



UNIVERSITÀ DEGLI STUDI DI PADOVA  
DIPARTIMENTO DI MATEMATICA

TESI PER LA LAUREA TRIENNALE  
IN MATEMATICA

# Quadratura di Fejér e Clenshaw-Curtis via DCT e DST, e sue applicazioni

*Candidato:*  
Deniz Montagner

*Relatore:*  
Prof. Alvise Sommariva

Anno accademico 2011-2012



# Indice

<b>Introduzione</b>	<b>5</b>
<b>1 Regole di quadratura</b>	<b>7</b>
1.1 Regola di Fejér del primo tipo . . . . .	7
1.1.1 Caso della funzione peso simmetrica . . . . .	9
1.2 Regola di Fejér del secondo tipo . . . . .	10
1.2.1 Caso della funzione peso simmetrica . . . . .	12
1.2.2 Momenti di Chebyshev di secondo tipo . . . . .	14
1.3 Regola di Clenshaw-Curtis . . . . .	15
1.3.1 Calcolo dei momenti di Chebyshev . . . . .	18
1.4 Grado di precisione . . . . .	19
<b>2 Implementazione in Matlab</b>	<b>21</b>
2.1 Regola di Fejér di primo tipo . . . . .	21
2.2 Regola di Fejér di secondo tipo . . . . .	23
2.3 Regola di Clenshaw-Curtis . . . . .	24
2.4 Esempio: funzione peso di Gegenbauer . . . . .	25
<b>3 Applicazioni alla quadratura trigonometrica</b>	<b>27</b>
3.1 Quadratura trigonometrica subperiodica . . . . .	27
3.2 Cubatura su lenti simmetriche . . . . .	32
<b>4 Appendice: codici usati nei test</b>	<b>41</b>
4.1 Funzione peso di Gegenbauer . . . . .	41
4.2 Quadratura trigonometrica . . . . .	43



# Introduzione

In questo lavoro viene dapprima descritto un metodo per il calcolo dei nodi e dei pesi per la quadratura numerica pesata con formule interpolatorie non gaussiane di tipo Fejér e Clenshaw-Curtis. Il principale vantaggio di tali regole consiste nel fatto che i nodi sono noti esplicitamente in quanto punti di Chebyshev, mentre i relativi pesi sono calcolati esplicitamente utilizzando la *Discrete Cosine Transform* (DCT) e la *Discrete Sine Transform* (DST).

In virtù del fatto che le DCT e le DST possono essere calcolate velocemente con algoritmi dedicati, tali operazioni sono particolarmente convenienti ed il calcolo dei pesi rapido rispetto alle corrispettive formule gaussiane.

L'utilizzo di queste trasformate per il calcolo dei pesi della regola di Clenshaw-Curtis era già stato considerato nel 1972 da W. Gentleman in [11] nel caso in cui  $w(x) \equiv 1$ . Nel corso degli anni sono state descritte alcune proprietà di convergenza (cfr. [17]) e stime dell'errore (cfr. [18]) per funzioni peso  $w$  per cui  $\int_{-1}^1 |w(x)|^p dx < +\infty$  (per qualche  $p > 1$ ). Eppure, nonostante quanto descritto da Gentleman, il calcolo dei pesi di tali formule di quadratura con DCT e DST non ha ricevuto per anni la dovuta attenzione.

L'interesse per la regola di Clenshaw-Curtis (nel caso del peso di Legendre  $w(x) \equiv 1$ ) è rinato alla luce delle considerazioni di L. Trefethen, che in [19] ha mostrato che essa ha spesso una *performance* paragonabile alle regole gaussiane in termini di precisione.

Nel 2006, J. Waldvogel ha dimostrato in [20] che è possibile determinare le formule di Fejér e di Clenshaw-Curtis tramite la *Fast Fourier Transform* e la sua inversa, considerando però solamente il caso della funzione peso di Legendre.

In questo lavoro, verrà mostrato come calcolare formule di quadratura di Fejér e Clenshaw-Curtis relativamente ad una arbitraria funzione peso, purché siano noti i suoi momenti pesati di Chebyshev di primo tipo.

Esperimenti numerici hanno mostrato che il nuovo metodo è vantaggioso sul piano della velocità di calcolo e della stabilità.

Di seguito, verranno mostrate alcune applicazioni alla quadratura trigonometrica. I risultati mostrano che mentre si è al momento in grado di calcolare formule trigonometriche di grado inferiore alle migliaia via formule gaussiane, con queste nuove regole si possono calcolare formule di quadratura aventi gradi di precisione superiori al milione.

Infine ci interesseremo alla determinazione di formule di cubatura sulle lenti circolari simmetriche. Introdurremo opportune formule prodotto e ne mostreremo le performance su particolari funzioni test, evidenziando quali problemi numerici affrontare per rendere il calcolo di tali formule ancora più efficiente.

# Capitolo 1

## Regole di quadratura

Si consideri una funzione  $f$  di  $[-1, 1]$  in  $\mathbb{R}$  e si supponga di voler calcolare l'integrale definito

$$\int_{-1}^1 f(x)w(x)dx$$

L'approssimazione suggerita da un metodo di quadratura è:

$$\int_{-1}^1 f(x)w(x)dx \approx \sum_{k=1}^n w_k f(x_k) \quad (1.1)$$

ove  $\{x_1, \dots, x_k, \dots, x_n\}$  è l'insieme dei nodi e  $\{w_1, \dots, w_k, \dots, w_n\}$  è l'insieme dei pesi.

Sono note numerose possibili scelte per i nodi e i corrispondenti pesi, ma non tutte sono ugualmente soddisfacenti. Nelle sezioni successive saranno prese in considerazione tre cosiddette *regole* per la scelta di nodi e pesi.

### 1.1 Regola di Fejér del primo tipo

La regola di Fejér del primo tipo, introdotta nel 1933 dall'ungherese L. Fejér in [7] è data dagli  $n$  nodi di Chebyshev in  $(-1, 1)$ :

$$x_k = \cos(\theta_k) \quad \text{con} \quad \theta_k = \frac{(2k-1)\pi}{2n} \quad k = 1, \dots, n \quad (1.2)$$

Poiché la regola di Fejér è interpolatoria, si può dimostrare che, data una funzione peso  $w : (-1, 1) \rightarrow \mathbb{R}$ , ciascun peso vale (cfr. [6, eq. (2.5.5.3)]):

$$w_k = \frac{-2}{T'_n(x_k)T_{n+1}(x_k)} \left( \frac{\gamma_0}{2} + \sum_{m=1}^{n-1} \gamma_m T_m(x_k) \right) \quad k = 1, \dots, n \quad (1.3)$$

ove:

$$\begin{aligned} T_m(x) &:= \cos(m \arccos(x)) \\ \gamma_m &= \int_{-1}^1 T_m(x) w(x) dx \end{aligned}$$

$\gamma_m$  è il momento pesato di  $T_m$ , ovvero dell' $m$ -esimo polinomio di Chebyshev di primo tipo. Ricordiamo che gli  $\{x_k\}_{k=1,\dots,n}$  sopra definiti sono gli zeri di  $T_n$ . Inoltre si verifica facilmente che  $T_n$  è un polinomio di grado  $n$  (cfr. [5, §3.3]).

È noto che:

$$T'_n(x_k) = T'_n(\cos \theta_k) = -\frac{-n \sin(n \arccos(x_k))}{\sqrt{1 - x_k^2}} = \frac{n \sin(n \theta_k)}{\sin \theta_k} = \frac{(-1)^{k-1} n}{\sin \theta_k}$$

e che:

$$T_{n+1}(x_k) = \cos((n+1)\theta_k) = \cos\left(k\pi - \frac{\pi}{2}\right) \cos(\theta_k) - \sin\left(k\pi - \frac{\pi}{2}\right) \sin(\theta_k) = (-1)^k \sin \theta_k$$

Allora da una semplice sostituzione si ha che i pesi sono:

$$w_k = \frac{2}{n} \left( \frac{\gamma_0}{2} + \sum_{m=1}^{n-1} \gamma_m T_m(x_k) \right) = \quad (1.4)$$

$$= \frac{1}{n} \left( \gamma_0 + 2 \sum_{m=1}^{n-1} \gamma_m T_m(x_k) \right) \quad (1.5)$$

Sostituendo, si ottiene:

$$w_k = \frac{1}{n} \left( \gamma_0 + 2 \sum_{m=1}^{n-1} \gamma_m \cos\left(m \frac{\pi(2k-1)}{2n}\right) \right) \quad k = 1, \dots, n \quad (1.6)$$

Si può mostrare che quanto scritto al membro destro dell'equazione (1.6) è il risultato di una *Discrete Cosine Transform* di terzo tipo (abbreviata in DCT-III). Tale trasformata mappa un insieme di  $n$  valori  $\{y_0, \dots, y_{n-1}\}$  in altrettanti  $\{\hat{y}_0^{(III)}, \dots, \hat{y}_{n-1}^{(III)}\}$ , ove:

$$\hat{y}_j^{(III)} = \frac{y_0}{2} + \sum_{m=1}^{n-1} y_m \cos\left(\frac{\pi m}{n} \left(j + \frac{1}{2}\right)\right) \quad k = 1, \dots, n \quad (1.7)$$

Sostituendo agli  $\{y_i\}_{i=0,\dots,n-1}$  nella definizione della DCT-III i momenti pesati  $\{\gamma_i\}_{i=0,\dots,n-1}$ , risulta:

$$w_j^* := w_{j+1} = \frac{2}{n} \left( \frac{\gamma_0}{2} + \sum_{m=1}^{n-1} \gamma_m \cos \left( m \frac{\pi(2j+1)}{2n} \right) \right) \quad j = 0, \dots, n-1 \quad (1.8)$$

ovvero:

$$w_j^* = \frac{2}{n} \hat{\gamma}_j^{(III)} \quad j = 0, \dots, n-1$$

In conclusione, applicando una DCT-III ai momenti pesati  $\{\gamma_i\}_{i=0,\dots,n-1}$  dei polinomi di Chebyshev si ottengono i nodi di Fejér del primo tipo.

### 1.1.1 Caso della funzione peso simmetrica

Se la funzione peso  $w$  è simmetrica, il calcolo è ancora più semplice e veloce, poiché i momenti pesati di indice dispari, ovvero i  $\gamma_{2k+1}$  sono nulli e i pesi sono chiaramente simmetrici ( $w_k = w_{n-k+1}$ ). Quindi è necessario effettivamente calcolare i nodi  $w_j$  per  $j = 0, \dots, \lfloor \frac{n-1}{2} \rfloor$ . Ponendo  $N := \lfloor \frac{n-1}{2} \rfloor + 1$ , si ottiene:

$$w_j^* = \frac{2}{n} \left( \frac{\gamma_0}{2} + \sum_{m=1}^{\lfloor (n-1)/2 \rfloor} \gamma_{2m} \cos \left( m \frac{\pi(2j+1)}{n} \right) \right) \quad (1.9)$$

$$= \frac{2}{n} \left( \frac{\gamma_0}{2} + \sum_{m=1}^{N-1} \gamma_{2m} \cos \left( m \frac{\pi(2j+1)}{n} \right) \right) \quad j = 0, \dots, N-1 \quad (1.10)$$

**Caso 1:  $n$  pari.** Se  $n$  è pari, allora  $n = 2N$  e la formula dei pesi è quindi:

$$w_j^* = \frac{2}{n} \left( \frac{\gamma_0}{2} + \sum_{m=1}^{N-1} \gamma_{2m} \cos \left( m \frac{\pi(2j+1)}{2N} \right) \right) \quad j = 0, \dots, N-1 \quad (1.11)$$

Applicando la definizione (1.7) della DCT-III ai momenti pari  $\{\gamma_{2m}\}_{m=1,\dots,N-1}$  si ottiene:

$$\hat{\gamma}_{2j}^{(III)} = \frac{\gamma_0}{2} + \sum_{m=1}^{N-1} \gamma_{2m} \cos \left( \frac{\pi m}{N} \left( j + \frac{1}{2} \right) \right) \quad j = 0, \dots, N-1 \quad (1.12)$$

che, confrontata con l'equazione (1.11), dà:

$$w_j^* = \frac{2\hat{\gamma}_{2j}^{(III)}}{n} = \frac{\hat{\gamma}_{2j}^{(III)}}{N} \quad j = 0, \dots, N-1 \quad (1.13)$$

**Caso 2:  $n$  dispari.** Se invece  $n$  è dispari, allora  $N = \frac{n+1}{2}$ , ovvero  $n = 2N - 1$ . Pertanto un generico peso  $w_j^*$  è:

$$w_j^* = \frac{2}{2N-1} \left( \frac{\gamma_0}{2} + \sum_{m=1}^{N-1} \gamma_{2m} \cos \left( m \frac{\pi(j + \frac{1}{2})}{N - \frac{1}{2}} \right) \right) \quad j = 0, \dots, N-1 \quad (1.14)$$

Per calcolare questi pesi si può utilizzare la *Discrete Cosine Transform* di settimo tipo (DCT-VII), la quale mappa  $\{y_0, \dots, y_{n-1}\}$  in  $\{\hat{y}_0^{(VII)}, \dots, \hat{y}_{n-1}^{(VII)}\}$ , che sono definite in questo modo:

$$\hat{y}_j^{(VII)} := \frac{y_0}{2} + \sum_{m=1}^{n-1} y_m \cos \left( \frac{\pi(j + \frac{1}{2})m}{n - \frac{1}{2}} \right) \quad (1.15)$$

Quindi si applica la DCT-VII ai momenti pesati pari  $\{\gamma_{2m}\}_{m=1, \dots, N-1}$  ottenendo:

$$\hat{\gamma}_{2j}^{(VII)} := \frac{\gamma_0}{2} + \sum_{m=1}^{N-1} \gamma_m \cos \left( \frac{\pi(j + \frac{1}{2})m}{N - \frac{1}{2}} \right) \quad (1.16)$$

e infine:

$$w_j^* = \frac{2\hat{\gamma}_{2j}^{(VII)}}{n} = \frac{2\hat{\gamma}_{2j}^{(VII)}}{2N-1} \quad j = 0, \dots, N-1 \quad (1.17)$$

## 1.2 Regola di Fejér del secondo tipo

Si definisce il polinomio di Chebyshev del secondo tipo di grado  $n$ :

$$U_n(x) = \frac{\sin((n+1)\arccos(x))}{\sin(\arccos(x))} \quad (1.18)$$

che ha come zeri i punti  $x_k = \cos(\theta_k)$  con  $\theta_k = \frac{k\pi}{n+1}$ ,  $k = 1, \dots, n$ . Tali zeri sono i nodi della regola di Fejér di secondo tipo (con  $n$  punti), introdotta da Fejér nel 1933 in [8]. Si può dimostrare facilmente che  $U_n$  è effettivamente un polinomio di grado  $n$ .

Data una funzione peso  $w : (-1, 1) \rightarrow \mathbb{R}$ , si vuole trovare i pesi  $w_k$  corrispondenti ai nodi  $x_k$ , come è stato fatto nel caso della regola di Fejér di primo tipo. Si richiama a tale scopo la formula di Christoffel-Darboux [10, §1.3.3], valida per i polinomi ortogonali come  $U_n$ :

$$\sum_{j=0}^n \frac{U_j(x)U_j(y)}{\|U_j\|} = \frac{2^n}{2^{n+1}\|U_n\|} \cdot \frac{U_n(y)U_{n+1}(x) - U_{n+1}(y)U_n(x)}{x-y} \quad (1.19)$$

dove  $2^n$  e  $2^{n+1}$  sono rispettivamente i coefficienti direttori di  $U_n$  e  $U_{n+1}$ . Ponendo  $y = x_k$ , si ricava:

$$U_n(x) = \frac{2(x_k - x)}{U_{n+1}(x_k)} \sum_{j=0}^n U_j(x_k) U_j(x) \quad (1.20)$$

È inoltre noto che, poiché le regole di Fejér sono interpolatorie, vale (cfr. [6, §2.5.5]):

$$w_k = \frac{1}{U'_n(x_k)} \int_{-1}^1 \frac{U_n(x) w(x)}{x - x_k} dx \quad (1.21)$$

Sostituendo il risultato della formula di Christoffel-Darboux e ponendo  $m_s := \int_{-1}^1 U_s(x) w(x) dx$ , si ottiene:

$$w_k = \frac{-2}{U'_n(x_k) U_{n+1}(x_k)} \sum_{s=0}^n U_s(x_k) \int_{-1}^1 U_s(x) w(x) dx = \quad (1.22)$$

$$= \frac{-2}{U'_n(x_k) U_{n+1}(x_k)} \sum_{s=0}^n U_s(x_k) m_s \quad (1.23)$$

Ora, si noti che:

$$T_{n+1}(x_k) = \cos((n+1) \arccos(\cos \theta_k)) = \cos((n+1)\theta_k)$$

e che:

$$\begin{aligned} U_{n+1}(x_k) &= \frac{\sin((n+2) \arccos(\cos \theta_k))}{\sin(\arccos(\cos \theta_k))} = \\ &= \frac{\sin((n+2)\theta_k)}{\sin(\theta_k)} = \\ &= \frac{\sin((n+1)\theta_k) \cos(\theta_k) + \sin(\theta_k) \cos((n+1)\theta_k)}{\sin(\theta_k)} = \\ &= \cos((n+1)\theta_k) = T_{n+1}(x_k) \end{aligned}$$

il che è giustificato dal fatto che  $\sin((n+1)\theta_k) = \sin((n+1)\frac{k\pi}{n+1}) = 0$ . Inoltre, essendo  $U_n(x_k) = 0$ ;  $T_{n+1}^2(x_k) = 1$ , risulta:

$$U'_n(x) = \frac{(n+1)T_{n+1}(x) - xU_n(x)}{x^2 - 1} \quad (1.24)$$

e allora:

$$\frac{1}{U'_n(x_k) U_{n+1}(x_k)} = \frac{x_k^2 - 1}{(n+1)T_{n+1}^2(x_k)} = \frac{x_k^2 - 1}{n+1} \quad (1.25)$$

È ora possibile ottenere una formula esplicita per  $w_k$ :

$$w_k = \frac{2(1 - x_k^2)}{n+1} \sum_{s=0}^n U_s(x_k) m_s = \quad (1.26)$$

$$= \frac{2(1 - \cos^2(\theta_k))}{n+1} \sum_{s=0}^n \frac{\sin((s+1)\theta_k)}{\sin(\theta_k)} m_s = \quad (1.27)$$

$$= \frac{2 \sin(\theta_k)}{n+1} \sum_{s=0}^n \sin((s+1)\theta_k) m_s = \quad (1.28)$$

$$= \frac{2 \sin(\theta_k)}{n+1} \sum_{s=0}^{n-1} \sin((s+1)\theta_k) m_s \quad (1.29)$$

poiché  $\sin((n+1)\theta_k) = 0$ , come visto in precedenza.

Si ripete ora lo slittamento degli indici, come effettuato nella sezione precedente. Ponendo infatti  $w_j^* = w_{j+1} = w_k$  si ha:

$$w_j^* = \frac{2 \sin(\theta_{j+1})}{n+1} \sum_{s=0}^{n-1} \sin((s+1)\theta_{j+1}) m_s \quad (1.30)$$

$$= \frac{2 \sin(\theta_{j+1})}{n+1} \sum_{s=0}^{n-1} \sin\left(\frac{\pi(s+1)(j+1)}{n+1}\right) m_s \quad j = 0, \dots, n-1 \quad (1.31)$$

A questo punto risulta di fondamentale utilità la *Discrete Sine Transform* di primo tipo (DST-I), che mappa un insieme  $\{y_0, \dots, y_{n-1}\}$  in un  $\{\hat{y}_0^{(I)}, \dots, \hat{y}_{n-1}^{(I)}\}$ , i cui elementi sono così definiti:

$$\hat{y}_j^{(I)} := \sum_{s=0}^{n-1} y_s \sin\left(\frac{\pi(s+1)(j+1)}{n+1}\right) \quad j = 0, \dots, n-1 \quad (1.32)$$

Pertanto i pesi della regola di Fejér di secondo tipo sono:

$$w_j^* = \frac{2 \sin(\theta_{j+1})}{n+1} \hat{m}_j^{(I)} \quad j = 0, \dots, n-1 \quad (1.33)$$

### 1.2.1 Caso della funzione peso simmetrica

Sia la funzione peso  $w$  simmetrica per ipotesi; allora i momenti dei polinomi  $U_{2j+1}$  sono nulli, poiché i polinomi di Chebyshev di secondo tipo sono funzioni dispari.

Ricordando poi che  $w_j = w_{n-k+1}$ , il problema si riduce al calcolo di soli  $\lfloor \frac{n-1}{2} \rfloor + 1 =: N$  pesi. In particolare:

$$w_j^* = \frac{2 \sin(\theta_{j+1})}{n+1} \sum_{s=0}^{N-1} \sin\left(\frac{\pi(2s+1)(j+1)}{n+1}\right) m_{2s} \quad j = 0, \dots, N-1 \quad (1.34)$$

**Caso 1:  $n$  pari.** Se  $n$  è pari, allora  $n = 2N$  e la formula dei pesi è quindi:

$$w_j^* = \frac{2 \sin(\theta_{j+1})}{2N+1} \sum_{s=0}^{N-1} \sin\left(\frac{\pi(2s+1)(j+1)}{2N+1}\right) m_{2s} \quad j = 0, \dots, N-1 \quad (1.35)$$

Questa si può calcolare tramite la DST-VI, che mappa un insieme  $\{y_0, \dots, y_{n-1}\}$  in un  $\{\hat{y}_0^{(VI)}, \dots, \hat{y}_{n-1}^{(VI)}\}$  definito da:

$$\hat{y}_j^{(VI)} := \sum_{s=0}^{n-1} y_s \sin\left(\frac{\pi(s+\frac{1}{2})(j+1)}{n+\frac{1}{2}}\right) \quad j = 0, \dots, n-1 \quad (1.36)$$

Quindi, applicando la DST-VI ai momenti  $\{m_{2j}\}_{j=0, \dots, N-1}$ , si ha:

$$\hat{m}_{2j}^{(VI)} = \sum_{s=0}^{N-1} m_{2s} \sin\left(\frac{\pi(s+\frac{1}{2})(j+1)}{N+\frac{1}{2}}\right) \quad j = 0, \dots, N-1 \quad (1.37)$$

da cui si ottiene:

$$w_j^* = \frac{2 \sin(\theta_{j+1})}{2N+1} \hat{m}_{2j}^{(VI)} \quad j = 0, \dots, N-1 \quad (1.38)$$

**Caso 2:  $n$  dispari.** Se invece  $n$  è dispari, allora  $N = \frac{n+1}{2}$ , ovvero  $n = 2N - 1$ . Pertanto un generico peso  $w_j^*$  è:

$$w_j^* = \frac{2 \sin(\theta_{j+1})}{2N-1} \sum_{s=0}^{N-1} \sin\left(\frac{\pi(s+\frac{1}{2})(j+1)}{N}\right) m_{2s} \quad j = 0, \dots, N-1 \quad (1.39)$$

Questa volta si utilizza la DST-II. Essa mappa  $\{y_0, \dots, y_{n-1}\}$  in  $\{\hat{y}_0^{(II)}, \dots, \hat{y}_{n-1}^{(II)}\}$  definiti da:

$$\hat{y}_j^{(II)} := \sum_{s=0}^{n-1} y_s \sin\left(\frac{\pi(s+\frac{1}{2})(j+1)}{n}\right) \quad j = 0, \dots, n-1 \quad (1.40)$$

Dunque la DST-II mappa i momenti  $\{m_{2j}\}_{j=0,\dots,N-1}$  in:

$$\hat{m}_{2j}^{(II)} := \sum_{s=0}^{N-1} m_{2s} \sin\left(\frac{\pi(s + \frac{1}{2})(j+1)}{N}\right) \quad j = 0, \dots, N-1 \quad (1.41)$$

e pertanto si ottiene che un generico peso  $w_j^*$  si esprime come:

$$w_j^* = \frac{\hat{m}_{2j}^{(II)} \sin(\theta_{j+1})}{N} \quad j = 0, \dots, N-1 \quad (1.42)$$

### 1.2.2 Momenti di Chebyshev di secondo tipo

È noto che (cfr. [16, §1.2]):

$$U_k(x) = \begin{cases} 2 \sum_{j \text{ dispari}}^k T_j(x) & \text{se } k \text{ è dispari} \\ 2 \sum_{j \text{ pari}}^k T_j(x) - 1 & \text{se } k \text{ è pari} \end{cases} \quad (1.43)$$

Da questo si ricava che, se  $k$  è dispari:

$$m_k = \int_{-1}^1 U_k(x) w(x) dx = 2 \sum_{j \text{ dispari}}^k \gamma_j \quad (1.44)$$

ovvero i momenti di Chebyshev del secondo tipo di indice dispari sono esprimibili in funzione di quelli di primo tipo. Lo stesso principio vale per quelli di indice pari:

$$m_k = \int_{-1}^1 U_k(x) w(x) dx = 2 \sum_{j \text{ pari}}^k \gamma_j - \gamma_0 \quad (1.45)$$

dove si è usato il fatto che  $\gamma_0 = \int_{-1}^1 w(x) dx$ .

Questo risultato verrà utilizzato nell'implementazione in Matlab.

### 1.3 Regola di Clenshaw-Curtis

I nodi della regola di Clenshaw-Curtis sono gli  $n - 2$  nodi di Fejér di secondo tipo, più i punti  $-1$  e  $+1$ , estremi dell'intervallo in oggetto:

$$x_k = \cos\left(\frac{\pi(k-1)}{n-1}\right) \quad k = 1, \dots, n \quad (1.46)$$

Ponendo  $P_n(x) := (1 - x^2)U_{n-2}(x)$ , cioè definendolo come il polinomio di  $n$ -esimo grado che ha come zeri gli  $n$  punti di Clenshaw-Curtis, allora si ha, come conseguenza della formula di interpolazione:

$$w_j = \frac{1}{P'_n(x_j)} \int_{-1}^1 \frac{P_n(x)w(x)}{x - x_j} dx \quad (1.47)$$

$$= \frac{1}{(1 - x_j^2)U'_{n-2}(x_j)} \int_{-1}^1 \frac{(1 - x^2)U_{n-2}(x)w(x)}{x - x_j} dx \quad (1.48)$$

$$= \frac{1}{(1 - x_j^2)U'_{n-2}(x_j)} \int_{-1}^1 \frac{U_{n-2}(x)w^*(x)}{x - x_j} dx \quad (1.49)$$

dove  $w^*(x) := (1 - x^2)w(x)$ . Ora, ricordando che i pesi della regola di Fejér di secondo tipo (su  $n - 2$  nodi) sono:

$$w_k^{F2} = \frac{1}{U'_{n-2}(x_k)} \int_{-1}^1 \frac{U_{n-2}(x)w^*(x)}{x - x_k} dx \quad k = 1, \dots, n - 2 \quad (1.50)$$

allora i pesi di Clenshaw-Curtis sono, per  $j = 2, \dots, n - 1$ :

$$w_j = \frac{w_{j-1}^{F2}}{1 - x_j^2} \quad (1.51)$$

Si vuole ora calcolare i pesi  $w_1$ , relativo a  $x_1 = 1$ , e  $w_n$ , relativo a  $x_n = -1$ . Si utilizza il fatto che  $P'(1) = -2U_{n-2}(1)$  e che  $U_{n-2}(1) = n - 1$ , ottenendo:

$$\frac{P_n(x)}{P'_n(1)} = \frac{(1 - x^2)U_{n-2}(x)}{-2U_{n-2}(1)} = \frac{(1 - x^2)U_{n-2}(x)}{-2(n - 1)} \quad (1.52)$$

da cui si ha:

$$w_1 = \frac{1}{P'_n(1)} \int_{-1}^1 \frac{P_n(x)w(x)}{x - 1} dx = \frac{1}{2(n - 1)} \int_{-1}^1 (1 + x)U_{n-2}(x)w(x) dx \quad (1.53)$$

L'integrale è calcolabile a partire dai momenti di Chebyshev (cfr. la sezione 1.2.2). Si considerano ora i due casi, a seconda della parità di  $n - 2$ .

**Caso 1:  $n$  pari.** Sono noti i seguenti fatti:

$$T_0(x) = 1 \quad T_1(x) = xT_0(x)$$

$$U_k(x) = 2 \sum_{j \text{ pari}}^k T_j(x) - 1 = T_0(x) + 2 \sum_{j=1}^{k/2} T_{2j}(x)$$

$$xT_j(x) = \frac{T_{j+1}(x) + T_{j-1}(x)}{2} \quad j \geq 1$$

dove l'ultimo risultato deriva da  $\cos(n\theta) \cos(m\theta) = \frac{\cos((n+m)\theta) + \cos(|n-m|\theta)}{2}$  (cfr. [16, §1.1]). Dunque risulta:

$$\int_{-1}^1 xU_k(x)w(x)dx = \gamma_1 + 2 \sum_{j=1}^{k/2} \int_{-1}^1 xT_{2j}(x)w(x)dx = \quad (1.54)$$

$$= \gamma_1 + \sum_{j=1}^{k/2} (\gamma_{2j+1} + \gamma_{2j-1}) = \quad (1.55)$$

$$= \gamma_1 + 2 \sum_{j=1}^{\frac{k}{2}-1} \gamma_{2j+1} + \gamma_{k+1} \quad (1.56)$$

ove  $\gamma_j$  è il  $j$ -esimo momento pesato di Chebyshev. Questo risultato, insieme a:

$$\int_{-1}^1 U_k(x)w(x)dx = \gamma_0 + 2 \sum_{j=1}^{k/2} \gamma_{2j} \quad (1.57)$$

dà come conseguenza:

$$\int_{-1}^1 (1+x)U_k(x)w(x)dx = \gamma_0 + 2 \sum_{j=1}^k \gamma_j + \gamma_{k+1} \quad (1.58)$$

**Caso 2:  $n$  dispari.** Osservando che, come nel caso 1, vale:

$$\int_{-1}^1 xU_k(x)w(x)dx = \sum_{j \text{ dispari}}^k (\gamma_{j+1} + \gamma_{j-1}) = \gamma_0 + 2 \sum_{j \text{ pari}}^{k-1} \gamma_j + \gamma_{k+1} \quad (1.59)$$

ne segue lo stesso risultato:

$$\int_{-1}^1 (1+x)U_k(x)w(x)dx = 2 \sum_{j \text{ dispari}}^k \int_{-1}^1 (1+x)T_j(x)w(x)dx = \quad (1.60)$$

$$= 2 \sum_{j \text{ dispari}}^k \gamma_j + \sum_{j \text{ dispari}}^k (\gamma_{j+1} + \gamma_{j-1}) = \quad (1.61)$$

$$= \gamma_0 + 2 \sum_{j=1}^k \gamma_j + \gamma_{k+1} \quad (1.62)$$

**Conclusione dei due casi.** Si è ottenuto che

$$w_1 = \frac{1}{2(n-1)} \int_{-1}^1 (1+x)U_{n-2}(x)w(x)dx = \frac{1}{2(n-1)} \left( \gamma_0 + 2 \sum_{j=1}^{n-2} \gamma_j + \gamma_{n-1} \right) \quad (1.63)$$

Per  $x_n = -1$  si procede analogamente. Da  $U_n(-1) = (-1)^n(n+1)$  si ha:

$$\frac{P_n(x)}{P'_n(-1)} = \frac{(1-x^2)U_{n-2}(x)}{-2U_{n-2}(-1)} = \frac{(1-x^2)U_{n-2}(x)}{2(-1)^n(n-1)} \quad (1.64)$$

e da esso:

$$w_n = \frac{1}{P'_n(-1)} \int_{-1}^1 \frac{P_n(x)w(x)}{x+1} dx = \frac{1}{2(n-1)(-1)^n} \int_{-1}^1 (1-x)U_{n-2}(x)w(x)dx \quad (1.65)$$

L'integrale dell'ultima equazione è calcolabile allo stesso modo del caso  $x_1 = +1$ . Si ottiene:

$$\int_{-1}^1 (1-x)U_s(x)w(x)dx = \begin{cases} 2 \sum_{k=0}^{s+1} (-1)^k \gamma_k - \gamma_0 + \gamma_{s+1} & \text{se } s \text{ è pari} \\ 2 \sum_{k=0}^{s+1} (-1)^{k+1} \gamma_k + \gamma_0 + \gamma_{s+1} & \text{se } s \text{ è dispari} \end{cases} \quad (1.66)$$

Procedendo come sopra, si ottiene:

$$w_n = \begin{cases} \frac{1}{2(n-1)} (2 \sum_{k=0}^{n-1} (-1)^k \gamma_k - \gamma_0 + \gamma_{n-1}) & \text{se } n \text{ è pari} \\ \frac{-1}{2(n-1)} (2 \sum_{k=0}^{n-1} (-1)^{k+1} \gamma_k + \gamma_0 + \gamma_{n-1}) & \text{se } n \text{ è dispari} \end{cases} \quad (1.67)$$

Si ottiene quindi la seguente espressione esplicita per  $w_n$  come funzione dei momenti  $\gamma_j$ :

$$w_n = \frac{1}{2(n-1)} \left( 2 \sum_{k=0}^{n-1} (-1)^k \gamma_k - \gamma_0 + (-1)^n \gamma_{n-1} \right) \quad (1.68)$$

**Nota.** Si ricorda che la regola di Clenshaw-Curtis è interpolatoria, ovvero costruita sulla base di un polinomio che interpoli la funzione integranda. Come conseguenza di ciò, la somma dei pesi è uguale al momento  $\gamma_0 = \int_{-1}^1 w(x)dx$ , ove chiaramente  $w$  è la funzione peso.

Pertanto, una volta calcolati tutti i pesi  $w_j$  tranne uno, l'ultimo può essere ricavato da una banale sottrazione:

$$w_n = \gamma_0 - \sum_{j=1}^{n-1} w_j$$

e addirittura se la funzione peso è simmetrica, vale  $w_1 = w_n$  e quindi:

$$w_1 = \frac{1}{2} \left( \gamma_0 - \sum_{j=2}^{n-1} w_j \right) = w_k$$

ovvero i pesi relativi ai nodi  $w_1 = 1$  e  $w_n = -1$  sono immediatamente calcolabili una volta ottenuti il momento  $\gamma_0$  e gli altri  $n-2$  pesi.

### 1.3.1 Calcolo dei momenti di Chebyshev

È stato osservato che, noti i pesi di Fejér  $w_{j-1}^{F2}$  rispetto alla funzione peso  $w^*(x) = (1-x^2)w(x)$ , è possibile ricavarne i pesi di Clenshaw-Curtis. È però necessario calcolare i momenti:

$$\mu_k := \int_{-1}^1 (1-x^2)U_k(x)w(x)dx$$

Ora, si ricorda che:

$$(1-x^2)U_k(x) = xT_{k+1}(x) - T_{k+2}(x) \quad k \in \mathbb{N}$$

$$xT_j(x) = \frac{T_{j+1}(x) + T_{j-1}(x)}{2} \quad j \geq 1$$

Si ottiene  $\forall k \in \mathbb{N}$ :

$$(1-x^2)U_k(x) = xT_{k+1}(x) - T_{k+2}(x) = \tag{1.69}$$

$$= \frac{1}{2}(T_{k+2}(x) + T_k(x)) - T_{k+2}(x) = \tag{1.70}$$

$$= \frac{1}{2}(T_k(x) - T_{k+2}(x)) \tag{1.71}$$

e quindi:

$$\mu_k = \frac{1}{2} \int_{-1}^1 (T_k(x) - T_{k+2}(x)) w(x) dx = \quad (1.72)$$

$$= \frac{\gamma_k - \gamma_{k+2}}{2} \quad (1.73)$$

Pertanto i momenti  $\{\mu_j\}_{j=0,\dots,n-1}$  sono facilmente calcolabili una volta noti i momenti  $\{\gamma_j\}_{j=0,\dots,n-1}$ .

## 1.4 Grado di precisione

Le tre regole descritte finora sono formule con grado di precisione algebrico pari a  $n - 1$  (se  $n$  è il numero di nodi), ovvero sono esatte se la funzione integranda  $f$  è un polinomio di grado al più  $n - 1$ :

$$\int_{-1}^1 p(x) w(x) dx = \sum_{k=1}^n w_k p(x_k) \quad (1.74)$$

con  $p \in \mathbb{P}_{n-1} := \{\text{polinomi algebrici di grado } n - 1\}$ .

È da notare che, benché la regola di Clenshaw-Curtis con  $n$  nodi abbia un grado di precisione nominale di  $n - 1$ , L. Trefethen ha mostrato in [19] che spesso essa ha una performance paragonabile alle regole gaussiane, ovvero quelle con un grado di precisione massimo, pari a  $2n - 1$ .



# Capitolo 2

## Implementazione in Matlab

Per implementare in Matlab le regole trattate nei capitoli precedenti, utilizzando le DCT e le DST, si deve ricorrere alle funzioni `idct` (che corrisponde a una DCT-III) e `dst` (DST-I), già presenti di default nelle versioni più recenti di Matlab. Queste tre possono bastare per scrivere un programma che calcoli nodi e pesi per le tre regole finora elencate nel caso di una generica funzione peso.

Invece per applicare i casi particolari con la funzione peso simmetrica, bisogna ricorrere a funzioni esterne, poiché la DCT-VII, la DST-II e la DST-VI non sono implementate nativamente in Matlab.

### 2.1 Regola di Fejér di primo tipo

In base a quanto osservato nel capitolo precedente, per calcolare i pesi della regola di Fejér di primo tipo si può applicare la DCT-III (e dunque la funzione `idct`) al vettore dei momenti. Stando alla documentazione di Matlab [14], la formula usata da `idct` è:

$$x_n = \sum_{k=1}^N h_k y_k \cos \left[ \frac{\pi(2n-1)(k-1)}{2N} \right] \quad n = 1, \dots, N$$

ove  $\{y_k\}_{k=1,\dots,N}$  sono i dati in input e:

$$h_k := \begin{cases} \frac{1}{\sqrt{N}} & k = 1 \\ \sqrt{\frac{2}{N}} & 2 \leq k \leq N \end{cases}$$

Si può ricavare una formula molto simile alla seconda tramite qualche cambiamento di indice. Infatti, indicando con  $\sum'$  una sommatoria in cui il primo termine è dimezzato, si scrive:

$$\begin{aligned}
w_k &= \frac{2}{n} \left[ \sum_{m=0}^{n-1}' \gamma_m \cos \left( \frac{m(2k-1)\pi}{2n} \right) \right] \stackrel{(n \rightarrow N)}{=} \\
&= \frac{2}{N} \left[ \sum_{m=0}^{N-1}' \gamma_m \cos \left( \frac{m(2k-1)\pi}{2N} \right) \right] \stackrel{(k \rightarrow n)}{=} \\
w_n &= \frac{2}{N} \left[ \sum_{m=0}^{N-1}' \gamma_m \cos \left( \frac{m(2n-1)\pi}{2N} \right) \right] \stackrel{(m \rightarrow k)}{=} \\
&= \frac{2}{N} \left[ \sum_{k=0}^{N-1}' \gamma_k \cos \left( \frac{k(2n-1)\pi}{2N} \right) \right] = \\
&= \frac{2}{N} \left[ \sum_{k=1}^N' \gamma_{k-1} \cos \left( \frac{(k-1)(2n-1)\pi}{2N} \right) \right] = \\
&= \sqrt{\frac{2}{N}} \left[ \sum_{k=1}^N' \sqrt{\frac{2}{N}} \gamma_{k-1} \cos \left( \frac{(k-1)(2n-1)\pi}{2N} \right) \right]
\end{aligned}$$

Si ottiene una formula molto simile a quella usata da Matlab in `idct`: l'unica differenza fra le formule è la presenza di  $\sum'$ . Si può notare di aver ottenuto:

$$w_n = \sqrt{\frac{2}{N}} \left[ \frac{\gamma_1}{2} \sqrt{\frac{2}{N}} + \sum_{k=2}^N \sqrt{\frac{2}{N}} \gamma_{k-1} \cos \left( \frac{(k-1)(2n-1)\pi}{2N} \right) \right]$$

Come primo termine della somma si ha dunque  $\frac{\gamma_1}{2} \sqrt{\frac{2}{N}} = \frac{\gamma_1}{\sqrt{2N}}$ , mentre la formula usata da Matlab per  $k = 1$  calcola  $\frac{y_k}{\sqrt{N}}$  (in questo caso specifico,  $y_1 = \gamma_1$ ).

Quindi se nel vettore  $\gamma = (\gamma_1, \gamma_2, \dots, \gamma_n)$  dei momenti dei polinomi di Chebyshev si divide il primo elemento  $\gamma_1$  per  $\sqrt{2}$ , si ottiene un vettore:

$$\gamma^* := \left( \frac{\gamma_1}{\sqrt{2}}, \gamma_2, \gamma_3, \dots, \gamma_n \right)$$

tale che:

$$w = \sqrt{\frac{2}{N}} \text{idct}(\gamma^*)$$

sia il vettore dei pesi di Fejér.

Un programma in Matlab che, dato il vettore `mom` degli  $n$  momenti (pesati rispetto alla funzione peso) di Chebyshev, restituisca un vettore `x` dei nodi e un vettore `w` dei pesi di Fejér di primo tipo, è:

```
function [x1,w1]=fejer1(n,mom)
x1=cos((2*(1:n))-1)*pi/(2*n));
mom(1)=mom(1)*sqrt(1/2);
w1=sqrt(2/n).*idct(mom);
end
```

## 2.2 Regola di Fejér di secondo tipo

Come già visto, per ricavare i pesi di Fejér di secondo tipo nel caso generico, si può fare affidamento alla DST-I. In Matlab essa è implementata dalla funzione `dst`, che nella documentazione ufficiale di Matlab [13] è così definita, dati gli  $\{x_n\}_{n=1,\dots,N}$  in input:

$$y_k = \sum_{n=1}^N x_n \sin\left(\frac{\pi k n}{N+1}\right) \quad k = 1, \dots, N$$

Si nota che tale definizione è in linea con quella enunciata nel capitolo precedente e dunque, al contrario della `idct`, non vi è bisogno di normalizzazioni. Dunque si ottiene che, noto il vettore `mom2` dei momenti  $(m_0, \dots, m_{n-1})$ , il vettore dei pesi è dato da:

$$w = \frac{2 \sin(\theta)}{n+1} \text{dst}(\text{mom2}) \quad (2.1)$$

Inoltre, è stato osservato che `mom2` può essere facilmente ricavato dallo stesso vettore dei momenti di Chebyshev precedentemente dato in input alla funzione `fejer1`. Quindi, supponendo di avere una funzione `mf2` che a partire dal vettore  $(\gamma_0, \dots, \gamma_{n-1})$  calcoli il vettore  $(m_0, \dots, m_{n-1})$ , si può scrivere il seguente programma:

```
function [x2,w2]=fejer2(n,mom)
theta=(1:n)'*pi/(n+1);
x2=cos(theta);
mom2=mf2(mom);
w2=(2*sin(theta)/(n+1)).*dst(mom2);
end
```

Un'implementazione di `mf2` è data da:

```
function [mom2]=mf2(mom)
n=length(mom);
mom2=zeros(n,1);
pari=mom(1:2:end);
dispari=mom(2:2:end);
mom2(1:2:end)=2*cumsum(par)-mom(1);
mom2(2:2:end)=2*cumsum(dispari);
end
```

## 2.3 Regola di Clenshaw-Curtis

Come visto nella sottosezione 1.3.1, è possibile calcolare i momenti  $\mu_k$  (pesati rispetto a  $w^*(x) := (1-x^2)w(x)$ ) a partire dai momenti dei polinomi di Chebyshev, pesati rispetto a  $w$ . Si supponga dunque di avere una funzione `momcc` che, a partire dal vettore `mom`, calcoli  $(\mu_0, \dots, \mu_{n-1})$ ; allora un programma che produca i nodi e i pesi di Clenshaw-Curtis è:

```
function [x3,w3]=clenshawcurtis(n,mom)
mom3=momcc(mom);
theta=(1:n-2)'*pi/(n-1);
x=cos(theta); % nodi non estremali
x3=[1;x;-1];
w=((2*sin(theta)/(n-1)).*dst(mom3))./(1-x.^2);
w1=(2*sum(mom)-mom(1)-mom(end))/(2*(n-1));
wn=mom(1)-w1-sum(w);
w3=[w1,w,wn];
end
```

Chiaramente, se è noto che la funzione peso  $w$  è simmetrica, allora è possibile imporre alla variabile `wn` la stessa condizione di `w1`, come osservato nella nota finale della sezione 1.3.

Un esempio di funzione `momcc` è:

```
function mu=momcc(g)
g=g(1:n+3);
mu=(g(1:end-2)-g(3:end))./2;
end
```

## 2.4 Esempio: funzione peso di Gegenbauer

La funzione peso di Gegenbauer è una funzione di  $(-1, 1)$  in  $\mathbb{R}$  così definita (cfr. [10, §1.70]):

$$w^{(\lambda)}(x) = (1 - x^2)^{\lambda - \frac{1}{2}} \quad \lambda > -\frac{1}{2} \quad (2.2)$$

I momenti della funzione peso di Gegenbauer sono facili da calcolare, noto che:

$$\gamma_0 := \int_{-1}^1 (1 - x^2)^{\lambda - \frac{1}{2}} dx = \sqrt{\pi} \frac{\Gamma(\lambda + \frac{1}{2})}{\Gamma(\lambda + 1)} \quad (2.3)$$

e che, come dimostrato in [12, §1] per mezzo di approssimazioni finite con polinomi di Chebyshev, per  $r \in \mathbb{N}^+$  vale:

$$\gamma_{2r} = \int_{-1}^1 (1 - x^2)^{\lambda - \frac{1}{2}} T_{2r}(x) = \quad (2.4)$$

$$= \sqrt{\pi} \frac{\Gamma(\lambda + \frac{1}{2})}{\Gamma(\lambda + 1)} G_r(\lambda) = \gamma_0 G_r(\lambda) \quad (2.5)$$

con:

$$G_r(\lambda) := \prod_{j=1}^r \frac{j - \lambda - 1}{j + \lambda} \quad (2.6)$$

mentre, poiché la funzione peso di Gegenbauer è simmetrica e i  $T_{2r+1}$  sono funzioni dispari, i momenti di indice dispari sono nulli:  $\gamma_{2r+1} = 0$ .

Una routine appositamente creata, eseguita su Matlab 7.12.0.635 R2011a, installato su un computer con processore Intel Core i5-2500 a 3.30 GHz con 3.1 GiB di RAM, ha restituito i risultati mostrati nella tabella 2.1, la quale confronta i tempi di calcolo dei nodi e dei pesi delle tre regole prese finora in considerazione con quelle gaussiane, per  $\lambda = 0.75$ . Si nota che già per un numero di nodi inferiore alle centinaia i metodi non gaussiani si rivelano fortemente competitivi. I codici usati sono nel capitolo 4 (*Appendice*).

Nella tabella 2.1 il tempo di calcolo di Gauss per un numero di nodi superiore a 8192 si rivela nell'ordine di diversi minuti e quindi non viene indicato.

Si ricorda che, dal momento che i vettori dei momenti pesati dalla funzione di Gegenbauer sono rapidissimi da calcolare, la loro computazione richiede una piccola frazione del tempo mostrato nella tabella 2.1.

Numero di nodi	Fejér 1	Fejér 2	Clenshaw-Curtis	Gauss
2	$3.157 \cdot 10^{-4}$	$2.432 \cdot 10^{-4}$	$1.355 \cdot 10^{-4}$	$1.799 \cdot 10^{-4}$
4	$1.456 \cdot 10^{-4}$	$1.082 \cdot 10^{-4}$	$1.235 \cdot 10^{-4}$	$1.097 \cdot 10^{-4}$
8	$1.385 \cdot 10^{-4}$	$1.112 \cdot 10^{-4}$	$1.274 \cdot 10^{-4}$	$1.109 \cdot 10^{-4}$
16	$1.387 \cdot 10^{-4}$	$1.129 \cdot 10^{-4}$	$1.304 \cdot 10^{-4}$	$1.407 \cdot 10^{-4}$
32	$1.448 \cdot 10^{-4}$	$1.159 \cdot 10^{-4}$	$1.342 \cdot 10^{-4}$	$2.327 \cdot 10^{-4}$
64	$1.481 \cdot 10^{-4}$	$1.225 \cdot 10^{-4}$	$1.423 \cdot 10^{-4}$	$9.479 \cdot 10^{-4}$
128	$1.566 \cdot 10^{-4}$	$1.319 \cdot 10^{-4}$	$1.524 \cdot 10^{-4}$	$5.254 \cdot 10^{-3}$
256	$1.748 \cdot 10^{-4}$	$1.522 \cdot 10^{-4}$	$1.714 \cdot 10^{-4}$	$1.390 \cdot 10^{-2}$
512	$2.093 \cdot 10^{-4}$	$1.930 \cdot 10^{-4}$	$2.162 \cdot 10^{-4}$	$6.731 \cdot 10^{-2}$
1024	$2.763 \cdot 10^{-4}$	$2.713 \cdot 10^{-4}$	$3.120 \cdot 10^{-4}$	$5.046 \cdot 10^{-1}$
2048	$4.240 \cdot 10^{-4}$	$5.189 \cdot 10^{-4}$	$5.362 \cdot 10^{-4}$	4.290
4096	$7.774 \cdot 10^{-4}$	$8.817 \cdot 10^{-4}$	$7.748 \cdot 10^{-4}$	33.18
8192	$1.421 \cdot 10^{-3}$	$1.706 \cdot 10^{-3}$	$2.295 \cdot 10^{-3}$	
16384	$2.684 \cdot 10^{-3}$	$3.881 \cdot 10^{-3}$	$4.384 \cdot 10^{-3}$	
32768	$4.919 \cdot 10^{-3}$	$1.217 \cdot 10^{-2}$	$7.506 \cdot 10^{-3}$	
65536	$1.019 \cdot 10^{-2}$	$1.574 \cdot 10^{-2}$	$1.680 \cdot 10^{-2}$	
131072	$2.140 \cdot 10^{-2}$	$3.471 \cdot 10^{-2}$	$3.872 \cdot 10^{-2}$	
262144	$4.192 \cdot 10^{-2}$	$6.046 \cdot 10^{-2}$	$6.572 \cdot 10^{-2}$	
524288	$8.539 \cdot 10^{-2}$	$1.435 \cdot 10^{-1}$	$1.479 \cdot 10^{-1}$	
1048576	$1.706 \cdot 10^{-1}$	$4.096 \cdot 10^{-1}$	$3.086 \cdot 10^{-1}$	

Tabella 2.1: Valori medi (su 30 esperimenti) del tempo di calcolo in secondi per le regole di Fejér di primo e secondo tipo e di Clenshaw-Curtis, per la quadratura di Gegenbauer con  $\lambda = 0.75$ .

# Capitolo 3

## Applicazioni alla quadratura trigonometrica

In questo capitolo applichiamo i risultati descritti a problemi di quadratura trigonometrica, ovvero alla quadratura mediante polinomi trigonometrici interpolanti. In particolare, si determineranno i nodi e i pesi di quadratura lungo un arco di circonferenza e si descriveranno formule di cubatura su lenti circolari simmetriche.

### 3.1 Quadratura trigonometrica subperiodica

Indicato con  $\{\xi_j\}_{1 \leq j \leq 2n+1}$  l'insieme degli zeri del polinomio di Chebyshev  $T_{2n+1}(x)$  di grado  $2n + 1$ , ovvero:

$$\xi_j = \cos \left( \frac{(2j-1)\pi}{2(2n+1)} \right) \quad j = 1, \dots, 2n+1 \quad (3.1)$$

e, dato un intervallo  $[-\omega, \omega]$ , si definiscono i nodi angolari:

$$\theta_j = 2 \arcsin(\alpha \xi_j) \quad j = 1, \dots, 2n+1 \quad \theta_j \in [-\omega, \omega] \quad (3.2)$$

come gli zeri di  $T_{2n+1}(\frac{\sin(\theta/2)}{\alpha})$ , con  $\alpha = \sin(\frac{\omega}{2})$ .

Sotto queste ipotesi, vale quanto dimostrato da L. Bos e M. Vianello in [1, §2]:

**Teorema.** L'insieme  $\{\xi_j\}_{1 \leq j \leq 2n+1}$  è unisolvente in

$$\mathbb{T}_n([-\omega, \omega]) = \text{span}\{1, \cos(k\theta), \sin(k\theta) : 1 \leq k \leq n, \theta \in [-\omega, \omega]\}$$

Inoltre, la formula di quadratura interpolatoria trigonometrica basata su  $\{\xi_j\}_{1 \leq j \leq 2n+1}$  ha come pesi positivi:

$$w_j = 2\alpha \int_{-1}^1 \frac{\ell_j(x)}{\sqrt{1 - \alpha^2 x^2}} dx = \quad (3.3)$$

$$= \frac{2\alpha}{2n+1} \left( m_0 + 2 \sum_{k=1}^n m_{2k} T_{2k}(\xi_j) \right) \quad j = 1, \dots, 2n+1 \quad (3.4)$$

con  $\ell_j(x)$  il  $j$ -esimo polinomio algebrico di Lagrange per i nodi  $\{\xi_j\}_{1 \leq j \leq 2n+1}$  e con:

$$m_{2k} = \int_{-1}^1 \frac{T_{2k}(x)}{\sqrt{1 - \alpha^2 x^2}} dx \quad k = 0, 1, \dots, n \quad (3.5)$$

Si nota che in (3.4) appare una formula ben familiare: i valori dei pesi non sono altro che quelli della regola di Fejér di primo tipo. L'intenzione è dunque di applicare i risultati dei capitoli precedenti in questo caso particolare, utilizzando le formule di Fejér di secondo tipo e di Clenshaw-Curtis.

In [1], gli autori forniscono una funzione Matlab, `trigquad`, che calcola i nodi angolari e i rispettivi pesi, come previsto in (3.4). Nel dettaglio, una volta ottenuti il vettore `x` degli  $\{\xi_j\}_{1 \leq j \leq 2n+1}$  e il vettore `mom` dei momenti di Chebyshev di indice pari (calcolati grazie alla formula ricorsiva), il codice fornito determina:

```
T=cos((2*(1:n))'*acos(x));
pesi=((mom(1)+2*(mom(2:n+1))*T)/(2*n+1))';
```

Queste due righe possono essere sostituite, rispettivamente, dalle funzioni `fejer1`, `fejer2` e `clenshawcurtis`, estendendo il vettore `mom` con l'aggiunta dei momenti di indice dispari, che sono nulli. Nel caso della regola di Fejér di primo tipo, si ha:

```
moms=zeros(1,2*n+1);
moms(1:2:2*n+1)=mom(1:n+1);
[nodi,pesi]=fejer1(2*n+1,moms);
```

I casi con le altre due funzioni sono analoghi. Si noti che il vettore `nodi` è il vettore degli  $\{\xi_j\}_{1 \leq j \leq 2n+1}$ , da cui si dovrà poi ricavare il vettore delle  $\{\theta_j\}_{1 \leq j \leq 2n+1}$  tramite l'equazione (3.2).

In [4, §2], G. Da Fies e M. Vianello hanno ottenuto il seguente, importante risultato:

**Teorema.** Se  $\{(\xi_j, \lambda_j)\}_{1 \leq j \leq n+1}$  è l'insieme dei nodi e dei pesi positivi della formula di quadratura algebrica gaussiana per la funzione peso definita da:

$$w(x) = \frac{2 \sin(\frac{\omega}{2})}{\sqrt{1 - \sin^2(\frac{\omega}{2})x^2}} \quad x \in (-1, 1) \quad (3.6)$$

allora, posto:

$$\phi_j = 2 \arcsin \left( \sin \left( \frac{\omega}{2} \right) \xi_j \right) \quad j = 1, \dots, n+1 \quad (3.7)$$

risulta per  $f \in \mathbb{T}_n([-\omega, \omega])$ , con  $\omega \in [0, \pi]$ :

$$\int_{-\omega}^{\omega} f(\theta) d\theta = \sum_{j=1}^{n+1} \lambda_j f(\phi_j) \quad (3.8)$$

Si vuole ora calcolare i nodi di Fejér e di Clenshaw-Curtis per la funzione peso  $w$  definita in (3.6) ad un grado di precisione che sia uguale a quello delle formule gaussiane. Chiaramente questa scelta renderà necessario il calcolo di un numero maggiore di nodi; ma d'altra parte ci si aspetta che essa avvenga più rapidamente rispetto alla formula gaussiana.

Sarà inoltre necessario poter calcolare il vettore dei momenti di Chebyshev, pesati rispetto alla funzione  $w$  descritta in (3.6). Un metodo relativamente rapido consiste nell'algoritmo di Olver [15], il quale, utilizzando i coefficienti di ricorsione dei polinomi di Chebyshev, ne ottiene i momenti tramite la risoluzione di un sistema lineare definito da una matrice tridiagonale.

Si può sostituire quindi la formula algebrica gaussiana indicata nell'enunciato del teorema, che ha grado di precisione  $2n + 1$  in quanto effettuata su  $n + 1$  nodi, con una formula di Fejér o di Clenshaw-Curtis su  $2n + 2$  nodi, che ha sempre grado di precisione  $2n + 1$ . In questo modo, si possono costruire funzioni apposite (che chiamiamo rispettivamente `trigquadf1`, `trigquadf2` e `trigquadcc`) per il calcolo dei nodi e dei pesi trigonometrici.

Data una funzione `olver` che calcoli il vettore dei momenti di Chebyshev, un esempio di funzione `trigquadf1` è:

```
function tw = trigquadf1(n,alpha,beta)
omega=(beta-alpha)/2;

% algoritmo di Olver
mom=olver(2*n+1,omega);
```

$m$	Fejér 1	Fejér 2	Clebschaw-Curtis	Gauss
2	$3.531 \cdot 10^{-4}$	$1.709 \cdot 10^{-4}$	$1.355 \cdot 10^{-4}$	$6.987 \cdot 10^{-4}$
4	$1.509 \cdot 10^{-4}$	$1.151 \cdot 10^{-4}$	$1.361 \cdot 10^{-4}$	$6.797 \cdot 10^{-4}$
8	$1.540 \cdot 10^{-4}$	$1.195 \cdot 10^{-4}$	$1.410 \cdot 10^{-4}$	$6.922 \cdot 10^{-4}$
16	$1.592 \cdot 10^{-4}$	$1.236 \cdot 10^{-4}$	$1.453 \cdot 10^{-4}$	$9.361 \cdot 10^{-4}$
32	$1.670 \cdot 10^{-4}$	$1.339 \cdot 10^{-4}$	$1.527 \cdot 10^{-4}$	$1.324 \cdot 10^{-3}$
64	$1.828 \cdot 10^{-4}$	$1.513 \cdot 10^{-4}$	$1.713 \cdot 10^{-4}$	$3.170 \cdot 10^{-3}$
128	$2.381 \cdot 10^{-4}$	$1.940 \cdot 10^{-4}$	$2.198 \cdot 10^{-4}$	$8.254 \cdot 10^{-2}$
256	$3.068 \cdot 10^{-4}$	$2.722 \cdot 10^{-4}$	$2.838 \cdot 10^{-4}$	$2.123 \cdot 10^{-2}$
512	$4.045 \cdot 10^{-4}$	$3.634 \cdot 10^{-4}$	$3.783 \cdot 10^{-4}$	$9.005 \cdot 10^{-2}$
1024	$6.948 \cdot 10^{-4}$	$6.516 \cdot 10^{-4}$	$7.169 \cdot 10^{-4}$	—
2048	$1.518 \cdot 10^{-3}$	$1.550 \cdot 10^{-3}$	$1.288 \cdot 10^{-3}$	—
4096	$2.800 \cdot 10^{-3}$	$2.603 \cdot 10^{-3}$	$2.283 \cdot 10^{-3}$	—
8192	$4.690 \cdot 10^{-3}$	$4.370 \cdot 10^{-3}$	$6.488 \cdot 10^{-3}$	—
16384	$9.476 \cdot 10^{-3}$	$1.354 \cdot 10^{-2}$	$1.109 \cdot 10^{-2}$	—
32768	$2.230 \cdot 10^{-2}$	$3.051 \cdot 10^{-2}$	$1.788 \cdot 10^{-2}$	—
65536	$3.334 \cdot 10^{-2}$	$4.465 \cdot 10^{-2}$	$4.312 \cdot 10^{-2}$	—
131072	$8.330 \cdot 10^{-1}$	$9.909 \cdot 10^{-2}$	$7.555 \cdot 10^{-2}$	—
262144	$1.440 \cdot 10^{-1}$	$1.558 \cdot 10^{-1}$	$1.810 \cdot 10^{-1}$	—
524288	$3.377 \cdot 10^{-1}$	$4.093 \cdot 10^{-1}$	$4.770 \cdot 10^{-1}$	—
1048576	$8.092 \cdot 10^{-1}$	$7.772 \cdot 10^{-1}$	$8.095 \cdot 10^{-1}$	—

Tabella 3.1: Valore medio (su 30 esperimenti) del tempo di calcolo in secondi delle regole di Fejér di primo e secondo tipo e di Clenshaw-Curtis per la quadratura trigonometrica con  $\omega = \frac{\pi}{4}$  e grado di precisione trigonometrico  $m$ .

```
[nodi,pesi]=fejer1(2*n+2,mom);
tw(:,1)=2*asin(sin(omega/2)*nodi)+(alpha+beta)/2;
tw(:,2)=pesi;
```

Le funzioni `trigquadf2` e `trigquadcc` si ottengono modificando la terzultima riga del codice sopra riportato, ovvero sostituendo `fejer1` con `fejer2` e `clenshawcurtis` rispettivamente.

Una routine appositamente creata ha restituito i risultati mostrati nella tabella 3.1. Essa confronta i tempi medi di calcolo, su 30 esperimenti, delle regole di Fejér e di Clenshaw-Curtis con quelli delle regole gaussiane (ottenute tramite la funzione `trigauss` disponibile in [3]), nel caso di  $\omega = \frac{\pi}{4}$ . Similmente al caso precedente

$m$	Fejér 1	Fejér 2	Clenshaw-Curtis	Gauss
2	$2.220 \cdot 10^{-16}$	$4.441 \cdot 10^{-16}$	$5.274 \cdot 10^{-16}$	$1.110 \cdot 10^{-15}$
4	$2.220 \cdot 10^{-16}$	$4.441 \cdot 10^{-16}$	$1.305 \cdot 10^{-15}$	$4.441 \cdot 10^{-16}$
8	$5.551 \cdot 10^{-16}$	$1.388 \cdot 10^{-16}$	$1.887 \cdot 10^{-15}$	$3.775 \cdot 10^{-15}$
16	$2.665 \cdot 10^{-15}$	$1.138 \cdot 10^{-15}$	$1.971 \cdot 10^{-15}$	$3.803 \cdot 10^{-15}$
32	$3.081 \cdot 10^{-15}$	$2.373 \cdot 10^{-15}$	$3.650 \cdot 10^{-15}$	$6.870 \cdot 10^{-15}$
64	$3.907 \cdot 10^{-15}$	$4.295 \cdot 10^{-15}$	$3.102 \cdot 10^{-15}$	$1.465 \cdot 10^{-14}$
128	$4.378 \cdot 10^{-15}$	$4.408 \cdot 10^{-15}$	$5.756 \cdot 10^{-15}$	$2.460 \cdot 10^{-14}$
256	$1.258 \cdot 10^{-14}$	$9.044 \cdot 10^{-14}$	$1.072 \cdot 10^{-14}$	$5.215 \cdot 10^{-14}$
512	$1.422 \cdot 10^{-14}$	$1.278 \cdot 10^{-14}$	$1.753 \cdot 10^{-14}$	$7.903 \cdot 10^{-14}$

Tabella 3.2: Valori massimi degli errori assoluti delle formule di quadratura (con grado di precisione  $m$ ) rispetto a quella effettiva, nel calcolo dell'integrale di  $\cos(kx)$  con  $\omega = 1.5, k = 1, \dots, m$ .

$m$	Fejér 1	Fejér 2	Clenshaw-Curtis	Gauss
2	$1.943 \cdot 10^{-16}$	$1.949 \cdot 10^{-16}$	$4.580 \cdot 10^{-16}$	$3.331 \cdot 10^{-16}$
4	$4.267 \cdot 10^{-16}$	$3.886 \cdot 10^{-16}$	$1.218 \cdot 10^{-15}$	$8.049 \cdot 10^{-16}$
8	$2.255 \cdot 10^{-15}$	$1.200 \cdot 10^{-15}$	$7.927 \cdot 10^{-16}$	$2.942 \cdot 10^{-15}$
16	$1.027 \cdot 10^{-15}$	$1.093 \cdot 10^{-15}$	$2.415 \cdot 10^{-15}$	$3.976 \cdot 10^{-15}$
32	$2.061 \cdot 10^{-15}$	$2.914 \cdot 10^{-15}$	$3.587 \cdot 10^{-15}$	$9.671 \cdot 10^{-15}$
64	$4.073 \cdot 10^{-15}$	$2.942 \cdot 10^{-15}$	$4.725 \cdot 10^{-15}$	$1.245 \cdot 10^{-14}$
128	$5.843 \cdot 10^{-15}$	$3.623 \cdot 10^{-15}$	$3.973 \cdot 10^{-15}$	$1.828 \cdot 10^{-14}$
256	$6.327 \cdot 10^{-15}$	$7.230 \cdot 10^{-15}$	$1.020 \cdot 10^{-14}$	$4.103 \cdot 10^{-14}$
512	$1.468 \cdot 10^{-14}$	$1.129 \cdot 10^{-14}$	$2.523 \cdot 10^{-14}$	$8.367 \cdot 10^{-14}$

Tabella 3.3: Valori massimi degli errori assoluti delle formule di quadratura (con grado di precisione  $m$ ) rispetto a quella effettiva, nel calcolo dell'integrale di  $\sin(kx)$  con  $\omega = 1.5, k = 1, \dots, m$ .

della funzione peso di Gegenbauer, anche qui si nota che le regole di Fejér e di Clenshaw-Curtis sono notevolmente vantaggiose. I codici usati sono riportati nel capitolo 4.

Si osservi che per motivi di stabilità numerica della funzione `eig` non si è riusciti a calcolare formule gaussiane con un grado di precisione maggiore di 512 tramite `trigauss`, mentre, senza problemi, si sono determinate regole non gaussiane con più di un milione di nodi.

L'esattezza sui polinomi trigonometrici delle formule appena costruite è suggerita

da test effettuati su apposite routine: gli errori assoluti degli integrali delle funzioni coseno e seno risultano sufficientemente piccoli. I valori massimi sono riportati nelle tabelle 3.2 e 3.3 a pagina 31.

## 3.2 Cubatura su lenti simmetriche

Il teorema sopra riportato ha interessanti applicazioni all'integrazione su alcuni domini bidimensionali, in particolare a regioni del cerchio: settori, segmenti e lenti circolari.

Quale applicazione alla cubatura, consideriamo l'esempio di una lente circolare simmetrica, ovvero l'intersezione di due cerchi di raggio  $R$  e i cui centri distano meno di  $2R$ . Tale regione del piano può essere definita, a meno di rotazioni e traslazioni, come:

$$\Omega := \{(x, y) = (Rt(\cos \theta - \cos \omega), R \sin \theta) : t \in [-1, 1], \theta \in [-\omega, \omega]\} \quad (3.9)$$

con  $\omega \in (0, \frac{\pi}{2}]$ . Come descritto in [4, §2.3.1], per le lenti simmetriche vale la seguente formula di cubatura di tipo prodotto:

$$\iint_{\Omega} f(x, y) dx dy = \sum_{j=1}^{n+3} \sum_{i=1}^{\lceil \frac{n+1}{2} \rceil} W_{ij} f(x_{ij}, y_{ij}) \quad \forall f \in \mathbb{P}_n^2(\Omega) \quad (3.10)$$

con:

$$W_{ij} := R^2 \cos(\phi_j) (\cos \phi_j - \cos \omega) w_i^{GL} \lambda_j$$

$$(x_{ij}, y_{ij}) := (R t_i^{GL} (\cos \phi_j - \cos \omega), R \sin \phi_j)$$

in cui  $\{(\phi_j, \lambda_j)\}$  sono definiti come nell'enunciato del teorema, mentre  $\{(t_i^{GL}, w_i^{GL})\}$  sono rispettivamente i nodi e i pesi della formula di Gauss-Legendre con grado di esattezza  $n$  su  $[-1, 1]$ . Il numero di nodi utilizzati è  $\frac{n^2}{2} + \frac{5}{2}n + 3$  se  $n$  è pari, mentre è  $\frac{n^2}{2} + 2n + \frac{3}{2}$  se  $n$  è dispari.

In particolare, tale formula di quadratura risulta esatta su  $\mathbb{P}_n^2(\Omega)$ , ovvero sui polinomi in due variabili di grado totale  $n$ .

La formula di quadratura sopra descritta consiste nell'applicazione di due formule di quadratura gaussiane in una dimensione. Si può però sostituire ciascuna di queste formule con una di Fejér o di Clenshaw-Curtis aventi appropriati gradi di precisione  $n$ . Benché esse richiedano un maggior numero di nodi per avere lo stesso grado di esattezza, la loro rapidità di esecuzione si mostrerà vantaggiosa, come pure la possibilità di fornire formule di grado  $n$  alto.

Innanzitutto, la formula di Gauss-Legendre con grado algebrico di esattezza  $m$  su  $[-1, 1]$ , che richiede  $\lceil \frac{m+1}{2} \rceil$  nodi, può essere sostituita da una formula di Fejér di primo o secondo tipo o di Clenshaw-Curtis su  $m + 1$  nodi.

In maniera simile, sull'arco potremmo utilizzare al posto di una formula gaussiana una di Fejér o di Clenshaw-Curtis. La regola gaussiana, essendo invocata su  $n + 3$  nodi, ha un grado di precisione di  $2(n + 3) - 1 = 2n + 5$  e quindi, per avere una nuova formula con lo stesso grado di precisione, si possono chiamare le funzioni `trigquadf1`, `trigquadf2` e `trigquadcc` rispettivamente, le quali, ricevuto il valore  $n$  in input, restituiscono una regola con  $2n + 2$  nodi e con grado di precisione  $2n$ . Se viene dato loro in input il valore  $n + 2$ , il loro grado di esattezza è  $2(n + 2) + 2 = 2n + 6$ , non inferiore a quello della regola gaussiana presa in esame.

Una formula di cubatura di tipo prodotto che utilizzi solamente formule non gaussiane sia sull'arco che su  $[-1, 1]$  sarà dunque:

$$\iint_{\Omega} f(x, y) dx dy = \sum_{j=1}^{2n+6} \sum_{i=1}^{n+1} W_{ij} f(x_{ij}, y_{ij}) \quad \forall f \in \mathbb{P}_n^2(\Omega) \quad (3.11)$$

con:

$$W_{ij} := R^2 \cos(\phi_j) (\cos \phi_j - \cos \omega) w_i \lambda_j$$

$$(x_{ij}, y_{ij}) := (R t_i (\cos \phi_j - \cos \omega), R \sin \phi_j)$$

in cui  $\{(\phi_j, \lambda_j)\}$  sono i nodi e i pesi della formula con  $2n + 6$  nodi applicata a  $[-\omega, \omega]$ , mentre  $\{(t_i, w_i)\}$  sono rispettivamente i nodi e i pesi della formula, con  $n + 1$  nodi, applicata a  $[-1, 1]$ . Il numero totale di nodi per una formula non gaussiana di grado di precisione algebrico pari a  $n$  è  $2n^2 + 8n + 6$ .

A questo punto è possibile scrivere una funzione `gqsymmlens2` che permetta di calcolare i nodi e i pesi corrispondenti alle sedici possibili combinazioni, sull'arco e su  $[-1, 1]$ , delle quattro regole di quadratura finora considerate: gaussiana, di Fejér di primo tipo, di Fejér di secondo tipo, di Clenshaw-Curtis.

Un esempio di tale funzione, che utilizza la `trigauss`, è:

```
function xyw = gqsymmlens2(n, omega, r, in1, in2)

% input:
%   n: grado algebrico di precisione
%   omega: lunghezza del semintervallo angolare (0<omega<=pi/2 )
%   r: raggio
%   in1: 0 = trigauss sull'arco
%        1 = trigquadf1 sull'arco
%        2 = trigquadf2 sull'arco
```

```

%      3 = trigquadcc sull'arco
% in2: 0 = trigauss su [-1,1]
%      1 = trigquadf1 su [-1,1]
%      2 = trigquadf2 su [-1,1]
%      3 = trigquadcc su [-1,1]

% output:
%   xyw: array di nodi (1a colonna: ascisse, 2a colonna: ordinate)
%       e pesi (3a colonna)

% formula trigonometrica sull'arco [-omega, omega]
switch in1
  case 0
    tw=trigauss(n+2,-omega,omega);
  case 1
    tw=trigquadf1(n+2,-omega,omega);
  case 2
    tw=trigquadf2(n+2,-omega,omega);
  case 3
    tw=trigquadcc(n+2,-omega,omega);
  otherwise
    warning('Parametro non valido. Sono validi solo: 0, 1, 2, 3.');
end

% formula algebrica su [-1,1]
switch in2
  case 0 % Gauss
    ab=r_jacobi(ceil((n+1)/2),0,0);
    xw=gauss(ceil((n+1)/2),ab);
    [t,theta]=meshgrid(xw(:,1),tw(:,1));
    [w1,w2]=meshgrid(xw(:,2),tw(:,2));
  case 1 % Fejer 1
    [x,w]=fejer1(n+1);
    [t,theta]=meshgrid(x,tw(:,1));
    [w1,w2]=meshgrid(w,tw(:,2));
  case 2 % Fejer 2
    [x,w]=fejer2(n+1);
    [t,theta]=meshgrid(x,tw(:,1));
    [w1,w2]=meshgrid(w,tw(:,2));
  case 3 % Clenshaw-Curtis

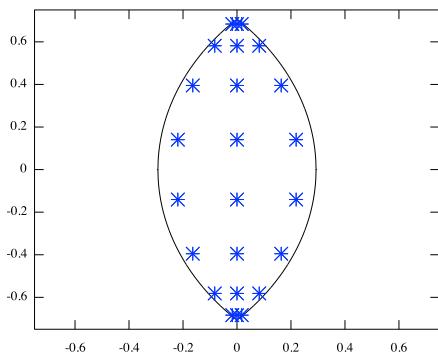
```

```

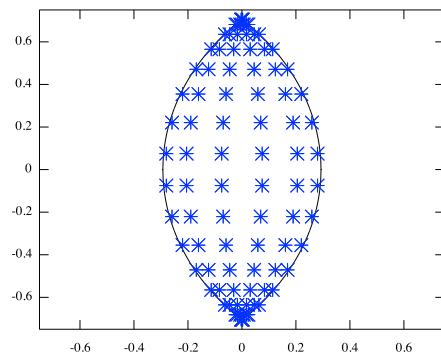
[x,w]=clenshawcurtis(n+1);
[t,theta]=meshgrid(x,tw(:,1));
[w1,w2]=meshgrid(w,tw(:,2));
otherwise
    warning('Parametro non valido. Sono validi solo: 0, 1, 2, 3.');
end
diffc=cos(theta(:))-cos(omega);
xyw(:,1)=r*t(:).*diffc;
xyw(:,2)=r*sin(theta(:));
xyw(:,3)=r^2*cos(theta(:)).*diffc.*w1(:).*w2(:,1);
end

```

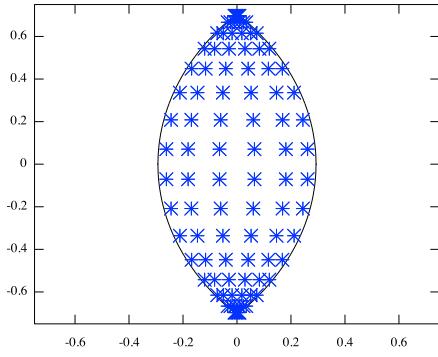
La figura 3.1 mostra i nodi di cubatura dei quattro metodi, applicati sia sull'arco



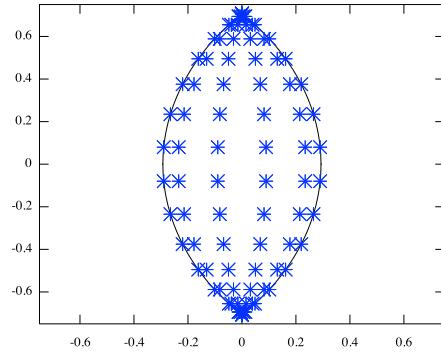
(a) Gauss



(b) Fejér di primo tipo



(c) Fejér di secondo tipo



(d) Clenshaw-Curtis

Figura 3.1: Nodi delle regole di cubatura con grado di precisione  $m = 5$  su una lente circolare simmetrica con  $\omega = \frac{\pi}{4}$  e  $r = 1$ . In ciascuno dei casi, la regola è stata applicata sia su  $[-\omega, \omega]$ , sia su  $[-1, 1]$ .

che su  $[-1, 1]$  con grado di precisione  $n = 5$ , su una lente circolare simmetrica con  $\omega = \frac{\pi}{4}$  e raggio  $r = 1$ .

Come esempio della precisione di `gqsymmlens2`, è stato studiato l'errore nell'integrazione delle seguenti funzioni su una lente simmetrica con  $\omega = \frac{\pi}{4}$  e raggio  $r = 1$ , rispetto al valore fornito da `trigquad` con grado opportuno:

- a.  $f(x, y) = (x + y)^{19}$
- b.  $f(x, y) = \cos(x + y)$
- c.  $f(x, y) = \cos(20(x + y))$
- d.  $f(x, y) = (x^2 + y^2)^{\frac{3}{2}}$
- e.  $f(x, y) = (x^2 + y^2)^{\frac{5}{2}}$

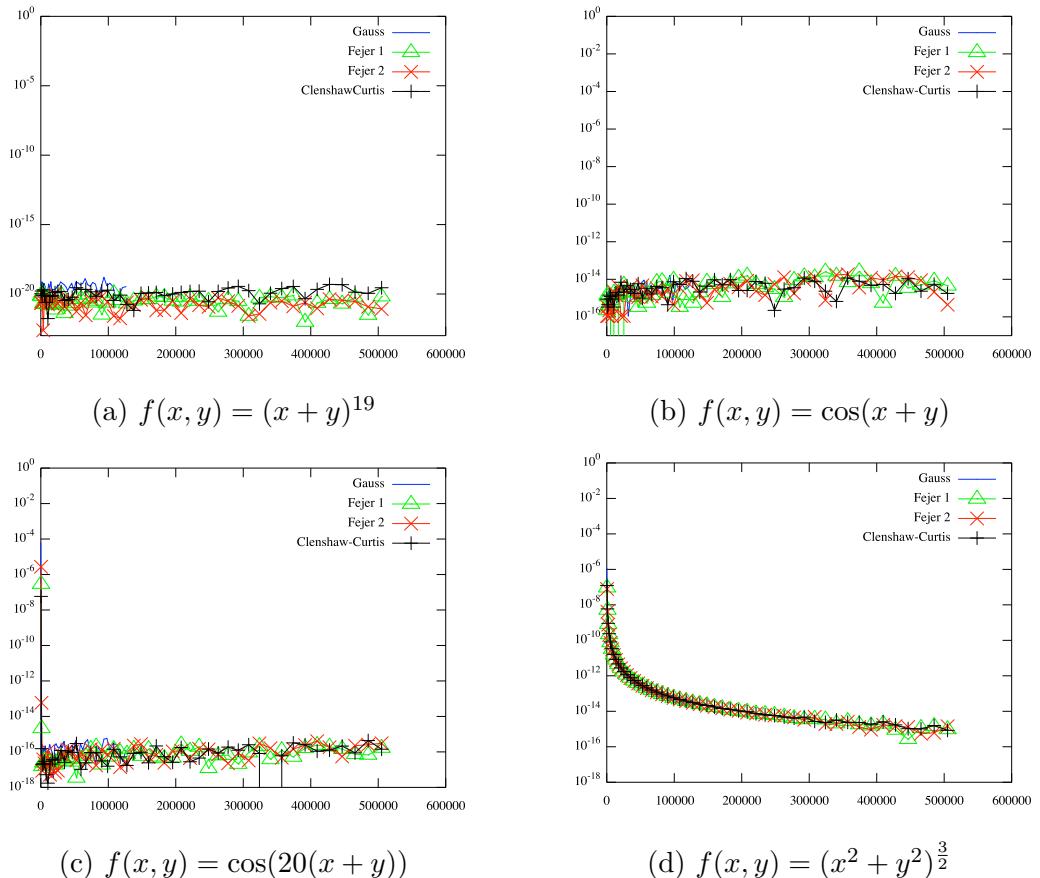


Figura 3.2: Errori assoluti nel calcolo degli integrali su una lente simmetrica con  $\omega = \frac{\pi}{4}$  e raggio 1, al variare del numero di nodi.

f.  $f(x, y) = \exp(-x^2 - y^2)$

g.  $f(x, y) = \exp(-100(x^2 + y^2))$

I grafici nelle figure 3.2 e 3.3 mostrano l'errore assoluto dell'integrale al variare del numero di nodi utilizzati.

Si studiano ora gli errori della cubatura per  $I(\omega, r) := \iint_{\Omega} (x + y + 2)^m dx dy$ , ove  $\Omega$  è la lente circolare simmetrica definita in (3.9). Per confrontare il risultato fornito dalle quadrature di Gauss, Fejér e Clenshaw-Curtis, si utilizzerà il risultato

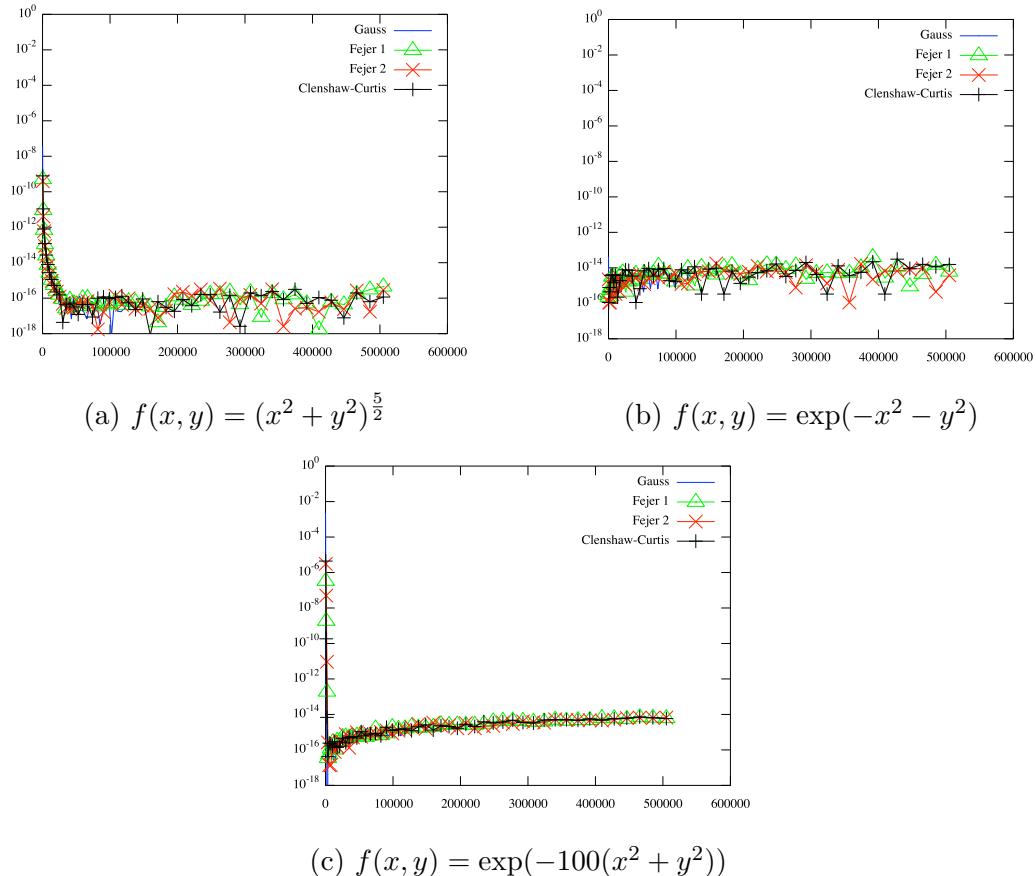


Figura 3.3: Errori assoluti nel calcolo degli integrali su una lente simmetrica con  $\omega = \frac{\pi}{4}$  e raggio 1, al variare del numero di nodi.

di:

$$I(\omega, r) = \int_{-\omega}^{\omega} \int_{-1}^1 (Rt(\cos \theta - \cos \omega) + R \sin \theta + 2)^m R^2 \cos \theta (\cos \theta - \cos \omega) dt d\omega = \\ (3.12)$$

$$= \int_{-\omega}^{\omega} R \cos \theta \left[ \frac{(Rt(\cos \theta - \cos \omega) + R \sin \theta + 2)^{m+1}}{m+1} \right]_{t=-1}^{t=1} d\omega \quad (3.13)$$

che è facilmente calcolabile tramite una formula di quadratura trigonometrica, per esempio `trigquad` con precisione pari a  $n = 200$ .

Innanzitutto, bisogna verificare che i codici ottenuti abbiano un errore accettabile. La tabella 3.4 mostra gli errori relativi, massimi e medi su 30 esperimenti, del calcolo di  $\iint_{\Omega} (x + y + 2)^m dx dy$  con  $\omega = \frac{\pi}{4}$  e raggio  $r = 1$ , mediante le quattro regole finora studiate, applicate sia su  $[-\omega, \omega]$  che su  $[-1, 1]$  con grado di precisione  $m$ . Si noti che la funzione `gqsymmlens2` mostra errori relativi comparabili, se non addirittura inferiori, a quelli delle formule gaussiane.

$\omega$	$\pi/16$	$\pi/8$	$\pi/4$	$\pi/2$	$3\pi/4$	$7\pi/8$	$15\pi/16$
$E_{max}^{\text{Gauss}}$	$6.35 \cdot 10^{-15}$	$1.05 \cdot 10^{-14}$	$8.55 \cdot 10^{-15}$	$1.31 \cdot 10^{-14}$	$2.68 \cdot 10^{-14}$	$2.85 \cdot 10^{-14}$	$2.95 \cdot 10^{-14}$
$E_{av}^{\text{Gauss}}$	$2.12 \cdot 10^{-15}$	$3.03 \cdot 10^{-15}$	$2.48 \cdot 10^{-15}$	$4.12 \cdot 10^{-15}$	$6.52 \cdot 10^{-15}$	$7.35 \cdot 10^{-15}$	$6.89 \cdot 10^{-15}$
$E_{max}^{\text{Fejér1}}$	$3.82 \cdot 10^{-15}$	$1.88 \cdot 10^{-15}$	$1.68 \cdot 10^{-15}$	$2.15 \cdot 10^{-15}$	$1.34 \cdot 10^{-15}$	$2.97 \cdot 10^{-15}$	$2.44 \cdot 10^{-15}$
$E_{av}^{\text{Fejér1}}$	$1.12 \cdot 10^{-15}$	$6.74 \cdot 10^{-16}$	$4.49 \cdot 10^{-16}$	$7.50 \cdot 10^{-16}$	$8.26 \cdot 10^{-16}$	$1.64 \cdot 10^{-15}$	$1.28 \cdot 10^{-15}$
$E_{max}^{\text{Fejér2}}$	$2.75 \cdot 10^{-15}$	$2.82 \cdot 10^{-15}$	$1.18 \cdot 10^{-15}$	$2.04 \cdot 10^{-15}$	$1.57 \cdot 10^{-15}$	$2.94 \cdot 10^{-15}$	$2.43 \cdot 10^{-15}$
$E_{av}^{\text{Fejér2}}$	$1.08 \cdot 10^{-15}$	$7.81 \cdot 10^{-16}$	$4.48 \cdot 10^{-16}$	$6.80 \cdot 10^{-16}$	$7.47 \cdot 10^{-16}$	$1.43 \cdot 10^{-15}$	$9.86 \cdot 10^{-16}$
$E_{max}^{\text{ClenC}}$	$3.81 \cdot 10^{-15}$	$2.07 \cdot 10^{-15}$	$1.61 \cdot 10^{-15}$	$3.07 \cdot 10^{-15}$	$4.20 \cdot 10^{-15}$	$5.64 \cdot 10^{-15}$	$5.40 \cdot 10^{-15}$
$E_{av}^{\text{ClenC}}$	$1.03 \cdot 10^{-15}$	$6.00 \cdot 10^{-16}$	$4.25 \cdot 10^{-16}$	$1.10 \cdot 10^{-15}$	$1.66 \cdot 10^{-15}$	$1.86 \cdot 10^{-15}$	$1.71 \cdot 10^{-15}$

Tabella 3.4: Errori relativi massimi ( $E_{max}$ ) e medi ( $E_{av}$ ) dell'integrazione di  $(x + y + 2)^m$  su un dominio a forma di lente circolare simmetrica con raggio 1 e semiangolo  $\omega$ , mediante diverse formule (applicate sull'arco e su  $[-1, 1]$  contemporaneamente) con grado di precisione  $m$ . Gli esperimenti sono stati effettuati per  $m = 5, 10, 15, \dots, 100$ .

Per quanto riguarda i tempi di esecuzione, i risultati sono esposti nella tabella 3.5, la quale mostra che i tempi di calcolo delle nuove regole sono simili a quelle gaussiane, in cui è stata utilizzata la funzione `trigauss`. È però da ricordare che una parte preponderante dei programmi usati per calcolare i pesi e i nodi della quadratura trigonometrica è rappresentata dal codice necessario per calcolare

$m$	Fejér 1	Fejér 2	Clenshaw-Curtis	Gauss
2	$3.486 \cdot 10^{-4}$	$2.958 \cdot 10^{-4}$	$3.958 \cdot 10^{-4}$	$8.805 \cdot 10^{-4}$
4	$3.574 \cdot 10^{-4}$	$2.947 \cdot 10^{-4}$	$3.430 \cdot 10^{-4}$	$8.620 \cdot 10^{-4}$
8	$3.890 \cdot 10^{-4}$	$3.863 \cdot 10^{-4}$	$3.810 \cdot 10^{-4}$	$1.083 \cdot 10^{-3}$
16	$5.286 \cdot 10^{-4}$	$4.416 \cdot 10^{-4}$	$4.783 \cdot 10^{-4}$	$1.380 \cdot 10^{-3}$
32	$7.574 \cdot 10^{-4}$	$8.365 \cdot 10^{-4}$	$8.643 \cdot 10^{-3}$	$2.131 \cdot 10^{-3}$
64	$1.887 \cdot 10^{-3}$	$1.989 \cdot 10^{-3}$	$2.063 \cdot 10^{-3}$	$3.716 \cdot 10^{-3}$
128	$6.614 \cdot 10^{-3}$	$6.227 \cdot 10^{-3}$	$6.903 \cdot 10^{-3}$	$9.445 \cdot 10^{-3}$
256	$2.231 \cdot 10^{-2}$	$2.547 \cdot 10^{-2}$	$1.417 \cdot 10^{-2}$	$3.079 \cdot 10^{-2}$
512	$8.527 \cdot 10^{-2}$	$8.264 \cdot 10^{-2}$	$7.945 \cdot 10^{-2}$	$1.410 \cdot 10^{-1}$

Tabella 3.5: Valori medi (su 30 esperimenti) del tempo di calcolo in secondi dell'integrale di  $(x + y + 2)^m$  su una lente circolare simmetrica con  $\omega = \frac{\pi}{4}$  e raggio 1, con grado di precisione  $m$ .

i momenti tramite l'algoritmo di Olver, ed essa è una parte comune a tutti i programmi. Se si confrontano i tempi di esecuzione della parte non in comune, i metodi non gaussiani si dimostrano più veloci di anche due ordini di grandezza, come mostrato nella tabella 3.6 a pagina 40.

Inoltre, per costruire la griglia di quadratura sul dominio è stata utilizzata la funzione `meshgrid`, la quale ha tempi di calcolo che dipendono dal numero di nodi utilizzati. Per esempio, la tabella 3.7 a pagina 40 mostra i tempi di calcolo di tale funzione all'interno di `gqsymmlens2`. La differenza fra i valori del caso gaussiano e gli altri è dovuta al fatto che la funzione `gauss` è stata invocata su  $\lceil \frac{m+1}{2} \rceil$  nodi, mentre sulle altre l'input è stato di  $m + 1$ .

Essendo presente una porzione di codice che richiede tempi di calcolo non indifferenti rispetto al totale, l'effetto delle regole di Fejér e di Clenshaw-Curtis viene nascosto.

Possibili miglioramenti di questi codici permettono di aumentare la performance delle nuove regole, che comunque sono applicabili anche con un grado di precisione maggiore di 512.

$m$	Fejér 1	Fejér 2	Clenshaw-Curtis	Gauss
2	$2.328 \cdot 10^{-4}$	$1.486 \cdot 10^{-4}$	$1.805 \cdot 10^{-4}$	$1.879 \cdot 10^{-4}$
4	$1.679 \cdot 10^{-4}$	$1.097 \cdot 10^{-4}$	$1.319 \cdot 10^{-4}$	$1.426 \cdot 10^{-4}$
8	$1.665 \cdot 10^{-4}$	$1.134 \cdot 10^{-4}$	$1.281 \cdot 10^{-4}$	$1.479 \cdot 10^{-4}$
16	$1.809 \cdot 10^{-4}$	$1.164 \cdot 10^{-4}$	$1.397 \cdot 10^{-4}$	$1.632 \cdot 10^{-4}$
32	$1.606 \cdot 10^{-4}$	$1.077 \cdot 10^{-4}$	$1.197 \cdot 10^{-4}$	$1.730 \cdot 10^{-4}$
64	$1.647 \cdot 10^{-4}$	$1.062 \cdot 10^{-4}$	$1.264 \cdot 10^{-4}$	$2.822 \cdot 10^{-4}$
128	$1.898 \cdot 10^{-4}$	$1.158 \cdot 10^{-4}$	$1.354 \cdot 10^{-4}$	$1.032 \cdot 10^{-3}$
256	$2.311 \cdot 10^{-4}$	$1.424 \cdot 10^{-4}$	$1.534 \cdot 10^{-4}$	$6.576 \cdot 10^{-3}$
512	$3.096 \cdot 10^{-4}$	$1.961 \cdot 10^{-4}$	$1.999 \cdot 10^{-4}$	$1.961 \cdot 10^{-2}$

Tabella 3.6: Valori medi (su 30 esperimenti) del tempo di calcolo in secondi dei nodi e dei pesi della quadratura trigonometrica all'interno di *gqsymmlens2* con grado di precisione  $m$ , una volta noto il vettore dei momenti di Chebyshev ( $\omega = \frac{\pi}{4}$ ).

Gauss	Fejér 1	Fejér 2	Clenshaw-Curtis
$1.632 \cdot 10^{-3}$	$3.413 \cdot 10^{-3}$	$3.392 \cdot 10^{-3}$	$3.411 \cdot 10^{-3}$

Tabella 3.7: Valori medi (su 30 esperimenti) del tempo di calcolo in secondi delle due funzioni *meshgrid* invocate in *gqsymmlens2* con  $m = 500$ .

# Capitolo 4

## Appendice: codici usati nei test

### 4.1 Funzione peso di Gegenbauer

Per l'esecuzione dei programmi sotto riportati è necessario disporre anche delle funzioni `fejer1`, `fejer2`, `clenshawcurtis` così come definite nel capitolo 2. Per il calcolo della formula gaussiana sono state usate le funzioni `r_jacobi` e `gauss` dalla suite OPQ di W. Gautschi [9].

#### Calcolo dei momenti dei polinomi di Chebyshev di primo tipo, pesati con la funzione di Gegenbauer

```
function mg=gegenmom(n,lambda)

% Input: n = massimo grado dei momenti
%         lambda = parametro della funzione di Gegenbauer (> -0.5)
%
% Output: mg = vettore dei momenti (da 0 a n, con n+1 elementi)

mg=zeros(n+1,1);
mg(1)=gamma(lambda+0.5)*sqrt(pi)/gamma(lambda+1);
g=-lambda/(1+lambda);
for j=3:2:length(mg)
    mg(j)=mg(1)*g;
    g=g*((j+1)/2-1-lambda)/(((j+1)/2)+lambda);
end
```

## Misura dei cputime

```
function tm=gegentest(n,lambda,nesp)

% Input: n = massimo valore da testare
%         lambda = parametro funzione di Gegenbauer (lambda > -0.5)
%         nesp = numero ripetizioni esperimento (per le medie)
%
% Output: tm = array dei cputime
%          1a colonna = numero nodi
%          2a colonna = cputime medio per Fejer 1
%          3a colonna = cputime medio per Fejer 2
%          4a colonna = cputime medio per Clenshaw-Curtis

tm=zeros(n,5);

for j=1:n
    tm(j,1)=2^j;

    for k=1:nesp
        tic;
        mom=gegenmom(2^j-1,lambda);
        [x,w]=fejerI(2^j,mom);
        time=toc;
        tm(j,2)=tm(j,2)+time;
    end
    tm(j,2)=tm(j,2)/nesp;

    for k=1:nesp
        tic;
        mom=gegenmom(2^j-1,lambda);
        [x,w]=fejerII(2^j,mom);
        time=toc;
        tm(j,3)=tm(j,3)+time;
    end
    tm(j,3)=tm(j,3)/nesp;

    for k=1:nesp
        tic;
        mom=gegenmom(2^j-1,lambda);
```

```

[x,w]=clenshaw_curtis(2^j,mom);
time=toc;
tm(j,4)=tm(j,4)+time;
end
tm(j,4)=tm(j,4)/nesp;

for k=1:nsp
tic;
ab=r_jacobi(2^j,lambda-0.5,lambda-0.5);
xw=gauss(2^j,ab);
time=toc;
tm(j,5)=tm(j,5)+time;
end
tm(j,5)=tm(j,5)/nsp;
end

```

## 4.2 Quadratura trigonometrica

Si suppone nota una funzione `momgen` che generi i momenti rispetto alla funzione peso  $w$  descritta in (3.6). Per il calcolo della regola gaussiana si è utilizzata la funzione `trigauss`, realizzata da G. Da Fies e M. Vianello [3].

### Calcolo dei nodi di Fejér e Clenshaw-Curtis per la quadratura trigonometrica

```

function [twf1,twf2,twcc,twg,t]=testnp(n,omega,nsp)

% Test del calcolo dei momenti trigonometrici mediante Fejer 1,
%   Fejer 2, Clenshaw-Curtis, Gauss
%
% Input: n = numero di prove
%         omega = semiangolo
%         nsp = numero di esperimenti per ogni prova
%
% Output: twf1 = array dei nodi e dei pesi per Fejer 1
%         twf2 = array dei nodi e dei pesi per Fejer 2
%         twcc = array dei nodi e dei pesi per Clenshaw-Curtis
%         twg = array dei nodi e dei pesi per Gauss

```

```

%           t = array dei tempi, in secondi
%           (1a colonna: Fejer 1, 2a colonna: Fejer 2,
%           3a colonna: Clenshaw-Curtis, 4a colonna: Gauss)

t=zeros(n,4);

for j=1:n
    t(j,1)=2*2^j+1;

    time = 0;
    for k=1:nsp          % Fejer 1
        tic;
        twf1=trigquadf1(2^j,-omega,omega);
        time=time+toc;
    end
    t(j,2)=time/nsp;

    time = 0;
    for k=1:nsp          % Fejer 2
        tic;
        twf2=trigquadf2(2^j,-omega,omega);
        time=time+toc;
    end
    t(j,3)=time/nsp;

    time = 0;
    for k=1:nsp          % Clenshaw-Curtis
        tic;
        twcc=trigquadcc(2^j,-omega,omega);
        time=time+toc;
    end
    t(j,4)=time/nsp;

    time = 0;
    for k=1:nsp          % Gauss
        tic;
        twg=trigauss(2^j,-omega,omega);
        time=time+toc;
    end
    t(j,5)=time/nsp;

```

## Confronto della precisione su $\cos(kx)$ delle formule trigonometriche di Fejér e Clenshaw-Curtis

```
function [eamax,ermax]=testescos(n,omega)

eamax=zeros(n,5);
ermax=zeros(n,5);

for j=1:n
    ermax(j,1)=2^j;
    eamax(j,1)=2^j;

    int0=zeros(2^j+1,1);
    intes=zeros(2^j+1,1);
    tw=trigauss(2^j,-omega,omega);
    int0(1,1)=sum(tw(:,2));
    intes(1,1)=2*omega;
    for k=1:2^j
        int0(k+1,1)=tw(:,2)'*cos(k.*tw(:,1));
        intes(k+1,1)=2*sin(k.*omega)./k;
    end
    errass=abs(int0-intes);
    eamax(j,2)=max(errass);
    errrel=errass./abs(intes);
    ermax(j,2)=max(errrel);

    int1=zeros(2^j+1,1);
    intes=zeros(2^j+1,1);
    tw2=trigquadf1(2^j,-omega,omega);
    int1(1,1)=sum(tw2(:,2));
    intes(1,1)=2*omega;
    for k=1:2^j
        int1(k+1,1)=tw2(:,2)'*cos(k.*tw2(:,1));
        intes(k+1,1)=2*sin(k.*omega)./k;
    end
    errass=abs(int1-intes);
    errrel=errass./abs(intes);
    eamax(j,3)=max(errass);
    ermax(j,3)=max(errrel);
```

```

int2=zeros(2^j+1,1);
intes=zeros(2^j+1,1);
tw2=trigquadf2(2^j,-omega,omega);
int2(1,1)=sum(tw2(:,2));
intes(1,1)=2*omega;
for k=1:2^j
    int2(k+1,1)=tw2(:,2)'*cos(k.*tw2(:,1));
    intes(k+1,1)=2*sin(k.*omega)./k;
end
errass=abs(int2-intes);
errrel=errass./abs(intes);
eamax(j,4)=max(errass);
ermax(j,4)=max(errrel);

int3=zeros(2^j+1,1);
intes=zeros(2^j+1,1);
tw2=trigquadcc(2^j,-omega,omega);
int3(1,1)=sum(tw2(:,2));
intes(1,1)=2*omega;
for k=1:2^j
    int3(k+1,1)=tw2(:,2)'*cos(k.*tw2(:,1));
    intes(k+1,1)=2*sin(k.*omega)./k;
end
errass=abs(int3-intes);
errrel=errass./abs(intes);
ermax(j,5)=max(errrel);
eamax(j,5)=max(errass);

end

```

## Confronto della precisione su $\sin(kx)$ delle formule trigonometriche di Fejér e Clenshaw-Curtis

```

function [eamax]=testessin(n,omega)

eamax=zeros(n,5);

for j=1:n
    eamax(j,1)=2^j;

```

```

int0=zeros(2^j+1,1);
tw=trigauss(2^j,-omega,omega);
for k=1:2^j
    int0(k+1,1)=tw(:,2)'*sin(k.*tw(:,1));
end
intes=zeros(size(int0));
errass=abs(int0-intes);
eamax(j,2)=max(errass);

int1=zeros(2^j+1,1);
tw2=trigquadf1(2^j,-omega,omega);
for k=1:2^j
    int1(k+1,1)=tw2(:,2)'*sin(k.*tw2(:,1));
end
intes=zeros(size(int1));
errass=abs(int1-intes);
eamax(j,3)=max(errass);

int2=zeros(2^j+1,1);
tw2=trigquadf2(2^j,-omega,omega);
for k=1:2^j
    int2(k+1,1)=tw2(:,2)'*sin(k.*tw2(:,1));
end
intes=zeros(size(int2));
errass=abs(int2-intes);
eamax(j,4)=max(errass);

int3=zeros(2^j+1,1);
tw2=trigquadcc(2^j,-omega,omega);
for k=1:2^j
    int3(k+1,1)=tw2(:,2)'*sin(k.*tw2(:,1));
end
intes=zeros(size(int3));
errass=abs(int3-intes);
eamax(j,5)=max(errass);

end

```

## Prima verifica della precisione di gqsymmlens2

```
function [intesatto,int,t,ea,er,dt]=testgq(n,omega,in1,in2,in3,in4,nesp)

% Confronto di due formule di quadratura per la funzione
%   cos(x)+sin(y)
%
% Input: n = numero massimo di nodi (2^(1:n))
%         omega = semiangolo
%         in1, in2 = scelta 1a formula su arco e raggio risp.
%         in3, in4 = scelta 2a formula su arco e raggio risp.
%         nesp = numero di esperimenti (per media dei cputime)
%
% Scelta della formula: 0 = Gauss      ; 1 = Fejer I
%                      2 = Fejer II   ; 3 = Clenshaw-Curtis
%
% Output: intesatto = valore esatto dell'integrale (tramite trigquad)
%          int = valori degli integrali al variare dei nodi
%          t = cputime al variare del numero di nodi
%          ea = errori assoluti delle due formule
%          er = errori relativi delle due formule
%          dt = differenza cputime (1a formula - 2a formula)

r=1;      % raggio

tw=trigquad(200,-omega,omega);
f=r*cos(tw(:,1)).*(sin(r.*cos(tw(:,1))-r.*cos(omega))-
    sin(r.*cos(omega)-r.*cos(tw(:,1)))); 
intesatto=f'*tw(:,2);

for j=1:n

    int(j,1)=2^j;
    t(j,1)=int(j,1);
    ea(j,1)=int(j,1);
    er(j,1)=int(j,1);
    dt(j,1)=int(j,1);

    time=0;
    for k=1:nesp
```

```

[int1,t1]=testint(int(j,1),omega,r,in1,in2);
int(j,2)=int1;
time=time+t1;
end
t(j,2)=time/nesp;

time=0;
for k=1:nesp
    [int2,t2]=testint(int(j,1),omega,r,in3,in4);
    int(j,3)=int2;
    time=time+t2;
end
t(j,3)=time/nesp;

ea(j,2)=abs(intesatto-int(j,2));
ea(j,3)=abs(intesatto-int(j,3));
er(j,2)=ea(j,2)/abs(intesatto);
er(j,3)=ea(j,3)/abs(intesatto);
dt(j,2)=t(j,2)-t(j,3);

end

function [int,t]=testint(n,omega,r,in1,in2)

tic;
xyw=gqsymmlens2(n,omega,r,in1,in2);
f=cos(xyw(:,1))+sin(xyw(:,2));      % funzione integranda
int=xyw(:,3)'*f;
t=toc;

```

## Integrazione di $(x + y + 2)^n$ su lenti circolari

```

function [intes,int,t,ea,er,dt]=testpol(n,omega,in1,in2,in3,in4,nesp)

% Confronto di due formule di quadratura per la funzione  $(x+y+2)^n$ 
%
% Input: n = numero massimo di nodi ( $2^{(1:n)}$ )
%         omega = semiangolo

```

```

%
%      in1, in2 = scelta formula su arco e raggio della 1a formula
%      in3, in4 = scelta formula su arco e raggio della 2a formula
%      nesp = numero di esperimenti (per la media dei cputime)
%
% Scelta della formula:  0 = Gauss          ;   1 = Fejer I
%                      2 = Fejer II         ;   3 = Clenshaw-Curtis
%
% Output: intes = valore esatto dell'integrale (tramite trigquad)
%          int = valori degli integrali al variare dei nodi
%          t = cputime al variare del numero di nodi
%          ea = errori assoluti delle formule
%          er = errori relativi delle formule
%          dt = differenza cputime (1a form. - 2a)

esp=n;    % esponente della funzione integranda (n)
r=10;     % raggio
tw=trigquad(200,-omega,omega);
f=r*cos(tw(:,1)).*((r*cos(tw(:,1))-r*cos(omega)+r*sin(tw(:,1))+2).^
(esp+1)-(r*cos(omega)-r*cos(tw(:,1))+r*sin(tw(:,1))+2).^(esp+1))
./ (esp+1);
intes=f'*tw(:,2);
for j=1:n
    int(j,1)=2^j;
    t(j,1)=int(j,1);
    ea(j,1)=int(j,1);
    er(j,1)=int(j,1);
    dt(j,1)=int(j,1);
    time=0;
    for k=1:nesp
        [int1,t1]=testintpol(esp,int(j,1),omega,r,in1,in2);
        int(j,2)=int1;
        time=time+t1;
    end
    t(j,2)=time/nesp;
    time=0;
    for k=1:nesp
        [int2,t2]=testintpol(esp,int(j,1),omega,r,in3,in4);
        int(j,3)=int2;
        time=time+t2;
    end
end

```

```

t(j,3)=time/nesp;
ea(j,2)=abs(intes-int(j,2));
ea(j,3)=abs(intes-int(j,3));
er(j,2)=ea(j,2)/abs(intes);
er(j,3)=ea(j,3)/abs(intes);
dt(j,2)=t(j,2)-t(j,3);
end

function [int,t]=testintpol(esp,n,omega,r,in1,in2)
tic;
xyw=gqsymmlens2(n,omega,r,in1,in2);
f=(xyw(:,1)+xyw(:,2)+2).^esp;      % funzione integranda
int=xyw(:,3)'*f;
t=toc;

```



# Bibliografia

- [1] L. Bos, M. Vianello, “Subperiodic trigonometric interpolation and quadrature”, *Applied Mathematics and Computation*, n. 218 (2012), pagg. 10630-10638. Disponibile presso <http://www.math.unipd.it/~marcov/CAApbl.html>
- [2] C. W. Clenshaw, A. R. Curtis, “A method for numerical integration on an automatic computer”, *Numerische Mathematik*, vol. 2 (1960), pagg. 197-205.
- [3] G. Da Fies, M. Vianello, *Matlab function for subperiodic trigonometric Gaussian quadrature - version 1.0*. Disponibile presso <http://www.math.unipd.it/~marcov/CAAsoft.html>
- [4] G. Da Fies, M. Vianello, “Trigonometric Gaussian quadrature on subintervals of the period”, *Electronic Transactions on Numerical Analysis*, n. 39 (2012), pagg. 102-112. Disponibile presso <http://www.math.unipd.it/~marcov/pdf/trigauss.pdf>
- [5] P. J. Davis, *Interpolation & Approximation*, Dover Publications Inc., New York, 1975.
- [6] P. J. Davis, P. Rabinowitz, *Methods of Numerical Integration*, Academic Press, Londra, 1975.
- [7] L. Fejér, “Mechanische Quadraturen mit positiven Cotesschen Zahlen”, *Mathematische Zeitschrift*, vol. 37, n. 1 (1933), pagg. 287-309.
- [8] L. Fejér, “On the infinite sequences arising in the theories of harmonic analysis, of interpolation, and of mechanical quadratures”, *Bulletin of the American Mathematical Society*, vol. 39, n. 8 (1933), pagg. 521-534.
- [9] W. Gautschi, *OPQ: A Matlab suite of programs for generating orthogonal polynomials and related quadrature rules*. Disponibile presso <http://www.cs.purdue.edu/archives/2002/wxg/codes/OPQ.html>
- [10] W. Gautschi, *Orthogonal Polynomials: Computation and Approximation*, Oxford University Press, Oxford, 2004.

- [11] W. M. Gentleman, “Implementing Clenshaw-Curtis quadrature”, *Communications of the ACM*, vol. 15, n. 5 (maggio 1972), pagg. 337-346.
- [12] D. B. Hunter, H. V. Smith, “A quadrature formula of Clenshaw-Curtis type for the Gegenbauer weight-function”, *Journal of Computational and Applied Mathematics*, vol. 177, n. 2 (maggio 2005), pagg. 389-400.
- [13] MathWorks, “Discrete Sine Transform”, *Matlab R2012a Product Documentation, Partial Differential Equation Toolbox*. Disponibile presso <http://www.mathworks.com/help/toolbox/pde/ug/dst.html>
- [14] MathWorks, “Inverse Discrete Cosine Transform”, *Matlab R2012a Product Documentation, Signal Processing Toolbox*. Disponibile presso <http://www.mathworks.com/help/toolbox/signal/ref/idct.html>
- [15] F. W. J. Olver, “Numerical solution of second order linear difference equations”, *Journal of Research of the National Bureau of Standards*, vol. 71B, n. 2-3 (1967), pagg. 111-129.
- [16] T. J. Rivlin, *The Chebyshev Polynomials*, John Wiley & Sons, New York, 1974.
- [17] I. H. Sloan, W. E. Smith, “Product-integration with the Clenshaw-Curtis and related points: Convergence properties”, *Numerische Mathematik*, vol. 30, n. 4 (1978), pagg. 415-428.
- [18] I. H. Sloan, W. E. Smith, “Product-integration with the Clenshaw-Curtis points: Implementation and error estimates”, *Numerische Mathematik*, vol. 34, n. 4 (1980), pagg. 387-401.
- [19] L. N. Trefethen, “Is Gauss quadrature better than Clenshaw-Curtis?”, *SIAM Review*, vol. 50, no. 1 (2008), pagg. 67-87.
- [20] J. Waldvogel, “Fast construction of the Fejér and Clenshaw-Curtis quadrature rules”, *BIT Numerical Mathematics*, vol. 46 (2006), pagg. 195-202.