figura 3.4: Distribuzione dell'errore

infatti avendo utilizzato una funzione periodica si nota che, oltre ad avere un errore molto alto, anche la sua propagazione risulta al di sopra della gaussiana. Questo perchè, come è stato spiegato nella teoria, l'errore e la sua propagazione dipendono dalle caratteristiche della funzione integranda e non avendo un bound dell'errore, esso può essere anche molto alto.

## 3.2 Integrazione quasi-Monte Carlo

Come è stato esposto nella teoria, per i Metodi quasi-Monte Carlo non è richiesta la generazione di punti casuali, ma si lavora con successioni di punti uniformemente distribuiti e a bassa discrepanza all'interno del cubo  $[0, 1]^d$ . In particolare seguiranno la descrizione delle successioni di Halton e Sobol.

### 3.2.1 Successione di Halton

La successione di Halton  $d$ -dimensionale è una sequenza a bassa discrepanza. Si consideri un intero  $p \geq 2$ , ogni intero  $n \geq 0$  può essere univocamente scritto in base  $p$

$$n = \sum_{i=0}^{\infty} a_i(n) p^i$$

dove  $a_i(n) \in \{0, \dots, p-1\}$  per ogni  $i \geq 0$  ed esiste un  $k \geq 0$  tale che  $a_i(n) = 0$  per  $i > k$ , ovvero la somma è finita. Per ogni  $n \geq 0$  si costruiscono le trasformazioni  $\Phi_p(n) = \sum_{i=0}^{\infty} a_i(n) p^{-(i+1)}$ . Si definisce successione di Halton in base  $p_1, \dots, p_d$  la successione  $H_{d,N}$  di  $N+1$  punti su  $[0, 1]^d$  come

$$H_{d,N} := \{(\Phi_{p_1}(n), \dots, \Phi_{p_d}(n)), n = 0, \dots, N\}$$

La successione di Halton ha discrepanza star  $D^*(H_{d,N})$  dell'ordine  $O(N^{-1}(\log(N))^d)$  e di conseguenza tale successione è a bassa discrepanza.

Nella figura (3.5) è mostrato il grafico della dispersione dei punti di Halton.

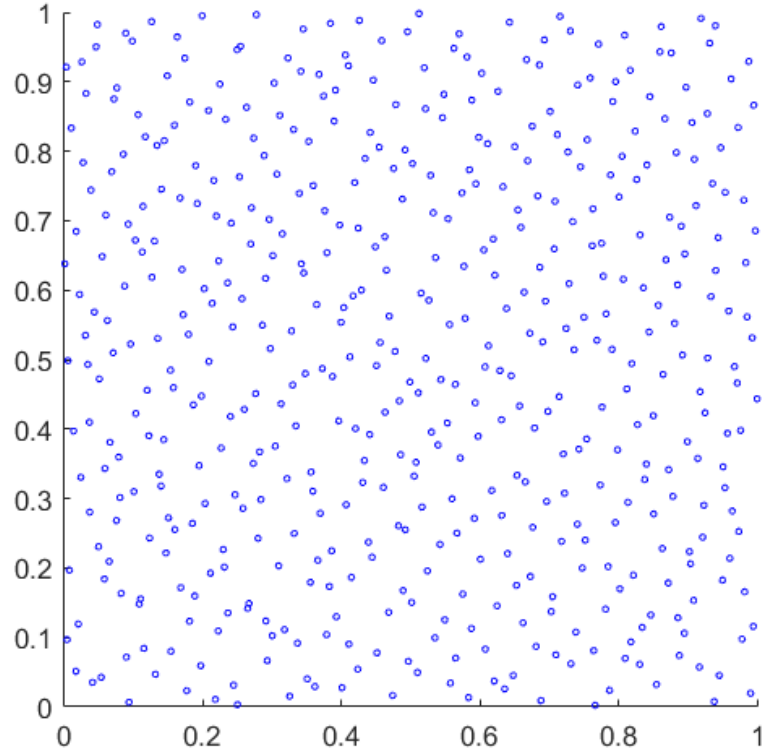


Figura 3.5: Punti di Halton su  $[0, 1]^3$ ,  $N = 500$

### 3.2.2 Successione di Sobol

La successione di Sobol è una successione a bassa discrepanza la quale fa uso dei polinomi primitivi. Una successione di Sobol  $S_{d,N}$  è formata da  $N$  punti sul cubo  $[0, 1]^d$ . Per generare l' $i$ -esimo punto della successione occorre scegliere un polinomio primitivo di grado arbitrario  $s_i$  a coefficienti nel campo  $\mathbb{Z}_2$ , ovvero un polinomio della forma

$$x_{s_i} + a_1 x_{s_i-1} + a_2 x_{s_i-2} + \dots + a_{s_i-t} x + 1$$

dove i coefficienti  $a_1, \dots, a_{s_t}$  sono 0 o 1. Il polinomio primitivo viene utilizzato per definire una successione di interi positivi  $\{m_{k,i}\}_{k \geq 1}$  attraverso la seguente relazione ricorsiva

$$m_{k,i} = 2a_1m_{k-1,i} \oplus 2^2a_2m_{k-2,i} \oplus \dots \oplus 2^{s_t}m_{k-s_t,i} \oplus m_{k-s_t,i}$$

dove il simbolo  $\oplus$  è l'operatore di somma bit-a-bit modulo 2, ovvero l'operazione che, dati due numeri, ne considera le scritture in binario e va a sommare a due a due le cifre che si trovano nella medesima posizione, senza tener conto di eventuali riporti.

Una volta ricavata la successione  $\{m_{k,i}\}_{k \geq 1}$ , si ricavano i direction numbers  $\{\nu_{k,i}\}_{k \in \mathbb{N}}$

$$\nu_{k,i} = \frac{m_{k,i}}{2^k}$$

Quindi la  $j$ -esima componente dell' $i$ -esimo punto della successione di Sobol è data da

$$x_{i,j} := i_1\nu_{1,j} \oplus i_2\nu_{2,j} \oplus \dots$$

dove  $i_k$  è la  $k$ -esima cifra da destra nella rappresentazione binaria di  $i = (\dots i_3 i_2 i_1)_2$ . Nella figura (3.6) è mostrato il grafico della dispersione dei punti di Sobol. Scegliendo  $d$  differenti polinomi primitivi per calcolare  $d$  diversi insiemi di direction numbers, l'insieme dei punti di Sobol  $S_{d,N}$  ha discrepanza dell'ordine  $O((\log(N))^d)$ . Quindi sia nel caso della successione di Halton che in quella di Sobol, la discrepanza è dell'ordine di  $O(N^{-1}(\log(N))^d)$ , quindi entrambe le successioni sono insiemi a bassa discrepanza e si può utilizzare il metodo quasi-Monte Carlo.

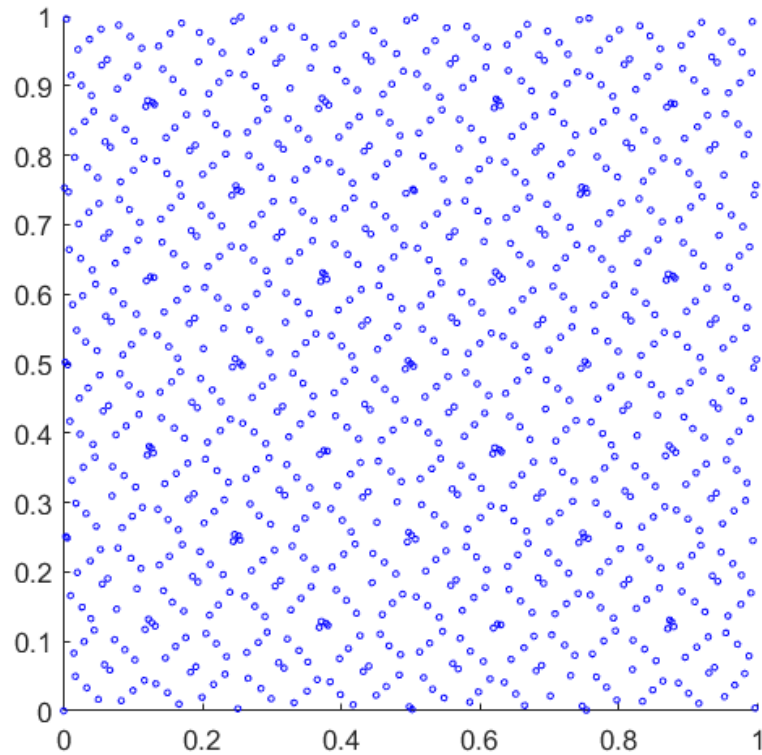


Figura 3.6: Punti di Sobol su  $[0, 1]^2$ ,  $N = 1000$

### 3.2.3 Esercizio 3

Riprendendo i dati dell'**Esercizio 2**

$$f = 10 + 5\sin(5x); \quad x \in [0, 1]; \quad N = 1000$$

applicando l'algoritmo implementato per generare punti di Halton e applicando l'integrazione quasi-Monte Carlo, si ottiene:

Integrale quasi-Monte Carlo = 1.123495e+01

Integrale Matlab = 1.071634e+01

Con errore = 5.186127e-01

Dove l'errore è stato calcolato tramite la differenza in valore assoluto tra il valore esatto dell'integrale e il valore dell'integrale calcolato attraverso l'algoritmo. Il calcolo dell'errore in questo modo è stato possibile siccome, come si è visto nella teoria, la disuguaglianza di Koksma-Hlawka fornisce un upper bound per l'errore.

Applicando l'algoritmo per generare punti di Sobol e applicando l'integrazione quasi-Monte Carlo, si ottiene:

Integrale quasi-Monte Carlo = 1.045795e+01

Integrale Matlab = 1.071634e+01

Con errore = 2.583839e-01

con errore calcolato ancora ricorrendo alla differenza in modulo degli integrali.



## Capitolo 4

# Lattice rules e worst-case error

### 4.1 Lattice rules

Una lattice rule è una formula di cubatura che approssima un integrale multidimensionale

$$\int_{[0,1]^s} f(x) dx$$

utilizzando come ascisse i punti di un reticolo di integrazione. Un reticolo (in inglese lattice)  $s$ -dimensionale (indicato convenzionalmente con  $\Lambda$ ) è un insieme di punti in  $\mathbb{R}$  che non contiene punti limite e tali che valga  $p, q \in \Lambda \implies p \pm q \in \Lambda$ . Sia  $\{w_1, \dots, w_t\}$  tale che ogni elemento  $u \in \Lambda$  può essere scritto come

$$u = \lambda_1 w_1 + \dots + \lambda_t w_t \quad \lambda \in \mathbb{Z}^t$$

Fissato  $n$  come il comun denominatore delle componenti di  $w$  e definendo  $z \in \mathbb{Z}^s$  come  $w = z/n$  si ottiene l'insieme dei punti

$$x_k = \left\{ \frac{kz}{n} \right\} \quad \text{per } k = 0, 1, \dots, n-1 \quad (4.1)$$

dove  $\{\cdot\}$  sta a denotare che si considera la parte frazionaria componente per componente. L'insieme (4.1) è l'insieme di nodi per una rank-1 lattice rule.

**Definizione 4.1.** Una lattice rule è una regola  $Q$  della forma

$$Q(f) = \frac{1}{n} \sum_{x_k \in P_n} f(x_k)$$

In generale una lattice rule è della forma

$$Q(f) = \frac{1}{m} \sum_{k_1=0}^{n_1-1} \dots \sum_{k_t=0}^{n_t-1} f\left(\left\{\frac{k_1 z_1}{n_1} + \dots + \frac{k_t z_t}{n_t}\right\}\right)$$

con insieme di generatori  $z_i/n_i \in \mathbb{Q}^s$  e  $m = n_1 \dots n_t \leq s$ .

Il *rank* è il minimo numero di sommatorie nella precedente formula. La classica lattice rule con insieme di punti  $P_n$  ha rank 1 e  $n$  punti differenti se  $\text{mcd}(z_1, \dots, z_s, n) = 1$ .

**Definizione 4.2.** Una rank-1 lattice rule è una regola  $Q$  della forma

$$Q(f) = \frac{1}{n} \sum_{k=0}^{n-1} f\left(\left\{\frac{kz}{n}\right\}\right)$$

dove  $z \in \mathbb{Z}^s$  è il vettore generatore.

Una lattice rule ha  $n$  punti distinti se e solo se ogni componente del vettore generatore  $z$  è coprima con  $n$ .

Per garantire la validità di tale proprietà si sceglie  $n$  numero primo. Il caso in cui  $n$  è generico non verrà trattato.

**Definizione 4.3.** Il reticolo duale  $\Lambda^\perp$  di un reticolo  $\Lambda$  è dato da

$$\Lambda^\perp := \{h \in \mathbb{Z}^s : h \cdot x \in \mathbb{Z} \quad \text{per ogni } x \in \Lambda\}$$

Si può dimostrare che  $D_\infty^*(P_n)$  può essere maggiorata con un quantità dell'ordine  $O(n^{-1}(\log(n))^s)$ . Tale maggiorazione rende  $P_n$  un insieme a bassa discrepanza, i suoi elementi possono dunque essere scelti come nodi per l'integrazione quasi-Monte Carlo, la quale appartiene alla classe delle rank-1 lattice.

## 4.2 Worst-case error

Quando una lattice rule si può considerare "buona"? Bisogna fare considerazioni sull'errore. In questa sezione verrà esposta la costruzione del *worst-case error* per una rank-1 lattice rule.

La funzione integranda  $f$  si può esprimere come una serie di Fourier assolutamente convergente

$$f(x) = \sum_{\mathbf{h} \in \mathbb{Z}^s} \hat{f}(\mathbf{h}) \exp(2\pi i \mathbf{h} \cdot \mathbf{x})$$

con  $\mathbf{h} \cdot \mathbf{x}$  prodotto interno in  $\ell_2$ , con coefficienti di Fourier dati da

$$\int_{[0,1]^s} f(x) \exp(-2\pi i \mathbf{h} \cdot \mathbf{x}) dx$$

Osserviamo che tale richiesta obbliga le funzioni ad essere periodiche in  $[0, 1]^s$

**Teorema 4.1.** *Per una rank-1 lattice rule  $Q$   $s$ -dimensionale di  $n$  punti, con vettore generatore  $z$  e una funzione  $f$  espressa come una serie di Fourier assolutamente convergente, l'errore può essere scritto come*

$$Q(f) - I(f) = \sum_{\substack{\mathbf{h} \in \mathbb{Z}^s / \{\mathbf{0}\} \\ \mathbf{h} \cdot \mathbf{z} \equiv 0 \pmod{n}}} \hat{f}(\mathbf{h})$$

**Definizione 4.4.** *Sia  $E_\alpha^s(c)$ ,  $\alpha > 1$  e  $c > 0$  la classe delle funzioni periodiche in  $[0, 1]^s$  per le quali*

$$|\hat{f}(\mathbf{h})| \leq c r(\mathbf{h})^{-\alpha}$$

con

$$r(\mathbf{h}) := \prod_{j=1}^s r(h_j) \quad r(h) := \max(1, |h|)$$

Inoltre  $\alpha$  deve essere maggiore di 1, in modo che si possa definire la funzione zeta di Riemann

$$\zeta(\alpha) := \sum_{h=1}^{\infty} \frac{1}{h^\alpha}$$

con  $\zeta(\alpha) < \infty$  per  $\alpha > 1$ .

**Definizione 4.5.** Per una rank-1 lattice rule di  $n$  punti con vettore generatore  $z \in \mathbb{Z}^s$  e  $\alpha > 1$  si definisce

$$P_\alpha(z, n) := \sum_{\substack{\mathbf{h} \in \mathbb{Z}^s / \{\mathbf{0}\} \\ \mathbf{h} \cdot \mathbf{z} \equiv 0 \pmod{n}}} r(\mathbf{h})^{-\alpha}$$

Quindi sussiste il seguente teorema

**Teorema 4.2.** Se  $Q$  è una lattice rule a  $n$  punti con vettore generatore  $z \in \mathbb{Z}^s$ ,  $\alpha > 1$  e  $c > 0$  allora

$$\max_{f \in E_\alpha^s(c)} |Q(f) - I(f)| = cP_\alpha(z, n)$$

**Corollario 4.1.** La quantità  $P_\alpha(z, n)$  può essere calcolata come l'errore di integrazione della funzione

$$f_\alpha(\mathbf{x}) := \sum_{\mathbf{h} \in \mathbb{Z}^s} r(\mathbf{h})^{-\alpha} \exp(2\pi i \mathbf{h} \cdot \mathbf{x}) \quad (4.2)$$

Si dimostra che la (4.2) è equivalente alla seguente espressione

$$f_\alpha(\mathbf{x}) = \prod_{j=1}^s \left( 1 + \frac{(2\pi)^\alpha}{(-1)^{\alpha/2-1} \alpha!} B_\alpha(x) \right) \quad 0 \leq x \leq 1$$

dove  $B_\alpha$  rappresenta il polinomio di Bernoulli di grado  $\alpha$ . I polinomi di Bernoulli godono delle seguenti proprietà:

$$B_\alpha(x) = (-1)^\alpha B_\alpha(1-x)$$

$$\int_0^1 B_\alpha(x) dx = 0 \quad \text{per ogni } \alpha > 0$$

Prima di definire il worst-case error, sarà necessario introdurre delle nozioni preliminari.

### 4.2.1 Reproducing kernel Hilbert space

Il seguente teorema è fondamentale dal momento in cui si lavorerà con gli spazi di Hilbert:

**Teorema 4.3** (Teorema di Riesz). *Se  $L$  è un funzionale lineare limitato su uno spazio di Hilbert  $\mathcal{H}$  allora esiste un unico elemento  $h_L \in \mathcal{H}$ , chiamato il rappresentante di  $L$ , tale che per ogni  $f \in \mathcal{H}$*

$$L(f) = \langle f, h_L \rangle_{\mathcal{H}}$$

**Definizione 4.6.** *Uno spazio di Hilbert  $\mathcal{H}$  è chiamato reproducing kernel Hilbert space se e solo se tutti i punti di valutazione dei funzionali su  $\mathcal{H}$ , denotati con  $T_y$  per ogni  $y \in D$  (dove  $D$  è il dominio della funzione) sono continui o equivalentemente limitati.*

Per il teorema di Riesz allora esiste un unico rappresentante per la valutazione di una funzione nel reproducing kernel Hilbert space. Tale rappresentante è chiamato *reproducing kernel*.

**Definizione 4.7.** *Una funzione  $K : D \times D \rightarrow \mathbb{R}$  è un reproducing kernel di uno spazio di Hilbert  $\mathcal{H}$  se e solo se per ogni  $y \in D$  e per ogni  $f \in \mathcal{H}$*

$$K(\cdot, y) = K_y \in \mathcal{H} \quad T_y(f) = f(y) = \langle f, K(\cdot, y) \rangle_{\mathcal{H}} = \langle f, K_y \rangle_{\mathcal{H}}$$

**Teorema 4.4.** *Se  $\mathcal{H}(K)$  è un reproducing kernel Hilbert space separabile allora il reproducing kernel è dato da*

$$K(x, y) = \sum_k \varphi_k(x) \overline{\varphi_k(y)} \quad (4.3)$$

dove  $\{\varphi_k\}_k$  è una qualsiasi base ortonormale per  $\mathcal{H}$

**Definizione 4.8.** *Un reproducing kernel è shift-invariante se per ogni  $\mathbf{x}, \mathbf{y}, \Delta \in [0, 1)^s$  si ha*

$$K(\{\mathbf{x} + \Delta\}, \{\mathbf{y} + \Delta\}) = K(\{\mathbf{x}, \mathbf{y}\})$$

**Teorema 4.5.** *Se un reproducing kernel è shift-invariante allora può essere scritto in termini di una sola variabile*

$$K(\{\mathbf{x}, \mathbf{y}\}) = K(K(\{\mathbf{x} - \mathbf{y}\}, \mathbf{0}) = K(\{\mathbf{x} - \mathbf{y}\})$$

Inoltre un kernel shift-invariante gode anche della proprietà di simmetria.

**Teorema 4.6.** *Dato un reproducing kernel Hilbert space  $\mathcal{H}(K)$ , si può sempre definire un kernel  $K^{shinv}$  associato a esso come*

$$K^{shinv}(\mathbf{x}, \mathbf{y}) := \int_{[0,1]^s} K(\{\mathbf{x} + \Delta\}, \{\mathbf{y} + \Delta\}) d\Delta = K^{shinv}(\{\mathbf{x} - \mathbf{y}\})$$

### 4.3 Integrazione nel contesto del worst-case error

In questa sezione si andrà a definire il worst-case error e le sue proprietà nel contesto di un reproducing kernel Hilbert space.

**Definizione 4.9.** *Il worst-case error di un algoritmo  $U$  per approssimare la soluzione  $S$  per problemi  $f \in \mathcal{F}$  è definito come*

$$e(U, \mathcal{F}) := \sup_{f \in \mathcal{F}} \|S(f) - U(f)\|$$

Nel caso di una regola di cubatura  $Q$  per l'approssimazione di un integrale  $I$  di una funzione in una palla unitaria di uno spazio di Banach  $\mathcal{B}$  è definito come

$$e(Q, \mathcal{B}) := \sup_{\substack{f \in \mathcal{B} \\ \|f\|_{\mathcal{B}} \leq 1}} \|I(f) - Q(f)\|$$

Siccome in un reproducing kernel Hilbert space le funzioni valutate sono limitate, allora anche l'integrazione e la cubatura su un dominio finito sono limitate. Questo implica che per il teorema di Riesz esistono i rappresentanti sia per  $I$  che per  $Q$ , rispettivamente  $h_I$  e  $h_Q$ , e sussiste il seguente teorema

**Teorema 4.7.** *Se  $\mathcal{H}(K)$  è un reproducing kernel Hilbert space e  $\xi_{K,Q}$  è il rappresentante dell'errore per una formula di cubatura  $Q$ , allora l'errore per le funzioni  $f \in \mathcal{H}(K)$  può essere limitato da*

$$|I(f) - Q(f)| \leq \|f\|_{\mathcal{H}} \|\xi_{K,Q}\|_{\mathcal{H}}$$

**Teorema 4.8.** *Il worst-case error quadratico per l'integrazione in  $\mathcal{H}(K)$  usando una regola di cubatura equipesata è dato da*

$$e^2(Q, K) = \int_{[0,1]^{2s}} K(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y} - \frac{2}{n} \sum_{k=0}^{n-1} \int_{[0,1]^s} K(\mathbf{x}_k, \mathbf{y}) d\mathbf{y} + \frac{1}{n^2} \sum_{k,l=0}^{n-1} K(\mathbf{x}_k, \mathbf{x}_l)$$

dove  $\mathbf{x}_k, \mathbf{x}_l \in P_n$  per  $k, l = 0, \dots, n-1$ .

**Corollario 4.2.** *Se il kernel  $K$  è shift-invariante allora il worst-case error quadratico è dato da*

$$e^2(Q, K) = - \int_{[0,1]^s} K(\mathbf{x}) d\mathbf{x} + \frac{1}{n^2} \sum_{k,l=0}^{n-1} K(\{\mathbf{x}_k - \mathbf{x}_l\})$$

**Corollario 4.3.** *Se il kernel  $K$  è shift-invariante e  $Q$  è una lattice rule, allora il worst-case error quadratico è dato da*

$$e^2(Q, K) = - \int_{[0,1]^s} K(\mathbf{x}) d\mathbf{x} + \frac{1}{n} \sum_{k=0}^{n-1} K(\{\mathbf{x}_k\})$$

Si arriva finalmente alla forma più interessante del worst-case error, che risulterà utile per la costruzione di un algoritmo component-by-component.

**Corollario 4.4.** *Se il kernel  $K$  è shift-invariante, formato da un prodotto tensoriale e  $Q$  è una lattice rule, allora il worst-case error quadratico è dato da*

$$e^2(Q, K) = - \prod_{j=1}^s \int_0^1 \eta_j(x) dx + \frac{1}{n} \sum_{k=0}^{n-1} \prod_{j=1}^s \eta_j(x_j^{(k)})$$

dove gli  $\eta_j$  sono i kernel di un reproducing kernel Hilbert spaces di dimensione uno, tali che  $K(\mathbf{x}) = \prod_{j=1}^s \eta_j(x_j)$

Per quanto riguarda il worst-case error per l'integrazione quasi-Monte Carlo, è sufficiente fare riferimento all'insieme dei punti  $P_n$  invece della regola di cubatura  $Q$ .

## 4.4 Reproducing kernel Hilbert spaces con pesi generici

Si consideri un reproducing kernel Hilbert space pesato  $\mathcal{H}(K)$ , di pesi  $\gamma_u \geq 0$  per ogni insieme  $u \subseteq \mathcal{D}_s$ , con kernel:

$$K(\mathbf{x}, \mathbf{y}) = \sum_{u \in \mathcal{D}_s} \gamma_u K_u(\mathbf{x}, \mathbf{y})$$

il quale è somma diretta di  $2^s$  reproducing kernel Hilbert spaces  $\mathcal{H}_u(K_u)$

$$\mathcal{H}(K) = \bigoplus_{u \subseteq \mathcal{D}_s} \mathcal{H}_u(K_u)$$

dove

$$K(\mathbf{x}, \mathbf{y}) = K_u(\mathbf{x}_u, \mathbf{y}_u) = \prod_{j \in u} \eta_j(x_j, y_j) \quad (4.4)$$

Si ponga  $K_\emptyset = 1$  e  $\gamma_\emptyset = 1$ . Siccome i kernels nella (4.4) sono formati da prodotti di kernels di dimensione uno, segue che gli spazi  $\mathcal{H}_u(K_u)$  sono prodotti tensoriali di spazi, mentre gli spazi  $\mathcal{H}(K)$  non sono necessariamente prodotti tensoriali di spazi.

In presenza di reproducing kernel Hilbert spaces pesato, il generico shifted worst-case error ha forma

$$e(P_n, K^{shinv}) = \left( -k_s + \frac{1}{n} \sum_{k=0}^{n-1} \sum_{u \subseteq \mathcal{D}_s} \gamma_u \prod_{j \in u} \omega(x_j^{(k)}) \right)^{1/2}$$

con

$$k_s = \sum_{u \subseteq \mathcal{D}_s} \gamma_u \prod_{j \in u} \int_0^1 \omega(x) dx$$



$$K_i^{shinv}(\mathbf{x}) = \prod_{j \in u} \omega(x_j)$$

$$\omega(x) = \int_0^1 \eta(\{x + \Delta\}, \Delta) d\Delta$$

E' conveniente avere

$$\int_0^1 \omega(x) dx = 0$$

in modo che  $k_s = \omega_\emptyset = 1$

## 4.5 Spazi di Korobov

**Definizione 4.10.** *Uno spazio di Koborov è un reproducing kernel Hilbert space il quale consiste in funzioni di periodo uno con serie di Fourier assolutamente convergenti e con reproducing kernel*

$$K(\mathbf{x}, \mathbf{y}) = \prod_{j=1}^s \left( 1 + \sum_{\substack{h \in \mathbb{Z} \\ h \neq 0}} r(h)^{-\alpha} \exp(2\pi i h(x_j - y_j)) \right)$$

Il reproducing kernel si può riscrivere come

$$K(\mathbf{x}, \mathbf{y}) = 1 + \prod_{j=1}^s \sum_{\substack{h \in \mathbb{Z} \\ h \neq 0}} \frac{\exp(2\pi i h(x_j - y_j))}{|h|^\alpha}$$

Come già menzionato quando  $\alpha$  è un intero pari, la somma infinta può essere riscritta in termini di polinomi di Bernoulli di grado  $\alpha$ ; infatti se  $P_n$  è l'insieme dei nodi di una lattice rule, allora il reproducing kernel per  $\alpha = 2$  è dato da

$$K(\mathbf{x}) = \prod_{j=1}^s (1 + 2\pi^2 B_2(x_j))$$

$$k_s = \left( 1 + 2\pi^2 \int_0^1 B_2(x) dx \right)^s = 1$$

con conseguente worst-case error

$$e(P_n, K) = \left( -1 + \frac{1}{n} \sum_{k=0}^{n-1} \prod_{j=1}^s (1 + 2\pi^2 B_2(x_j^{(k)})) \right)^{1/2}$$

Lo spazio Korobov pesato ha come reproducing kernel

$$K(\mathbf{x}, \mathbf{y}) = \sum_{u \subseteq \mathcal{D}_s} \gamma_u \prod_{j \in u} \sum_{\substack{h \in \mathbb{Z} \\ h \neq 0}} r(h)^{-\alpha} \exp(2\pi i h(x_j - y_j))$$

Questo spazio è shift-invariante e quindi

$$\omega(x) = \sum_{\substack{h \in \mathbb{Z} \\ h \neq 0}} r(h)^{-\alpha} \exp(2\pi i h x)$$

può essere espresso come polinomio di Bernoulli di grado  $\alpha$ , per ogni  $\alpha > 1$  pari.

Se  $P_n$  è un insieme di nodi per una lattice rule, allora il worst case error è dato da

$$e(P_n, K) = \left( \frac{1}{n} \sum_{k=0}^{n-1} \sum_{\substack{u \in \mathcal{D}_s \\ u \neq \emptyset}} \gamma_u \prod_{j \in u} \omega(x_j^{(k)}) \right)^{1/2} \quad (4.5)$$

## 4.6 Spazi di Sobolev

**Definizione 4.11.** *Uno spazio di Sobolev ancorato è un reproducing kernel Hilbert space il quale consiste in funzioni con derivate prime miste a quadrato integrabile. Il reproducing kernel è dato da*

$$K(\mathbf{x}, \mathbf{y}) = \prod_{j=1}^s \eta_j(x_j, y_j)$$

dove

$$\begin{aligned} \eta_j(x_j, y_j) &:= \begin{cases} 1 + \min(|x - a_j|, |y - a_j|) & \text{se } (x - a_j)(y - a_j) > 0 \\ 1 & \text{altrimenti} \end{cases} \\ &= 1 + \frac{1}{2}(|x - a_j| + |y - a_j| - |x - y|) \end{aligned}$$

dove  $\mathbf{a} \in [0, 1]^s$  è chiamato ancora

Si può ricavare che la norma di  $f$ , elemento dello spazio di Sobolev, è la versione  $L_2$  della variazione di Hardy and Krause. Allo stesso modo il worst-case error per questo spazio di funzioni è la versione  $L_2$  della discrepanza star  $D_2^*$ . Fissando  $\mathbf{a} = 1$ , la sua espressione è data prima calcolando l'associato kernel shift-invariante.

$$K^{shinv}(\{\mathbf{x} - \mathbf{y}\}) = \int_{[0,1]^s} \prod_{j=1}^s (2 - \max(\{x_j + \Delta_j\}, \{y_j + \Delta_j\})) d\Delta$$

Assumendo  $x > y$  si può ricavare che

$$\int_0^1 \max(\{x + \Delta\}, \{y + \Delta\}) d\Delta = -(x - y)^2 + (x - y) + \frac{1}{2}$$

e si ottiene che è parte del polinomio di Bernoulli di grado 2,

$B_2(x) = x^2 - x + 1/6$ . Quindi

$$K^{shinv}(\mathbf{x}) \prod_{j=1}^s \left( B_2(x_j) + \frac{4}{3} \right)$$

$$\int_{[0,1]^s} K^{shinv}(\mathbf{x}) d\mathbf{x} = \left( \frac{4}{3} \right)^s$$

ottenendo la formula per il worst-case error

$$e(P_n, K^{shinv}) = \left( - \left( \frac{4}{3} \right)^s + \frac{1}{n} \sum_{k=0}^{n-1} \prod_{j=1}^s \left( B_2(x_j^{(k)}) + \frac{4}{3} \right) \right)^{1/2}$$

Nel caso di uno spazio di Sobolev non ancorato pesato, si ha che il reproducing kernel è uguale a

$$K(\mathbf{x}, \mathbf{y}) = \sum_{u \subseteq \mathcal{D}_s} \gamma_u \prod_{j \in u} \frac{1}{2} B_2(\{x_j - y_j\}) + \left( x_j - \frac{1}{2} \right) \left( y_j - \frac{1}{2} \right)$$

con worst-case error della forma di (4.5).

## 4.7 Costruzione Component-by-Component

In questa sezione si descriverà un metodo per costruire un algoritmo component-by-component (CBC) per il worst-case error quadratico. Il contesto sarà quello di un worst-case error in un reproducing kernel Hilbert space shift-invariante. L'espressione per il worst-case error quadratico per prodotti pesati in questo spazio è data da

$$e_{s,\gamma,\beta}^2(z_1, \dots, z_s) = -\prod_{j=1}^s \beta_j + \frac{1}{n} \sum_{k=0}^{n-1} \prod_{j=1}^s \left( \beta_j + \gamma_j \omega\left(\frac{k \cdot z_j}{n}\right) \right)$$

con kernel

$$K_{s,\gamma,\beta}(\mathbf{x}) = \prod_{j=1}^s (\beta_j + \gamma_j \omega(x_j))$$

dove  $\gamma_j$  sono pesi positivi, ognuno associato a una dimensione. Il prodotto dei pesi  $\gamma_j$  è dato da una successione decrescente di pesi positivi

$$\gamma_1 \geq \gamma_2 \geq \dots \geq \gamma_s \geq 0$$

questo significa che l'importanza delle coordinate va a diminuire con l'aumento di dimensione.

Siccome l'algoritmo sarà un processo iterativo in  $s$ , la produttoria su  $s$  può essere divisa in due parti:

$$\prod_{j=1}^s \left( \beta_j + \gamma_j \omega\left(\frac{k \cdot z_j}{n}\right) \right) = \prod_{j=1}^{s-1} \left( \beta_j + \gamma_j \omega\left(\frac{k \cdot z_j}{n}\right) \right) \left( \beta_s + \gamma_s \omega\left(\frac{k \cdot z_s}{n}\right) \right)$$

Dato che all'iterazione  $s$  il prodotto di dimensione  $s-1$  è fissato, perchè  $z_j$  per  $j < s$  non cambia, allora la produttoria può essere memorizzata in un  $n$ -vettore denotato con  $p_{s-1}$

$$\prod_{j=1}^s \left( \beta_j + \gamma_j \omega\left(\frac{k \cdot z_j}{n}\right) \right) = p_{s-1}(k) \left( \beta_s + \gamma_s \omega\left(\frac{k \cdot z_s}{n}\right) \right)$$

Questo implica che l'espressione del worst-case error può essere riscritta come

$$e_s^2(z) = -\prod_{j=1}^s \beta_j + \frac{1}{n} \sum_{k=0}^{n-1} p_{s-1}(k) \left( \beta_s + \gamma_s \omega \left( \frac{k \cdot z_s}{n} \right) \right)$$

Se si vuole dare una formulazione vettoriale al worst-case error ottenuto, si definisce la matrice

$$\mathbf{\Omega}_n = \left[ \omega \left( \frac{k \cdot z}{n} \right) \right]_{\substack{z=1, \dots, n-1 \\ k=0, \dots, n-1}} \quad (\text{n numero primo})$$

che ha al più  $n$  elementi differenti che dipendono dalla funzione  $\omega$ . Una formulazione per l'errore può essere la seguente

$$\mathbf{e}_s^2 = -\bar{\beta}_s \mathbf{1}_{n-1 \times 1} + \frac{1}{n} (\beta_s \mathbf{1}_{n-1 \times n} + \gamma_s \mathbf{\Omega}_n) \mathbf{p}_{s-1}$$

dove  $\mathbf{1}_{m \times n}$  è una matrice unitaria e  $\bar{\beta}$  è la produttoria dei  $\beta_j$  fino alla dimensione  $s$ . Se si indica con  $p_s(k)$

$$p_s(k) = p_{s-1}(k) \left( \beta_s + \gamma_s \omega \left( \frac{k \cdot z_s}{n} \right) \right)$$

si definisce il kernel nel caso vettoriale come

$$\mathbf{K}_{s, \gamma, \beta} = \beta \mathbf{1}_{n-1 \times n} + \gamma \mathbf{\Omega}_n$$

Sia  $diag(\mathbf{x})$  la matrice diagonale con elementi di  $\mathbf{x}$  sulla diagonale e 0 altrove e sia  $\mathbf{v}_j$  il  $j$ -esimo vettore della matrice identica. Allora

$$\mathbf{p}_s = diag(\mathbf{v}_{z_s}^T \mathbf{K}_{n, \gamma_s, \beta_s} \mathbf{p}_{s-1}) = diag(\beta_s \mathbf{1}_{1 \times n} + \gamma_s \mathbf{v}_{z_s}^T \mathbf{\Omega}_n) \mathbf{p}_{s-1}$$

#### 4.7.1 La struttura di $\mathbf{\Omega}_n$

Si ponga

$$\omega_{z \cdot k} := \omega \left( \frac{k \cdot z}{n} \right)$$

Usando questa notazione, la struttura della matrice  $\mathbf{\Omega}_n$  è la seguente

$$\begin{aligned}\mathbf{\Omega}_n &= \begin{bmatrix} \omega_{1 \cdot 0} & \omega_{1 \cdot 1} & \omega_{1 \cdot 2} & \dots & \omega_{1 \cdot (n-1)} \\ \omega_{2 \cdot 0} & \omega_{2 \cdot 1} & \omega_{2 \cdot 2} & \dots & \omega_{2 \cdot (n-1)} \\ \omega_{(n-1) \cdot 0} & \omega_{(n-1) \cdot 1} & \omega_{(n-1) \cdot 2} & \dots & \omega_{(n-1) \cdot (n-1)} \end{bmatrix} = \\ &= \begin{bmatrix} \omega_0 & \omega_1 & \omega_2 & \dots & \omega_{(n-1)} \\ \omega_0 & \omega_2 & \omega_4 & \dots & \omega_{(n-2)} \\ \omega_0 & \omega_{(n-1)} & \omega_{(n-2)} & \dots & \omega_1 \end{bmatrix}\end{aligned}$$

**Definizione 4.12.** Una matrice circolante  $\mathbf{C}_m = \text{circ}(\mathbf{c})$  di ordine  $m$  è una matrice definita da  $m$  elementi nel vettore  $\mathbf{c}$ , indicizzati a partire da 0, come

$$[\mathbf{C}_m]_{\substack{k=0,\dots,m-1 \\ l=0,\dots,m-1}} = c_{k-l \mod m}$$

**Teorema 4.9.** Una matrice circolante  $\mathbf{C}_m$  di ordine  $m$  si può fattorizzare come

$$\mathbf{C}_m = \mathbf{F}_m^{-1} \mathbf{D} \mathbf{F}_m$$

dove  $\mathbf{F}_m$  è la matrice di Fourier di ordine  $m$  e gli autovalori sono definiti dalla trasformata discreta di Fourier della prima colonna  $\mathbf{c}$  di  $\mathbf{C}_m$

$$\mathbf{D} = \text{diag}(\mathbf{F}_m \mathbf{c})$$

**Corollario 4.5.** Un prodotto matrice vettore di un vettore  $\mathbf{x}$  con una matrice circolante  $\mathbf{C}_m$  può essere fatto in  $O(m \log n(m))$  operazioni attraverso la fattorizzazione di  $\mathbf{C}_m$

$$\mathbf{C}_m \mathbf{x} = \mathbf{F}_m^{-1} \text{diag}(\mathbf{F}_m \mathbf{c}) \mathbf{F}_m \mathbf{x}$$

Ogni trasformata discreta di Fourier viene calcolata tramite FFT (fast Fourier transformation), la quale è presente come function di Matlab.

Una fattorizzazione utile per strutture come quelle di  $\mathbf{\Omega}_n$  è quella di Rader,

che può essere generalizzata. Questa fattorizzazione è basata sulla seguente permutazione numerica (per  $n$  primo)

$$\mathbf{z} = \mathbf{\Pi}_g \mathbf{x} \iff z_l = x_k \quad \begin{cases} k = l & \text{se } l = 0 \\ k = g^{l-1} \bmod n & \text{se } l = 1, \dots, n-1 \end{cases}$$

dove  $g$  è una radice primitiva di  $n$ . La fattorizzazione di Rader nel caso della matrice  $\mathbf{\Omega}_n$  è data dal seguente teorema:

**Teorema 4.10.** *Data una matrice  $\mathbf{\Omega}_n$   $n-1 \times n$  e data una radice primitiva  $g$  di  $n$ , allora la matrice  $\mathbf{\Omega}_n$  può essere fattorizzata come*

$$\mathbf{\Omega}_n = \mathbf{\Pi}'_g \begin{bmatrix} \omega_0 \mathbf{1}_{n-1 \times 1} & \mathbf{C}_{n-1} \end{bmatrix} \mathbf{\Pi}_{g^{-1}}^T$$

Dove le permutazioni  $\mathbf{\Pi}'_g$  e  $\mathbf{\Pi}_{g^{-1}}^T$  sono definite come prima, la matrice circolante  $\mathbf{C}_{n-1}$  è definita come

$$\mathbf{C}_{n-1} = \text{circ}(\psi')$$

$$\psi := \mathbf{\Pi}_g^T \omega$$

e  $g^{-1}$  è l'inversa moltiplicativa di  $g$  modulo  $n$ .

## Capitolo 5

# Sperimentazione sul worst-case error

In questa sezione saranno mostrati diversi risultati riguardo il worst-case error quadratico su spazi di Sobolev pesati e non pesati. Il confronto tra questi due gioca un ruolo fondamentale nella parte sperimentale, siccome verranno confermati alcuni risultati teorici, in particolare sulla trattabilità. Inizialmente è stato costruito un algoritmo per il calcolo del worst-case error quadratico `fastrank1`, il quale ha in input:

- $n$ : numero di nodi
- $s$ : dimensione massima
- $\beta_s, \gamma_s$ : pesi
- $\omega(x)$  che nel caso di spazi di Sobolev è il polinomio di Bernoulli di grado 2,  $B_2(x)$

e in output:

- $z$ : vettore generatore



-  $e^2$ : worst-case error quadratico

Questo algoritmo segue la costruzione esposta nell'ultimo paragrafo dello scorso capitolo; in particolare all'interno del codice è richiamata un'altra function `primitiverts` che calcola le radici primitive partendo dal numero di nodi. Per calcolare la trasformata di Fourier e l'inversa sono stati utilizzate rispettivamente le function Matlab `fft` e `ifft`.

## 5.1 Esercizio 1

In questo esempio sono assegnati i seguenti dati di input

$$\omega(x) = x^2 - x + \frac{1}{6}; \quad n = 4001; \quad s = 10; \quad \gamma_s = \beta_s = (1, 1, \dots, 1)^T.$$

L'output ci fornisce le seguenti stime del worst-case error quadratico e dei vettori generatori al variare di  $s$ .

s=	1,	z=	1,	e2=	1.0411e-08,	e=	1.0204e-04
s=	2,	z=	1478,	e2=	5.6889e-08,	e=	2.3851e-04
s=	3,	z=	655,	e2=	2.5691e-07,	e=	5.0686e-04
s=	4,	z=	1931,	e2=	9.6625e-07,	e=	9.8298e-04
s=	5,	z=	1219,	e2=	2.5611e-06,	e=	1.6003e-03
s=	6,	z=	288,	e2=	6.1110e-06,	e=	2.4720e-03
s=	7,	z=	952,	e2=	1.2790e-05,	e=	3.5764e-03
s=	8,	z=	409,	e2=	2.3619e-05,	e=	4.8599e-03
s=	9,	z=	770,	e2=	4.2835e-05,	e=	6.5448e-03
s=	10,	z=	1581,	e2=	7.2737e-05,	e=	8.5286e-03

Fornendo invece in input:

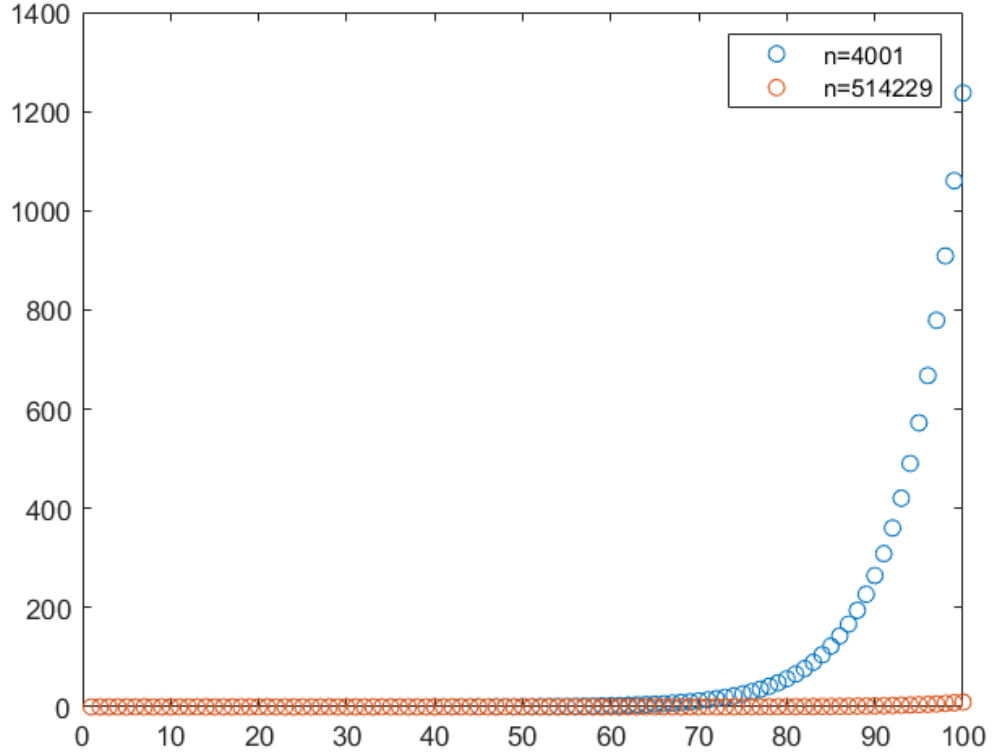
$$\omega(x) = x^2 - x + \frac{1}{6}; \quad n = 514229; \quad s = 10; \quad \gamma_s = \beta_s = (1, 1, \dots, 1)^T.$$

con  $n$  numero primo appartenente alla successione di Fibonacci.

I risultati ottenuti in questo caso sono:

s=	1,	z=130056,	e2=6.3016e-13,	e=7.9383e-07
s=	2,	z= 14625,	e2=4.4589e-12,	e=2.1116e-06
s=	3,	z=237909,	e2=3.2413e-11,	e=5.6933e-06
s=	4,	z=145974,	e2=1.8017e-10,	e=1.3423e-05
s=	5,	z=233669,	e2=7.2093e-10,	e=2.6850e-05
s=	6,	z=103030,	e2=2.6179e-09,	e=5.1165e-05
s=	7,	z=229770,	e2=7.0158e-09,	e=8.3760e-05
s=	8,	z= 40654,	e2=1.7048e-08,	e=1.3057e-04
s=	9,	z=210789,	e2=3.6647e-08,	e=1.9143e-04
s=	10,	z= 46204,	e2=7.1632e-08,	e=2.6764e-04

Se si plotta l'errore in funzione della dimensione, per  $s = 100$  si ottiene il seguente grafico:



Si nota che la successione degli errori corrispondente al numero di Fibonacci cresce molto più lentamente della successione degli errori per  $n = 4001$ .

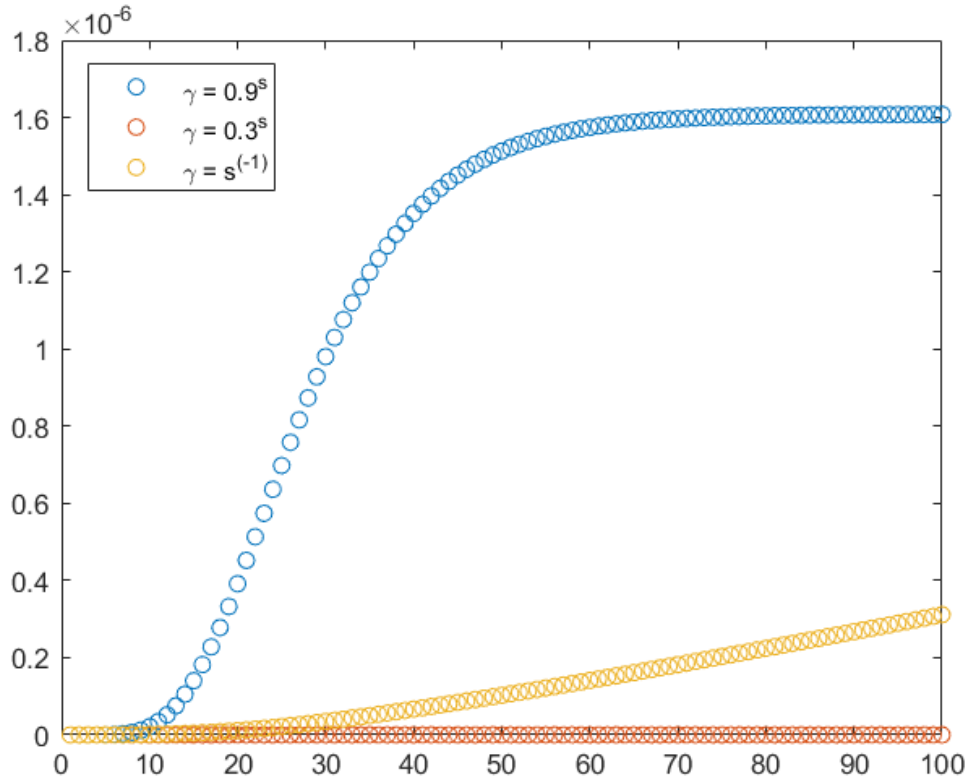
## 5.2 Confronto tra il caso pesato e il caso non pesato in spazi di Sobolev

Nel paragrafo precedente sono stati utilizzati pesi unitari. Oggetto di questa sezione sarà il confronto tra i risultati ottenuti nel caso precedente e quelli che invece si ottengono andando ad attribuire a  $\beta_s$  e  $\gamma_s$  valori diversi da uno. In particolare, come già osservato nella parte teorica, verranno attribuiti

valori ai pesi (che dovranno essere decrescenti) in modo tale che le prime componenti abbiano un'importanza maggiore rispetto a quelle corrispondenti ad una dimensione più alta. Questo porterà ad avere la convergenza del worst-case error quadratica in quanto verranno controllati con pesi sempre bassi i quali "attenueranno" la crescita della dimensionalità. Inoltre in spazi di Sobolev multidimensionali si ottiene che in queste ipotesi, se naturalmente la somma dei pesi è finita, l'ordine di convergenza è di  $O(n^{-1+\delta})$  per ogni  $\delta > 0$ .

## 5.3 Esercizio 2

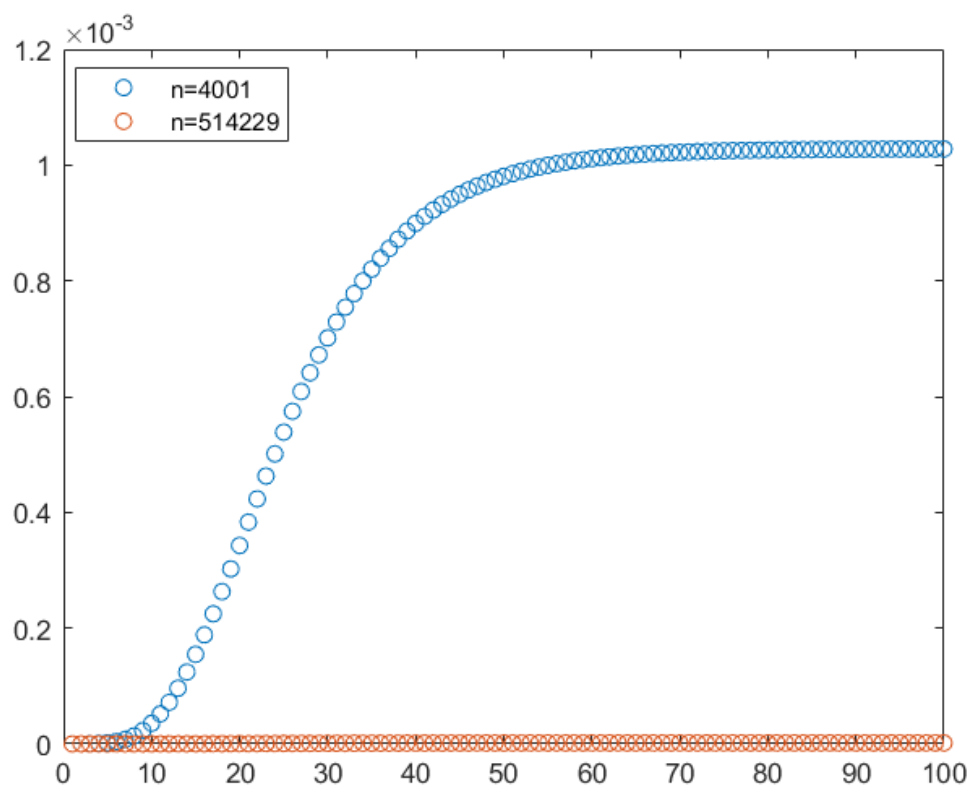
In questo esempio verrà confrontato il worst-case error in funzione della dimensione  $s = 100$  per  $n = 514229$  ma con pesi uguali a  $\gamma_{1s} = 0.9^s$ ,  $\gamma_{2s} = 0.3^s$  e  $\gamma_{3s} = s^{-1}$  con ordine di convergenza uguale rispettivamente a  $10^{-6}$ ,  $10^{-12}$ ,  $10^{-7}$  e  $\beta_s = 1 + \frac{\gamma_s}{3}$



## 5.4 Esercizio 3

In questo esempio si confronterà nuovamente il worst-case error in funzione della dimensione  $s$  per  $n = 4001$  e  $n = 514229$  ma con pesi non unitari decrescenti. Assegnati come dati di input:

$$\omega(x) = x^2 - x + \frac{1}{6}; \quad s = 100; \quad \gamma_s = 0.9^s \quad \beta_s = 1 + \frac{\gamma_s}{3}.$$



il primo con ordine di convergenza di  $10^{-3}$ , il secondo di  $10^{-6}$ .

# Conclusioni

Analizzando il lavoro svolto nel corpo del progetto, l'idea di base per l'integrazione multi-dimensionale nell'ambito dell'analisi numerica è la generalizzazione della regola "classica" dei rettangoli. Alla luce dei risultati ottenuti e, compatibilmente con quanto ci si aspettava dalla teoria, si evince come la scelta del rappresentante del dominio di integrazione e soprattutto la distribuzione dei punti al suo interno, giochino un ruolo chiave in questo contesto. In particolare, dopo aver scelto di rappresentare un qualunque dominio di integrazione multi-dimensionale ricorrendo al cubo  $[0, 1]^s$ , si è visto come alcune successioni di punti diano risultati migliori rispetto ad altre, è stato quindi necessario introdurre il concetto di discrepanza di un insieme e quello di successioni a bassa discrepanza che, utilizzate nell'algoritmo per l'integrazione con i metodi quasi-Monte Carlo hanno prodotto errori piccoli e maggiorabili con quantità dipendenti esclusivamente dalle caratteristiche della funzione integranda e dalla distribuzione dei punti all'interno del dominio di integrazione.

Si è osservato poi come sia possibile generalizzare ancora di più le regole di integrazione quasi Monte-Carlo rivedendole come delle rank 1 lattice rules in spazi specifici che sono stati abbondantemente descritti. L'analisi sul worst-case error invece ha messo in evidenza come le successioni dell'errore al variare della dimensione possano giungere a convergenza nel momento in

cui si passi al caso pesato.



# Bibliografia

- [1] Hayakawa, S. *Monte Carlo cubature construction*. Japan J. Indust. Appl. Math. 38, 561–577 (2021). <https://doi.org/10.1007/s13160-020-00451-x>
- [2] Bittante, C., De Marchi, S., Elefante, G.: *A new quasi-Monte Carlo technique based on nonnegative least squares and approximate Fekete points*. Numer. Math. Theory Methods Appl. 9(4), 640–663 (2016)
- [3] Dirk Nuyens, Ronald Cools, *Fast component-by-component construction of rank-1 lattice rules with a non-prime number of points*, Journal of Complexity, Volume 22, Issue 1, 2006, Pages 1-8, (<https://www.sciencedirect.com/science/article/pii/S0885064X05000658>)
- [4] Josef Dick , Peter Kritzer , Friedrich Pillichshammer *Lattice Rules: Numerical Integration, Approximation, and Discrepancy*. Springer Series in Computational Mathematics (SSCM, volume 58)