5

Calcoliamo aree e volumi: formule di quadratura

5.1 Introduzione

Il calcolo di integrali si presenta assai di frequente nelle applicazioni della matematica. Basti pensare all'interpretazione geometrica dell'integrale come misura dell'area sottesa da una curva, o alla lunghezza di una linea, oppure ancora all'uso di integrali multipli nel calcolo di aree di superfici e di volumi. Protagonista principale del capitolo sarà il seguente integrale

$$I[f] \equiv \int_a^b f(x)\,dx \tag{5.1}$$

che, nel caso in cui $f(x) \geq 0$, rappresenta l'area denotata in grigio in Figura 5.1.

Esempio 5.1 [Progettazione di oggetti]
Uno studio di design ha portato alla progettazione del pezzo visualizzato in Figura 5.2 (potrebbe essere un bicchiere, un portaombrelli o una lampada, il lettore immagini quello che più gli aggrada). Dovendo produrre l'oggetto in questione, per motivi economici o per verificare che sia utile allo scopo per cui è stato progettato potrebbe essere indispensabile valutare la misura della superficie laterale (quindi quanto materiale si impiegherà) o del volume (cosa potrà contenere). ■

Esempio 5.2 [Lunghezza di una linea]
Una linea nello spazio, in $\mathbb{R}^3$, è descritta in forma parametrica da equazioni del tipo

$$x = x(t), \quad y = y(t), \quad z = z(t), \qquad t \in [t_a,t_b]. \tag{5.2}$$

La lunghezza dell'arco di linea che connette il punto $(x(t_a),y(t_a),z(t_a))$ al punto $(x(t_b),y(t_b),z(t_b))$ secondo il percorso determinato da (5.2) è data da [23]

$$L = \int_{t_a}^{t_b} \sqrt{(x'(t))^2 + (y'(t))^2 + (z'(t))^2}, \tag{5.3}$$

dove x', y' e z' indicano le derivate prime rispetto a t.

Ci siamo quindi ricondotti al calcolo di un integrale in una dimensione. ■

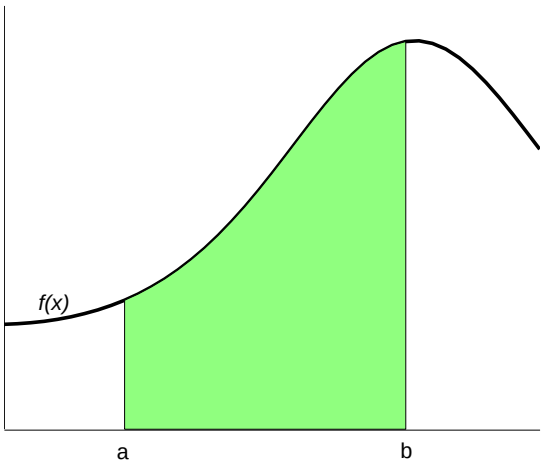

Figura 5.1 Relazione fra l'integrale e l'area sottesa da una curva.

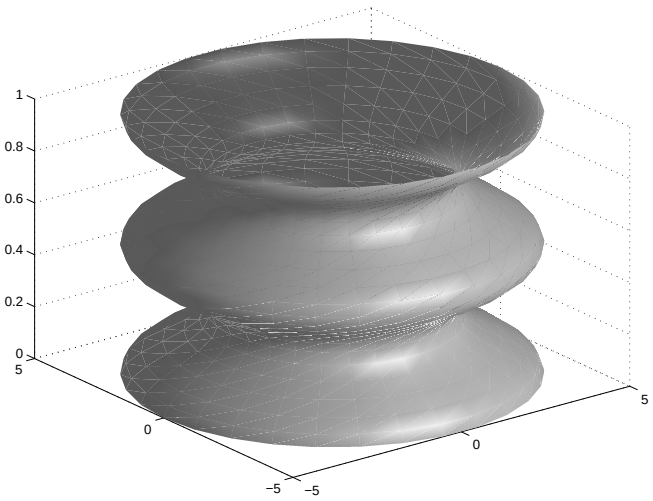

Figura 5.2 Un oggetto progettato di cui si vuole misurare il volume.

Esempio 5.3 [Equazioni differenziali]
Nella meccanica di un punto materiale di massa m in una dimensione, sotto l'azione di un potenziale $V(x)$, le equazioni del moto si possono "integrare" esattamente. Dalla conservazione dell'energia totale, infatti, segue l'equazione differenziale

$$\frac{1}{2}m(x'(t))^2 + V(x) = E, \tag{5.4}$$

dove E rappresenta l'energia totale. Questa relazione permette di "portare l'equazione alle quadrature", integrando infatti ricaviamo la legge oraria

$$t = t_0 \pm \int_{x_0}^{x} \left(\frac{2}{m}(E - V(\tilde{x}))\right)^{-1/2} d\tilde{x}, \tag{5.5}$$

dove il segno dipende dal segno della velocità iniziale. Nel caso di moti periodici, tale formula si applica al calcolo del periodo (vedi Figura 5.3)

$$T = 2\int_{a}^{b} \left(\frac{2}{m}(E - V(x))\right)^{-1/2} dx. \tag{5.6}$$

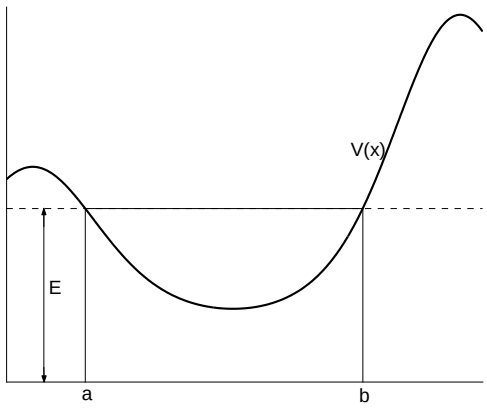

Figura 5.3 Moti periodici e calcolo del periodo

In molti altri problemi della meccanica, anche in più dimensioni, la soluzione delle equazioni del moto viene "riportata alle quadrature": cioè viene espressa mediante integrali. Un esempio classico è costituito dal problema dei due corpi soggetti a mutua interazione gravitazionale. La soluzione soddisfa le leggi di Keplero del moto dei pianeti, e si può esplicitare fornendo la traiettoria dei corpi e la legge oraria. Quest'ultima stabilisce la posizione dei corpi in funzione del tempo, ed è esprimibile mediante un integrale. ■

Esempio 5.4 [Valor medio di una funzione di variabile aleatoria]
Nel calcolo delle probabilità gli integrali intervengono tutte le volte che si vuole calcolare il valore medio di una funzione di una variabile aleatoria, della quale sia nota la densità di probabilità. L'integrale può essere in una o più dimensioni a

seconda che la variabile aleatoria sia monovariata o multivariata (si veda il Capitolo 7).

Sia X una variabile aleatoria a valori in $\mathbb{R}$ con densità $f(x)$, e sia g una funzione di X. Il valor medio di g è dato da

$$< g >= \int_{\mathbb{R}} f(x)g(x)\,dx. \tag{5.7}$$

■

Calcoli di integrali intervengono come passo intermedio nella applicazione di svariati metodi di approssimazione. Nel metodo dei minimi quadrati nel continuo, per esempio, la matrice dei coefficienti delle equazioni normali ha elementi che sono calcolati come prodotti scalari tra elementi delle funzioni della base discreta, e i termini noti del sistema si calcolano come prodotti scalari tra gli elementi della base e il termine noto. Mentre i coefficienti della matrice ammettono spesso una rappresentazione analitica, gli elementi del termine noto richiedono spesso l'approssimazione numerica di integrali. Un caso particolare di migliore approssimazione nel senso dei minimi quadrati si ha nel calcolo dei coefficienti di Fourier di una funzione periodica, e, più in generale, quando si vuole calcolare una approssimazione di una funzione in una base data.

In questo capitolo ci occupiamo principalmente del calcolo di integrali in una dimensione del tipo (5.1). Un cenno agli integrali in più dimensioni verrà fatto nel Paragrafo 5.6.

Dall'Analisi Matematica, si veda per esempio [22], sappiamo che, detta F una primitiva della funzione f (quindi $F'(x) = f(x)$), si ha

$$\int_a^b f(x)\,dx = F(b) - F(a). \tag{5.8}$$

Tale formula permette di calcolare analiticamente diversi integrali, ma spesso risulta insufficiente. Infatti le tecniche dell'analisi (per esempio integrazione per parti, e per sostituzione) spesso non bastano per ottenere una espressione esplicita della primitiva F in termini di combinazioni di funzioni elementari (polinomi e funzioni razionali, funzioni trigonometriche e loro inverse, funzioni esponenziali e logaritmi). Anche i sistemi di calcolo simbolico non aiutano ovviamente in queste situazioni: non si tratta di non avere a disposizione "trucchi di calcolo", per certe funzioni f si può dimostrare che non è possibile esprimere f come combinazione ei funzioni elementari. Inoltre spesso, anche se è possibile determinare una espressione analitica esplicita dell'integrale, questa risulta eccessivamente complicata e costosa da calcolare. In tutti questi casi è preferibile (o necessario) ricorrere a formule che forniscono direttamente una approssimazione numerica del valore dell'integrale.

Prima di addentrarci nella descrizione delle diverse formule di quadratura[1] vogliamo fare alcune considerazioni di carattere generale, che ci guideranno alla comprensione del testo, ed alla scelta della strategia più adatta per affrontare il calcolo di integrali.

[1] Il termine quadratura è preso dal Latino e deriva da un antico problema geometrico consistente nella costruzione di un quadrato con la stessa area di un dato cerchio utilizzando solo riga e compasso: costruzione che si è poi dimostrato essere impossibile.

La prima considerazione che facciamo è che molte formule di quadratura sono strettamente legate a formule di approssimazione. Infatti, se si riesce ad approssimare una funzione $f(x)$ con una funzione più semplice, $p(x)$, di cui si sa calcolare l'integrale, e se l'errore uniforme è piccolo, diciamo

$$||f-p||_\infty = \max_{x\in[a,b]} |f(x)-p(x)| < \varepsilon,$$

allora si ha automaticamente una stima sull'approssimazione dell'integrale:

$$\begin{aligned} |I[f]-I[p]| &= \left| \int_a^b f(x)dx - \int_a^b p(x)dx \right| \\ &= \left| \int_a^b (f(x)-p(x))dx \right| \leq \int_a^b |f(x)-p(x)|dx < \varepsilon(b-a). \end{aligned}$$

Se la funzione f è sufficientemente regolare, una buona approssimazione di può per esempio ottenere tramite un polinomio oppure una funzione spline (si veda il Capitolo 3).

Spesso è opportuno scrivere gli integrali nella forma

$$I[f] \equiv \int_a^b w(x)f(x)\,dx, \tag{5.9}$$

dove la funzione $w(x)$ (che di solito si assume non negativa) è detta *funzione peso*. Essa si suppone fissata, e viene scelta in modo che la funzione rimanente sia sufficientemente regolare, ma in modo che integrali del tipo $I[x^n]$ siano calcolabili analiticamente. In questo modo si "scaricano" eventuali singolarità della funzione integranda sulla funzione peso, e la funzione integranda risulterà ben approssimabile mediante funzioni di cui si sa calcolare l'integrale in maniera esatta (per esempio ancora polinomi).

La seconda osservazione di carattere generale è la seguente. Come vedremo ci sono formule con diverso grado di accuratezza e complessità. La scelta della formula più appropriata dipende anche dalla regolarità della funzione integranda (della f, nell'espressione (5.9)).

Se la funzione integranda non è sufficientemente regolare, è inutile (e costoso) utilizzare formule sofisticate ed accurate, poiché la rapidità con la quale una successione di formule di quadratura converge al risultato esatto in questi casi dipenderà più dalla regolarità della funzione che non dall'ordine di accuratezza della formula. In altri termini: prima di utilizzare una routine di quadratura a scatola chiusa è meglio "guardare in faccia" l'oggetto con cui si ha a che fare per evitare spiacevoli sorprese.

5.2 Formule interpolatorie

In questo paragrafo consideriamo formule di quadratura per l'approssimazione di integrali del tipo (5.9), aventi la seguente struttura

$$I_n[f] \equiv \sum_{i=0}^{n} w_i f(x_i), \tag{5.10}$$

dove $x_i, i = 0, \ldots, n$ sono detti *nodi* e $w_i, i = 0, \ldots, n$ sono i *pesi* della formula.
Generalmente i nodi sono interni all'intervallo $[a,b]$. Osserviamo che l'operatore $I[f]$ che definisce l'integrale e l'operatore $I_n[f]$ che definisce la formula di quadratura sono funzionali lineari, ossia operano in modo lineare sulle funzioni scelte come loro argomenti. Più formalmente, date due funzioni f e g, e due costanti reali α e β si ha

$$I[\alpha f + \beta g] = \alpha I[f] + \beta I[g], \quad I_n[\alpha f + \beta g] = \alpha I_n[f] + \beta I_n[g].$$

Come "misurare" la grandezza (norma) di tali funzionali? Si ripresenta un problema analogo a quello della definizione di una norma di matrici. Anche per la definizione di una norma per I e I_n ricorriamo a quanto viene lasciato in eredità dalle funzioni su cui operiamo, cioè alla norma che utilizziamo per le funzioni. Ricordiamo alcune norme di uso comune (supporremo che le espressioni scritte siano ben poste e finite) per la funzione f

$$\|f\|_\infty = \max_{a \le x \le b} |f(x)|, \quad \|f\|_p = \left(\int_a^b |f(x)|^p dx \right)^{1/p} \quad p > 0.$$

Le norme indotte dei funzionali I, I_n sono le seguenti ($p \in [1, +\infty]$)

$$\|I\|_p \equiv \max_{\|f\|_p = 1} |I[f]|, \quad \|I_n\|_p \equiv \max_{\|f\|_p = 1} |I_n[f]|.$$

La norma più comunemente utilizzata è la norma del massimo $\|f\|_\infty$, a tale norma ci riferiremo quando non specificheremo l'indice p.

5.2.1 Ordine polinomiale di una formula di quadratura

Definizione 5.1 *Diremo che la formula di quadratura (5.10) ha* ordine polinomiale m *se risulta esatta per tutti i polinomi di grado* m *e non lo è per almeno un polinomio di grado* $m+1$*, cioè se risulta*

$$I[f] = I_n[f], \forall f \in \mathbb{P}_m, \quad \exists f \in \mathbb{P}_{m+1} : I[f] \neq I_n[f].$$

Osserviamo che per la linearità del funzionale, condizione necessaria e sufficiente affiché una formula sia di ordine polinomiale m è che sia

$$I_n[x^k] = I[x^k], k = 0, \ldots, m, \quad I_n[x^{m+1}] \neq I[x^{m+1}].$$

Un elevato ordine polinomiale garantisce una buona accuratezza per il calcolo di integrali di funzioni che siano ben approssimabili da polinomi.

Il problema della convergenza, già accennato quando abbiamo trattato la successione di schemi interpolatori, si presenta anche nel caso di una successione di formule di quadratura.

Come vedremo, le condizioni di convergenza di una successione di formule di quadratura sono assai meno restrittive di quelle richieste dalla convergenza di una successione di schemi interpolatori. Riportiamo qui un teorema generale di convergenza. Pur considerando griglie uniformi, se riusciamo a controllare la norma di I_n si evitano fenomeni tipo quello di Runge presente nell'ambito dell'interpolazione polinomiale. In Figura 5.4 si mostra un possibile percorso di convergenza verso I,

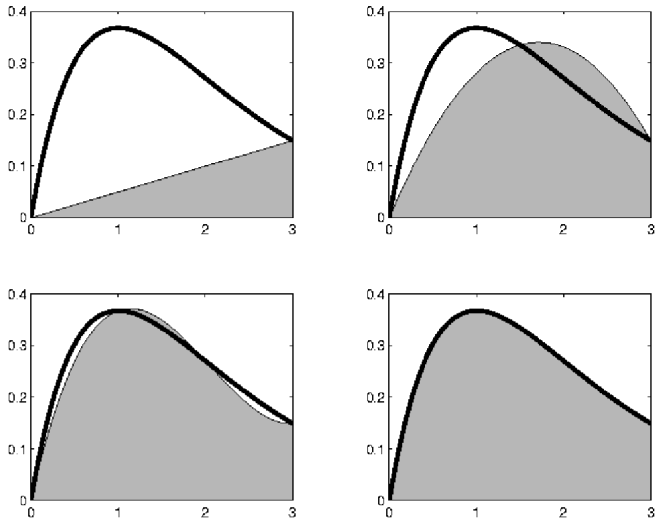

Figura 5.4 Storia della convergenza di I_n a I.

integrale della funzione f, tramite successive approssimazioni I_n i cui valori sono rappresentati graficamente tramite aree.

Teorema 5.1 *Consideriamo una successione di formule di quadratura I_n equilimitate, cioè tali che $\exists C > 0$ con $\|I_n\| \leq C \; \forall n$. Detto m_n l'ordine polinomiale di I_n, si ha che se $\lim_{n\to\infty} m_n = \infty$, allora*

$$\lim_{n\to\infty} |I_n[f] - I[f]| = 0 \quad \forall f \in C^0([a,b])$$

DIMOSTRAZIONE. Data una funzione $f \in C^0([a,b])$, esiste p_m^* il polinomio di grado m di miglior approssimazione uniforme per la funzione f [47]. Tale polinomio verifica

$$\|f - p_m^*\|_\infty \leq \|f - p_m\|_\infty, \;\; \forall p_m \in \mathbb{P}_m$$

ovverosia è il polinomio che sta più vicino uniformemente alla funzione f rispetto a tutti i polinomi di grado minore o uguale a m. Allora si ha

$$\begin{aligned} |I[f] - I_n[f]| &= |I[f] - I[p_{m_n}^*] + I_n[p_{m_n}^*] - I_n[f]| \\ &\leq |I[f - p_{m_n}^*]| + |I^n[p_{m_n}^* - f]| \\ &\leq (\|I\| + \|I_n\|)E_{m_n}[f] \leq (\|I\| + C)E_{m_n}[f] \end{aligned}$$

La prima uguaglianza è dovuta al fatto che $I[p_m^*] = I_n[p_m^*]$ poiché la formula è esatta per polinomi di grado m_n. La quantità $E_{m_n}[f] \equiv \max_{a\leq x\leq b} |f(x) - p_{m_n}^*(x)|$ è la norma della differenza fra la funzione f ed il suo polinomio di migliore approssimazione uniforme di grado m_n. Il Teorema di Weierstrass, riportato nel capitolo relativo all'approssimazione, assicura che tale norma tende a zero al tendere all'infinito del grado del polinomio. Poiché per ipotesi $m_n \to \infty$ per $n \to \infty$, allora si ha

$$\lim_{n\to\infty} |I[f] - I_n[f]| \to 0.$$

■

Risulta interessante calcolare le norme dei funzionali $\|I\|$ e $\|I_n\|$.
Sia $\|f\|_\infty = 1$, allora

$$|I[f]| = \left| \int_a^b w(x) f(x)\, dx \right| \leq \int_a^b w(x)|f(x)|\, dx \leq \int_a^b w(x)\, dx$$

per cui

$$\|I\| \leq \int_a^b w(x)\, dx.$$

Se poi scegliamo $f(x) \equiv 1$, si ottiene

$$\|I\| \geq |I[f]| = \left| \int_a^b w(x)\, dx \right| = \int_a^b w(x)\, dx,$$

per cui

$$\|I\| = \int_a^b w(x)\, dx.$$

Sia nuovamente $\|f\|_\infty = 1$, ne segue che

$$|I_n[f]| = \left| \sum_{i=0}^n w_i f(x_i) \right| \leq \sum_{i=0}^n |w_i||f(x_i)| \leq \sum_{i=0}^n |w_i|$$

per cui

$$\|I_n\| \leq \sum_{i=0}^n |w_i|.$$

Se poi scegliamo f tale che $\|f\|_\infty = 1$ e $f(x_i) = \text{sign}(w_i)$, $i = 0, \ldots, n$, allora otteniamo[2]

$$|I_n[f]| = \left| \sum_{i=0}^n w_i \,\text{sign}(w_i) \right| = \sum_{i=0}^n |w_i|,$$

per cui

$$\|I_n\| = \sum_{i=0}^n |w_i|.$$

Osserviamo che se $m_n \geq 0$ (che ovviamente possiamo assumere sempre vera), e se $w_i \geq 0$, $\forall i = 0, \ldots, n$, allora si ha

$$\|I_n\| = \sum_{i=0}^n |w_i| = \sum_{i=0}^n w_i = \int_a^b w(x)\, dx = \|I\|.$$

Pertanto una famiglia di formule di quadratura con pesi w_i non negativi è equilimitata da $\|I\|$.

[2]Ricordiamo che la funzione segno $\text{sign}(x)$ vale 1 se $x > 0$ e vale -1 nel caso in cui $x < 0$.

5.2.2 Formule interpolatorie

Definizione 5.2 *Una formula di quadratura si dice* interpolatoria *se è del tipo*

$$I_n[f] = I[p_n],$$

dove con p_n intendiamo il polinomio di interpolazione della funzione f sui nodi x_i,$i = 0,\ldots,n$.

Utilizzando la rappresentazione di Lagrange del polinomio interpolante abbiamo

$$\begin{aligned} I_n[f] &= I[p_n] = I\left[\sum_{i=0}^{n} L_i(x) f(x_i)\right] \\ &= \int_a^b w(x)\left(\sum_{i=0}^{n} L_i(x) f(x_i)\right) dx = \sum_{i=0}^{n}\left(\int_a^b w(x) L_i(x)\, dx\right) f(x_i), \end{aligned}$$

dove $L_i(x)$ è l'i-esimo polinomio elementare di Lagrange.

Confrontando questa espressione con l'espressione della generica formula di quadratura (5.10) si vede che effettivamente $I[p_n]$ è una formula di quadratura del tipo (5.10) con pesi

$$w_i = \int_a^b w(x) L_i(x)\, dx. \tag{5.11}$$

Dalla definizione è evidente che I_n ha ordine polinomiale $m \geq n$, in quanto risulta esatta per polinomi[3] di grado n.

Facciamo ora vedere che se I_n ha ordine polinomiale $m \geq n$ allora essa individua una formula di quadratura interpolatoria. Infatti, se $m \geq n$, la formula risulta esatta quando viene applicata ai polinomi elementari di Lagrange L_i, che sono polinomi di grado n

$$\int_a^b w(x) L_i(x)\, dx = \sum_{j=0}^{n} w_j L_i(x_j) = w_i.$$

L'ultima uguaglianza segue dalla proprietà dei polinomi elementari di Lagrange $L_i(x_j) = \delta_{ij}$. Utilizzando l'espressione appena trovata si ha

$$\begin{aligned} I_n[f] &= \sum_{i=0}^{n} w_i f(x_i) = \sum_{i=0}^{n}\left(\int_a^b w(x) L_i(x)\, dx\right) f(x_i) \\ &= \int_a^b w(x)\left(\sum_{i=0}^{n} L_i(x) f(x_i)\right) dx = I[L_n(f)]. \end{aligned}$$

Possiamo dunque riassumere questo risultato nel seguente Teorema.

Teorema 5.2 *Una formula di quadratura I_n è interpolatoria se e solo se è di ordine polinomiale $m_n \geq n$.*

[3] Ricordiamo che il polinomio interpolante di Lagrange $p_n = L_n(f)$ è un operatore di proiezione sullo spazio $\mathbb{P}_n$ dei polinomi algebrici di grado minore o uguale ad n, cioè risulta $L_n(p) = p$ se $p \in \mathbb{P}_n$.

Stime dell'errore

$$\mathcal{E}_n = I[f] - I_n[f]$$

per le formule di quadratura interpolatorie si possono ottenere a partire dalle stime sull'errore nelle formula di interpolazione. Infatti, dalla relazione

$$f(x) = p_n(x) + R_n(x),$$

segue

$$I[f] - I_n[f] = \int_a^b w(x)(f(x) - p_n(x))\,dx = \int_a^b w(x)R_n(x)\,dx.$$

Informazioni sul resto $R_n(x)$ si traducono quindi in informazioni sull'errore $\mathcal{E}_n$.

5.2.3 Formule di Newton-Cotes

Sono formule interpolatorie costruite su nodi equidistribuiti. Esse sono di due tipi: aperte e chiuse. Nelle formule aperte l'intervallo $[a,b]$ è diviso in $n+2$ parti uguali e gli estremi non sono inclusi tra i nodi

$$x_i = a + (i+1)h, \quad i = 0,\ldots,n, \quad h = (b-a)/(n+2).$$

Nelle formule chiuse l'intervallo $[a,b]$ è diviso in n parti uguali dai nodi

$$x_i = a + ih, \quad i = 0,\ldots,n, \quad h = (b-a)/n.$$

Queste formule hanno un ordine di accuratezza $m = n+1$ per n pari e $m = n$ per n dispari. Indicando con

$$r_n(x) = x(x-1)\cdots(x-n),$$

il cosiddetto *polinomio fattoriale*, una stima dell'errore è data da

n **pari**

$$\mathcal{E}_n = K\frac{f^{n+2}(\xi)}{(n+2)!}h^{n+3}, \quad \text{con } \xi \in [a,b]$$

dove

$$K = \begin{cases} \int_{-1}^{n+1} x r_n(x)\,dx & \text{per il tipo aperto,} \\ \int_0^n x r_n(x)\,dx & \text{per il tipo chiuso.} \end{cases}$$

n **dispari**

$$\mathcal{E}_n = K\frac{f^{n+1}(\xi)}{(n+1)!}h^{n+2}, \quad \text{con } \xi \in [a,b]$$

dove

$$K = \begin{cases} \int_{-1}^{n+1} r_n(x)\,dx & \text{per il tipo aperto,} \\ \int_0^n r_n(x)\,dx & \text{per il tipo chiuso.} \end{cases}$$

I programmi MATLAB `NewtonCotesA.m` e `NewtonCotesC.m` calcolano i pesi e le costanti K per formule di Newton-Cotes aperte e chiuse e salvano i risultati in appositi files. Per le formule aperte avremo lo script MATLAB che crea i files `NewtonCotesA.dat` e `NewtonCotesAK.dat`

```
% NewtonCotesA.m
%
% Calcola i pesi e le costanti di errore nelle formule di
% Newton-Cotes aperte con indice n da 1 a NP
clear
syms x
NP = 8;
fidw = fopen('NewtonCotesA.dat','w');
fidk = fopen('NewtonCotesAK.dat','w');
for n=1:NP
   h = 2/(n+2);
   for i=0:n
      w(i+1) = int(lag(x,i,n),-1,n+1);
   end
   if mod(n,2)==0
       K = int(x*rn(x,n),-1,n+1)/factorial(n+2);
   else
       K = int(rn(x,n),-1,n+1)/factorial(n+1);
   end
   % Conversione da variabili simboliche a variabili
   % in doppia precisione
   wi = h*double(w(1:n+1));
   xi = -1+h:h:1-h;
   Ki = double(K);
   disp(wi);
   fprintf(fidw,'%20.16f %20.16f\n',[xi;wi]);
   fprintf(fidw,'\n');
   fprintf(fidk,'%2d %22.16f\n',[n;Ki]);
end
fclose(fidw);
fclose(fidk);
```

ed analogamente per le formule chiuse lo script che crea i files `NewtonCotesC.dat` e `NewtonCotesCK.dat`

```
% NewtonCotesC.m
%
% Calcola i pesi e le costanti di errore nelle formule di
% Newton-Cotes chiuse con indice n da 1 a NP
clear
syms x
NP = 8;
fidw = fopen('NewtonCotesC.dat','w');
fidk = fopen('NewtonCotesCK.dat','w');
for n=1:NP
    h = 2/n;
   % Calcolo dei pesi
   for i=0:n
      w(i+1) = int(lag(x,i,n),0,n);
```

```
   end
   % Calcolo delle costanti d'errore
   if mod(n,2)==0
       K = int(x*rn(x,n),0,n)/factorial(n+2);
   else
       K = int(rn(x,n),0,n)/factorial(n+1);
   end
   % Conversione da variabili simboliche a variabili
   % in doppia precisione
   wi = h*double(w);
   xi = -1:h:1;
   Ki = double(K);
   disp(wi);
   fprintf(fidw,'%20.16f %20.16f\n',[xi;wi]);
   fprintf(fidw,'\n');
   fprintf(fidk,'%2d %22.16f\n',[n;Ki]);
end
fclose(fidw);
fclose(fidk);
```

Entrambi i programmi fanno uso delle funzioni `lag`, `rn` e `rni`, definite come segue

```
function p = lag(x,i,n)
%   Sintassi p = lag(x,i,n)
%
%   Definisce il polinomio di Lagrange relativo alla
%   distribuzione uniforme di nodi x = (0, 1, ..., n)
%
p=rni(x,i,n)/rni(i,i,n);
```

```
function r = rn(x,n)
%   Sintassi r = rn(x,n)
%
%   Calcola il prodotto fattoriale
%      r_n = x^(n+1) =  x (x-1) (x-2) ... (x-n)
%   del coefficiente di Lagrange L_i(x) relativo  alla
%   distribuzione di nodi 0, 1, ... n
%
r = 1;
for j=0:n
   r = r*(x-j);
end
```

```
function r = rni(x,i,n)
%   Sintassi r = rni(x,i,n)
%
%   Calcola il numeratore del coefficiente di Lagrange
%   L_i(x) relativo alla distribuzione di nodi 0, 1, ... n
%
r = 1;
for j=0:i-1
   r = r*(x-j);
end
for j=i+1:n
```

```
   r = r*(x-j);
end
```

In particolare i programmi `NewtonCotesA.m` e `NewtonCotesC.m` fanno uso del *Symbolic Mathematics toolbox* di MATLAB. L'istruzione `syms` consente di definire una variabile di tipo simbolico. In questo modo è possibile utilizzare funzioni *Maple* di calcolo simbolico come

$$\textit{Risultato} \texttt{ = int(}\textit{Espressione}\texttt{,}a\texttt{,}b\texttt{)},$$

che effettua (se possibile) un integrazione analitica esatta della funzione simbolica definita da *Espressione* nell'intervallo $[a,b]$, e restituisce il risultato nella variabile simbolica *Risultato*. Ad esempio `int('sin(x)',a,b)` restituisce `'cos(a)-cos(b)'`.

Il risultato calcolato, se di tipo numerico, per potere essere utilizzato come tale viene convertito in variabile numerica in doppia precisione tramite l'istruzione `double` del *Symbolic Mathematics toolbox*. Si noti anche l'uso della funzione `factorial` per il calcolo del fattoriale.

Riportiamo in Tabella 5.1 e in Tabella 5.2 i pesi per le formule di Newton-Cotes chiuse e, rispettivamente, aperte.

n	W									D
1	1	1								1
2	1	4	1							3
3	1	3	3	1						4
4	7	32	12	32	7					45
5	19	75	50	50	75	19				144
6	41	216	27	272	27	216	41			420
7	751	3577	1323	2989	2989	1323	3577	751		8640
8	989	5888	-928	10496	-4540	10946	-928	5888	989	14175

Tabella 5.1 Pesi delle prime otto formule di Newton-Cotes chiuse. I pesi w_i relativi ad una formula con $n+1$ punti sono dati da $w_i = W_i/D$, $i = 0,\ldots,n$, dove W_i e D sono interi.

n	W								D
0	2								1
1	1	1							1
2	4	-2	4						3
3	11	1	1	11					4
4	11	-14	26	-14	11				10
5	611	-453	562	562	-453	611			720
6	920	-1908	4392	-4918	4392	-1908	920		945
7	1787	-2083	4967	-1711	-1711	4967	-2803	1787	2240

Tabella 5.2 Pesi delle prime otto formule di Newton-Cotes aperte. I pesi w_i relativi ad una formula con $n+1$ punti sono dati da $w_i = W_i/D$, $i = 0,\ldots,n$, dove W_i e D sono interi.

Come si vede, i pesi di tali formule non sono tutti positivi, quindi le ipotesi del teorema generale di convergenza non sono verificate. È anzi possibile dimostrare il seguente

Teorema 5.3 (Kusmin) *Per qualunque successione di formule di Newton-Cotes si ha*

$$\lim_{n\to\infty} \|I_n\| = \lim_{n\to\infty} \sum_{i=0}^{n} |w_i| = +\infty,$$

e in generale non è garantita la convergenza per funzioni continue.

n	$\mathcal{E}_n$					
	Chiuse			Aperte		
0				$1/3$	h^3	$f^{(2)}(\xi)$
1	$-1/12$	h^3	$f^{(2)}(\xi)$	$3/4$	h^3	$f^{(2)}(\xi)$
2	$-1/90$	h^5	$f^{(4)}(\xi)$	$14/45$	h^5	$f^{(4)}(\xi)$
3	$-3/80$	h^5	$f^{(4)}(\xi)$	$95/144$	h^5	$f^{(4)}(\xi)$
4	$-8/945$	h^7	$f^{(6)}(\xi)$	$41/140$	h^7	$f^{(6)}(\xi)$
5	$-275/12076$	h^7	$f^{(6)}(\xi)$	$5257/8640$	h^7	$f^{(6)}(\xi)$
6	$-9/1400$	h^9	$f^{(8)}(\xi)$	$3956/14175$	h^9	$f^{(8)}(\xi)$

Tabella 5.3 Errori delle prime formule di Newton-Cotes.

Riportiamo nella Tabella 5.3 gli errori relativi alle formule di Newton-Cotes con indice "basso". Per trovare l'espressione di tali errori si utilizzano le proprietà degli errori di interpolazione. Osserviamo che la formula di Newton-Cotes aperta con $n = 0$ è nota come *formula del punto medio*, mentre le formule di Newton-Cotes chiuse con $n = 1$ ed $n = 2$ sono dette *formula dei trapezi* e, rispettivamente, *formula di Simpson* o *Cavalieri-Simpson*[4].
Le formule indicate in Tabella 5.3 per la stima dell'errore sono poco utilizzate nella pratica, sia perché non è detto che la funzione sia sufficientemente regolare da ammettere una derivata di ordine così elevato, sia perché il punto ξ non è noto a priori. Stime empiriche dell'errore sono spesso basate su tecniche di estrapolazione.

Esempio 5.5 [Derivazione di alcune formule elementari]
Consideriamo la deduzione di alcune formule elementari di quadratura: la formula del punto medio, la formula dei trapezi e la formula di Simpson. Iniziamo però con una formula ancora più semplice: la formula dei rettangoli. Considereremo comunque il caso di funzione peso $w(x) = 1$. La formula dei rettangoli consiste nell'approssimare la funzione f con una funzione costante $f(x) \approx f(x_0)$ con $x_0 \in [a,b]$. Supponiamo $f \geq 0$; dal punto di vista grafico stiamo approssimando l'area sottesa dal grafico della funzione f con un rettangolo di base $[a,b]$ e altezza $f(x_0)$,

[4] Thomas Simpson trovò la formula che porta il suo nome nel 1743 anche se in realtà era una riscoperta della medesima formula trovata da Cavalieri nel 1639.

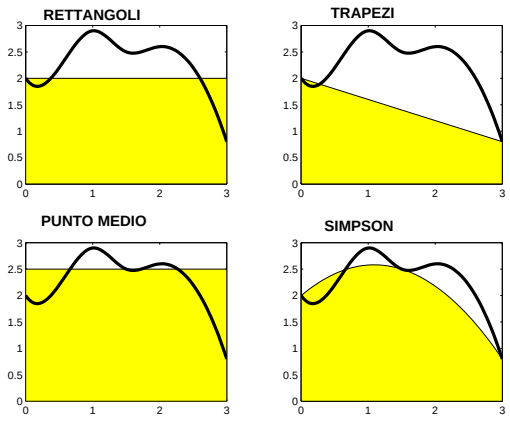

Figura 5.5 Formule di quadratura interpolatorie elementari.

ci confronti con Figura 5.5. Scegliendo, per esempio, $x_0 = a$, la formula si scrive come

$$I_0[f] = (b-a)f(a) = \text{“base per altezza”}.$$

Dall'espressione dell'errore di interpolazione si deduce che (al solito $f[\dots]$ indica una differenza divisa)

$$f(x) = f(a) + f[a,x](x-a),$$

da cui l'errore della formula di quadratura vale

$$\mathcal{E}_0 = \int_a^b f(x)dx - f(a)(b-a) = \int_a^b f(x)dx - \int_a^b f(a)dx,$$

e quindi

$$\mathcal{E}_0 = \int_a^b (f(x) - f(a))dx = \int_a^b f[a,x](x-a)dx.$$

Per il Teorema del valor medio per gli integrali, si veda [22], si ottiene

$$\mathcal{E}_0 = f[a,\tau] \int_a^b (x-a)dx,$$

con $\tau \in (a,b)$ da cui, per la forma delle differenze divise e integrando il polinomio $(x-a)$, si deduce

$$\mathcal{E}_0 = \frac{f'(\xi)(b-a)^2}{2}.$$

In generale per la formula dei rettangoli generica $f(x_0)(b-a)$, si arriva ad una espressione analoga dell'errore, se però si riuscisse a determinare x_0 in modo tale che

$$\int_a^b (x-x_0)dx = 0,$$

è ragionevole aspettarsi un maggior grado della formula di quadratura. Affinché tale integrale sia nullo è indispensabile che la funzione $(x-x_0)$ cambi segno in $[a,b]$. Da un semplice esame grafico, o integrando direttamente il polinomio $(x-x_0)$, si trova che $x_0 = (a+b)/2$ è il punto che fa per noi (l'integrale di $x-(a+b)/2$ tra a e b è nullo). Ricordando le proprietà di ricorrenza delle differenze divise, x, $x_1 \in [a,b]$,

$$f[x_0,x_1,x] = \frac{f[x_0,x] - f[x_0,x_1]}{(x-x_1)},$$

si ottiene

$$f[x_0,x] = f[x_0,x_1] + (x-x_1)f[x_0,x_1,x],$$

da cui

$$\mathcal{E}_0 = \int_a^b \left(f[x_0,x_1] + (x-x_1)f[x_0,x_1,x]\right)(x-x_0)dx,$$

e quindi

$$\mathcal{E}_0 = f[x_0,x_1]\int_a^b (x-x_0)dx + \int_a^b f[x_0,x_1,x](x-x_1)(x-x_0)dx.$$

Quando il primo integrale è nullo si ricava

$$\mathcal{E}_0 = \int_a^b f[x_0,x_1,x](x-x_1)(x-x_0)dx,$$

e se ora $(x-x_1)(x-x_0)$ non cambia segno possiamo ancora utilizzare i Teoremi della media e ottenere

$$\mathcal{E}_0 = \frac{f''(\xi)}{2}\int_a^b (x-x_1)(x-x_0)dx.$$

Una scelta semplice consiste nel prendere $x_1 = x_0 = (a+b)/2$ e quindi $(x-x_0)(x-x_1) = (x-x_0)^2$, da cui formula (del punto medio) ed errore sono dati da

$$I_M[f] = (b-a)f\left(\frac{a+b}{2}\right), \quad \mathcal{E}_M = \frac{f''(\xi)(b-a)^3}{24},\ \xi \in (a,b).$$

Come si vede il costo computazionale della formula del punto medio è uguale a quella dei rettangoli ma con il guadagno di un ordine di accuratezza: la formula dei rettangoli è esatta per funzioni costanti mentre la formula del punto medio per funzioni lineari.

Nella formula dei trapezi si approssima la funzione f con la retta interpolante nei punti $(a,f(a))$, $(b,f(b))$. Utilizzando la forma di Lagrange del polinomio interpolatore, tale retta ha espressione

$$r(x) = f(a)\frac{x-b}{a-b} + f(b)\frac{x-a}{b-a}.$$

La formula I_T è allora determinata integrando tale retta

$$I_T[f] = \int_a^b \left(f(a)\frac{x-b}{a-b} + f(b)\frac{x-a}{b-a}\right)dx,$$

da cui, per linearità

$$I_T[f] = \int_a^b f(a)\frac{x-b}{a-b}\,dx + \int_a^b f(b)\frac{x-a}{b-a}\,dx,$$

e quindi

$$I_T[f] = f(a)\frac{b-a}{2} + f(b)\frac{b-a}{2} = \frac{b-a}{2}(f(a)+f(b)).$$

Alla medesima formula si poteva giungere per via geometrica perché, almeno per $f(x) \geq 0$, l'area sottesa dalla retta, tra $x = a$ e $x = b$, è uguale all'area del trapezio[5] rettangolo di vertici $(a,0)$, $(a,f(a))$, $(b,f(b))$, $(b,0)$ (si veda Figura 5.5). La regola di Simpson si ottiene interpolando la funzione f nei nodi $x_0 = a$, $x_1 = (a+b)/2$, $x_2 = b$ e la sua espressione (dopo calcoli analoghi a quelli fatti per la formula dei trapezi) è la seguente

$$I_S[f] = \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right).$$

In Tabella 5.4 si riportano i risultati numerici relativi alla formula dei rettangoli, del punto medio, dei trapezi, di Simpson per il calcolo approssimato dell'integrale

$$\int_0^1 e^{-x^2}\,dx,$$

che è un esempio di integrale non risolvibile tramite il calcolo della funzione primitiva con funzioni elementari (si riportano comunque solo alcune cifre decimali del risultato). Vista la regolarità della funzione f, il valore più attendibile è quello fornito dalla formula di Simpson. Il valore corretto con 6 cifre significative è infatti 0.746824.

Calcolo di $\int_0^1 e^{-x^2}\,dx$	
rettangoli	1
punto medio	0.778800
trapezi	0.683939
Simpson	0.747180

Tabella 5.4 Confronto fra formule dei rettangoli, del punto medio, dei trapezi e di Simpson.

■

[5] Da cui il nome della formula.

5.3 Formule gaussiane

Abbiamo visto che le formule di quadratura di Newton-Cotes con n pari hanno ordine polinomiale $m = n + 1$. Ci chiediamo quale sia l'ordine massimo ottenibile con una opportuna scelta dei nodi e dei pesi. I gradi di libertà di una formula basata su $n + 1$ nodi sono $2(n + 1)$, mentre le condizioni da imporre affinché una formula sia di ordine m sono $m+1$, pari al numero di coefficienti di un polinomio di grado m. Se uguagliamo il numero di gradi di libertà con il numero di parametri otteniamo che il massimo ordine ottenibile risulta essere $m_{\max} = 2n + 1$. Ci aspettiamo che questo sia dunque l'ordine massimo per una formula di quadratura. Mostriamo dapprima che in effetti una formula di quadratura non può avere ordine superiore. Supponiamo per assurdo che esista una formula di quadratura di ordine $m = 2n + 2$. Consideriamo il polinomio $Q = \omega_{n+1}(x)^2$ di grado $2n + 2$, dove

$$\omega_{n+1}(x) = \prod_{i=0}^{n}(x - x_i). \tag{5.12}$$

Si ha $I_n[Q] = \sum_{i=0}^{n} w_i Q(x_i) = 0$, mentre $I[Q] = \int_a^b w(x)Q(x)\,dx > 0$, pertanto $I[Q] \neq I_n[Q]$ e quindi $m < 2n + 2$.
Adesso, sotto opportune condizioni, vediamo se e come sia possibile ottenere l'ordine massimo $m = 2n + 1$.
Supponiamo che la formula di quadratura interpolatoria abbia i nodi $x_0, x_1, \ldots, x_n$ uguali agli zeri di un polinomio $q_{n+1} \in \mathbb{P}_{n+1}$ tale che

$$I[p_n q_{n+1}] = \int_a^b w(x)p(x)\prod_{i=0}^{n}(x - x_i)\,dx = 0, \quad \forall p \in \mathbb{P}_n. \tag{5.13}$$

Una tale formula è detta formula di *quadratura gaussiana*.

La condizione (5.13) indica il fatto che il polinomio q_{n+1} è ortogonale, rispetto al prodotto scalare indotto dalla funzione peso w, a tutti i polinomi di grado n. Ricordiamo che due funzioni si dicono ortogonali se il loro prodotto scalare è nullo. Il prodotto scalare qui utilizzato è l'integrale del prodotto di due funzioni f e g e del peso w

$$(f,g)_w := \int_a^b w(x)f(x)g(x)\,dx.$$

Dapprima facciamo vedere che una formula gaussiana ha in effetti ordine polinomiale massimo $m = 2n+1$, e poi daremo delle indicazioni su come costruire questa formula.

Data una funzione continua f, la teoria dell'interpolazione polinomiale ci dice che essa si può scrivere come

$$f(x) = p_n(x) + R_n(x),$$

dove $p_n(x)$ indica il polinomio di interpolazione sui nodi $x_0, \ldots, x_n$, ed R_n rappresenta il resto. Utilizzando la rappresentazione dell'errore con le differenze divise si può scrivere il resto come

$$R_n(x) = f[x_0, \ldots, x_n, x]\prod_{i=0}^{n}(x - x_i),$$

dove $f[x_0,\ldots,x_n,x]$ indica una differenza divisa di ordine $n+1$. Se f è un polinomio di grado $2n+1$, la sua differenza divisa di ordine $n+1$ è un polinomio di grado n, pertanto si ha

$$f(x) = p_n(x) + q(x)\prod_{i=0}^{n}(x - x_i), \quad q \in \mathbb{P}_n.$$

Supponiamo ora che la formula di quadratura I_n sia una formula interpolatoria, dove i nodi sono gli zeri del polinomio $q_{n+1} \in \mathbb{P}_{n+1}$ ortogonale a tutti i polinomi $p \in \mathbb{P}_n$, rispetto al prodotto scalare individuato dalla funzione peso $w(x)$.
Sia $f \in \mathbb{P}_{2n+1}$, si ha dunque

$$I[f] - I_n[f] = I[p_n + qq_{n+1}] - I_n[p_n + qq_{n+1}] = I[qq_{n+1}],$$

dove $q_{n+1} = \prod_{i=0}^{n}(x-x_i)$. L'ultima uguaglianza deriva dal fatto che $I[p_n] = I_n[p_n]$ e $I_n[qq_{n+1}] = 0$. Per la proprietà di ortogonalità di q_{n+1}, risulta che $I[qq_{n+1}] = 0 \quad \forall q \in \mathbb{P}_n$, pertanto la formula risulta esatta fino a polinomi di grado $2n+1$.

Mostriamo che una formula di quadratura con massimo ordine ha tutti pesi positivi. Sia I_n una formula di quadratura di ordine $m = 2n+1$. Sia $L_i(x)$ l'i-mo coefficiente di Lagrange costruito sui nodi della formula di quadratura. Esso è un polinomio di grado n, ed il suo quadrato un polinomio di grado $2n$. Si ha pertanto

$$I[L_i^2] = \int_a^b w(x)L_i^2(x)\,dx = I_n[L_i^2] = \sum_{i=0}^{n} w_j L_i^2(x_j) = \sum_{i=0}^{n} w_j L_i(x_j) = w_i.$$

La seconda uguaglianza è dovuta al fatto che la formula di quadratura ha ordine polinomiale $m \geq 2n$ e quindi risulta esatta per il polinomio L_i^2, mentre le ultime uguaglianze sono dovute alla proprietà dei coefficienti di Lagrange, $L_i^2(x_j) = L_i(x_j) = \delta_{ij}$. Dalla catena di uguaglianze segue che $w_i > 0, \forall i = 0,\ldots,n$.

Rimane il problema di come costruire il polinomio q_{n+1}. Un metodo consiste nell'imporre che il polinomio $q_{n+1}(x) = \sum_{k=0}^{n+1} a_j x^{n+1-j}$ sia ortogonale a tutti i polinomi x^k, con $k = 0,\ldots,n$. Si ottiene un sistema di $n+1$ equazioni lineari nelle $n+1$ incognite $a_1,\ldots,a_{n+1}$ ($a_0 = 1$ poiché il polinomio q_{n+1} è monico).

Il sistema si può scrivere nella forma

$$\sum_{i=1}^{n+1} a_i(x^{n+1-i},x^k)_\omega = -(x^{n+1},x^k)_\omega, \quad k = 0,\ldots,n.$$

Sfruttando la simmetria del prodotto scalare si ottiene

$$\sum_{i=1}^{n+1} a_i p_{n+k+1-i} = p_{n+k+1}, \quad k = 0,\ldots,n,$$

dove

$$p_k = (x^k,1)_\omega = \int_a^b w(x)x^k\,dx.$$

Per calcolare i coefficienti del polinomio è sufficiente quindi calcolare $2n+2$ prodotti scalari, e risolvere un sistema lineare $(n+1)\times(n+1)$.

Una volta determinato il polinomio q_{n+1}, i nodi sono determinati calcolando gli zeri dello stesso polinomio mentre i pesi sono dati dalla espressione (5.11). La funzione MATLAB `Gauss.m` calcola nodi e pesi di formule gaussiane, utilizzando il metodo sopra descritto

```
function [xg,wg] = Gauss(w,a,b,n)
% Sintassi [xg,wg] = Gauss('w(x)',a,b,n)
%
% Calcola nodi e pesi di una formula di Gauss con n nodi e funzione
% peso individuata dalla stringa 'w(x)'
%
% Esempi
%
% [xg,wg] = Gauss('1',-1,1,5)
% Calcola nodi e pesi delle formule di Gauss-Legendre con 5 punti
%
% [xg,wg] = Gauss('exp(-x)',0,inf,7)
% Calcola nodi e pesi della formula di Gauss-Laguerre con 7 punti
%
% [xg,wg] = Gauss('exp(-x^2)',-inf,inf,4)
% Calcola nodi e pesi della formula di Gauss-Hermite con 4 punti
%
% [xg,wg] = Gauss('1/sqrt(1-x^2)',-1,1,6)
% Calcola nodi e pesi della formula di Gauss-Chebicev con 6 punti
%
syms x li;
% Calcolo dei prodotti scalari p_k = (x^k,1)
for k=0:2*n
   eval(['p(k+1) = int(' w '*x^k,a,b);']);
end
% Calcolo della matrice dei coefficienti
% Il calcolo viene mantenuto simbolico fino alla soluzione del
% sistema lineare, in modo da lavorare in aritmetica esatta
for i=0:n-1
    for j=1:n
        S(i+1,j) = p(n+i-j+1);
    end
end
for k=0:n-1
    beta(k+1) = -p(n+k+1);
end
% Calcolo dei coefficienti del polinomio ortogonale di grado n
alpha = double([1;S\beta']);
% Calcolo degli zeri del polinomio
xg = sort(roots(alpha));
% Calcolo dei pesi della formula du quadratura
for i=1:n
    if i==1
       li = prod((x-xg(2:n))./(xg(i)-xg(2:n)));
    else
       if i==n
          li = prod((x-xg(1:n-1))./(xg(i)-xg(1:n-1)));
       else
          li = prod((x-xg(1:i-1))./(xg(i)-xg(1:i-1))).*prod( ...
```

```
          (x - xg(i+1:n))./(xg(i)-xg(i+1:n)));
       end;
    end;
   eval(['wg(i) = int(' w '*li,a,b);']);
end
wg = double(wg)';
```

Occorre specificare la funzione peso $w(x)$ mediante una stringa di caratteri che la definisca, gli estremi di integrazione e il numero di nodi desiderati. Si noti l'uso della funzione della funzione MATLAB standard `sort` che se applicata ad un vettore ne riordina gli elementi in ordine crescente.

5.3.1 Polinomi ortogonali

Una tecnica più efficace ed elegante per il calcolo delle formule di tipo gaussiano si basa sulla teoria dei polinomi ortogonali. Senza voler entrare in dettaglio in questo settore, diamo qui un breve cenno di come si possano costruire i polinomi ortogonali. Una famiglia di polinomi ortogonali P_n è caratterizzata dalla proprietà che ciascun polinomio P_n è ortogonale a tutti i polinomi di grado minore o uguale ad n. In particolare si ha

$$(P_n,P_j) = 0, \quad j = 0,\ldots,n-1.$$

Supponiamo ora di conoscere gli $n+1$ polinomi ortogonali di grado minore o uguale ad n, e vogliamo determinare il polinomio P_{n+1}. Poiché i polinomi $\{P_0,\ldots,P_n\}$ formano una base di $\mathbb{P}_n$, possiamo scrivere il nuovo polinomio come combinazione lineare

$$P_{n+1}(x) = \sum_{i=0}^{n} C_i^n P_i(x) + \gamma_n x P_n(x). \tag{5.14}$$

Effettuando il prodotto scalare con P_k, $k = 0,\ldots,n$, si ottiene

$$0 = (P_{n+1},P_k) = C_k^n (P_k,P_k) + \gamma_n (xP_n,P_k), \quad k = 0,\ldots,n$$

La prima uguaglianza è dovuta all'imporre che il nuovo polinomio P_{n+1} sia ortogonale ai precedenti, mentre, per l'ortogonalità dei polinomi, solo un termine della sommatoria è diverso da zero. A sua volta, per la simmetria del prodotto scalare, l'ultimo termine può essere scritto nella forma

$$(xP_n,P_k) = (P_n,xP_k)$$

e si annulla per $k+1 < n$, per l'ipotesi di ortogonalità di P_n. Da quanto scritto sopra segue che solo i coefficienti C_{n-1}^n e C_n^n sono diversi da zero. L'equazione (5.14) pertanto può essere scritta nella forma

$$P_{n+1} = \alpha_n P_n + \beta_n P_{n-1} + \gamma_n x P_n \tag{5.15}$$

Effettuando il prodotto scalare per P_n e per P_{n-1} si ottengono le relazioni di ricorrenza

$$\alpha_n = -\frac{\gamma_n (xP_n,P_n)}{(P_n,P_n)}, \tag{5.16}$$

$$\beta_n = -\frac{\gamma_n (P_n,xP_{n-1})}{(P_{n-1},P_{n-1})} = -\frac{\gamma_n (P_n,P_n)}{\gamma_{n-1}(P_{n-1},P_{n-1})}. \tag{5.17}$$

La dimostrazione dell'ultima uguaglianza è lasciata al lettore.

Osservazione 5.1 I prodotti scalari che abbiamo considerato soddisfano la proprietà $(xf,g) = (f,xg)$, tuttavia il concetto di prodotto scalare è più generale di quello considerato in questo capitolo. Prodotti scalari di tipo diverso potrebbero non soddisfare tale relazione. ■

Le costanti γ_n sono arbitrarie e possono essere scelte o per soddisfare proprietà di normalizzazione dei polinomi, oppure per semplificare le relazioni di ricorrenza.

Alcune funzioni peso ricorrono di frequente nelle applicazioni. Per esse una espressione esplicita dei polinomi ortogonali è stata trovata, e i nodi e pesi per tali polinomi sono stati tabulati. Riportiamo qui alcune delle formule gaussiane di uso più comune.

Formule di Gauss-Legendre Si ottengono per $[a,b] = [-1,1]$, e $w(x) \equiv 1$. I polinomi ortogonali sono i polinomi di Legendre, definiti dalla relazione di ricorrenza

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x), \quad P_0 = 1, P_1 = x$$

I nodi $x_i, i = 1, \ldots, n$ di una formula di quadratura I_{n-1} sono gli zeri del polinomio di Legendre P_n. I pesi sono dati dalla relazione

$$w_i = \frac{2(1-x_i^2)}{[(n+1)P_{n+1}(x_i)]^2}.$$

Formule di Gauss-Chebicev Si ottengono per $[a,b] = [-1,1]$, e $w(x) \equiv 1/\sqrt{1-x^2}$. I polinomi ortogonali sono i polinomi di Chebicev, definiti dalla relazione di ricorrenza

$$T_{n+1}(x) = 2xT_n(x) - P_{n-1}(x), \quad T_0 = 1, T_1 = x$$

I nodi $x_i, i = 1, \ldots, n$ sono gli zeri del polinomio di Chebicev T_n, e sono dati dalla formula esplicita

$$x_i = \cos\left(\frac{(2i-1)\pi}{2n}\right), \quad i = 1, \ldots, n.$$

I pesi sono costanti pari a

$$w_i = \frac{\pi}{n}, \quad i = 1, \ldots, n.$$

Formule di Gauss-Laguerre Si ottengono per $[a,b] = [0, +\infty]$, e $w(x) \equiv \exp(-x)$. I polinomi ortogonali sono i polinomi di Laguerre, definiti dalla relazione di ricorrenza

$$(n+1)L_{n+1}(x) = (1+2n-x)L_n(x) - nL_{n-1}(x), \quad L_0 = 1, L_1 = 1-x.$$

I nodi $x_i, i = 1, \ldots, n$ di una formula di quadratura I_{n-1} sono gli zeri del polinomio di Laguerre L_n. I pesi sono dati dalla relazione

$$w_i = \frac{(n!)^2 x_i}{[L_{n+1}(x_i)]^2}, \quad i = 1, \ldots, n.$$

Formule di Gauss-Hermite Si ottengono per $[a,b] = [-\infty,+\infty]$, e $w(x) \equiv \exp(-x^2)$. I polinomi ortogonali sono i polinomi H_n di Hermite, definiti dalla relazione di ricorrenza

$$H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x), \quad H_0 = 1, H_1 = 2x.$$

I nodi $x_i, i = 1, \ldots, n$ di una formula di quadratura I_{n-1} sono gli zeri del polinomio di Hermite H_n. I pesi sono dati dalla relazione

$$w_i = \frac{2^{n+1} n! \sqrt{\pi}}{[H_{n+1}(x_i)]^2}.$$

Formule di Radau Nelle formule di Gauss-Legendre i nodi non coincidono con gli estremi dell'intervallo di integrazione. Se si impone che uno o due nodi stiano sul bordo dell'intervallo di integrazione si ottengono altre formule di quadratura. In particolare, se uno solo degli estremi è un nodo, la formula di ordine massimo così definita è detta formula di Radau, mentre le formule di ordine massimo che hanno come nodi entrambe gli estremi sono dette formule di Lobatto.

Se si fissa uno degli estremi, l'ordine massimo ottenibile con una formula con $n+1$ nodi è $m = 2n$, mentre se si fissano due nodi, l'ordine massimo ottenibile è $m = 2n - 1$.

Dalla teoria del polinomi ortogonali segue che i nodi delle formule di Radau I_{n-1} sono il punto -1 e gli zeri del polinomio

$$\varphi_{n-1} = P_{n-1}(x) + \frac{x-1}{n} P'_{n-1}(x) = \frac{P_{n-1}(x) + P_n(x)}{1+x}.$$

I pesi sono dati da

$$w_i = \frac{1}{n^2} \frac{1 - x_i}{[P_{n-1}(x_i)]^2}, \quad i = 1, \ldots, n-1,$$

mentre il peso relativo al nodo -1 vale

$$w_0 = \frac{2}{n^2}.$$

Formule di Lobatto Sono le formule di ordine massimo in cui entrambe gli estremi sono nodi. Gli altri $n-2$ nodi x_i, $i = 2, \ldots, n-1$ sono zeri del polinomio

$$\varphi(x) = P'_{n-1}(x).$$

I pesi sono dati da

$$w_i = \frac{2}{n(n-1)[P_{n-1}(x_i)]^2}, \quad i = 2, \ldots, n-1,$$

mentre i pesi per i nodi in ± 1 sono dati da

$$w_1 = w_n = \frac{2}{n(n-1)}.$$

Come illustrazione delle famiglie di polinomi ortogonali riportiamo in Figura 5.6 alcuni grafici di tali polinomi. I codici MATLAB che permettono di calcolare

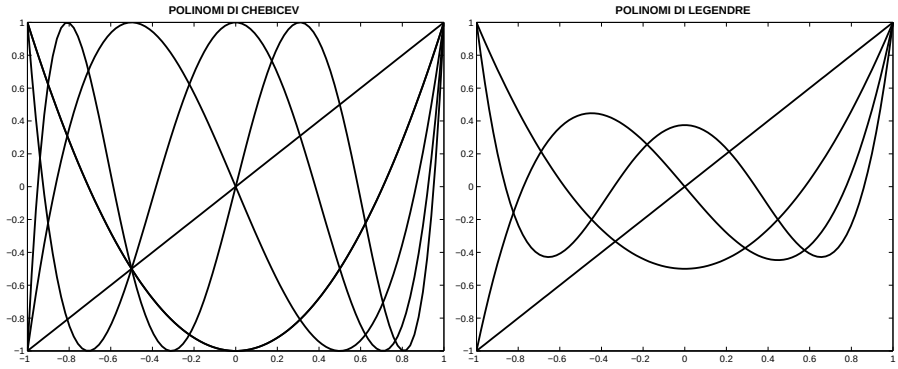

Figura 5.6 Alcuni polinomi ortogonali dalla famiglia di Chebicev e di Legendre.

nodi e pesi delle formule di tipo gaussiano sono riassunti in Tabella 5.5 e sono disponibili al sito Web del libro.

Riportiamo per brevità solo lo script nel caso delle formule di Lobatto

```
% Lobatto.m
%
% Il programma calcola nodi e pesi delle formule di quadratura
% di Lobatto, e li scrive nel file Lobatto.dat
clear;
syms x p phi;
N = 16;     % Indica il massimo numero di nodi
% Vengono definiti i primi due polinomi di Legendre
p(1) = 1;
p(2) = x;
% Vengono definiti ricorsivamente i successivi polinomi
for n=1:N-1
   p(n+2) = expand(((2*n+1)*x*p(n+1)-n*p(n))/(n+1));
end
fid = fopen('Lobatto.dat','w');
% Vengono trovare le radici xi dei polinomi ed i pesi w
for n=2:N-2
   % calcolo della derivata
   phi = diff(p(n));
   % Conversione da simbolico a numerico
   p1 = sym2poly(phi);
   p2 = sym2poly(p(n));
   % Calcolo dei nodi interni
   xi = sort(roots(p1));
   % Calcolo dei pesi interni
   wi = 1./polyval(p2,xi).^2;
   xn = [-1;xi;1];
   wn = 2*[1;wi;1]/(n*(n-1));
   disp([xn,wn]);
   disp(sum(wn)); % In aritmetica esatta dovrebbe essere 2
   fprintf(fid,'%2.16f %2.16f\n',[xn';wn']);
```

```
   fprintf(fid,'\n')'
end
fclose(fid);
```

File MATLAB	Formula di quadratura	File
`Gauss.m`	Formule gaussiane con assegnata funzione peso	-
`GaCe.m`	Formule di Gauss-Chebicev	GCe.dat
`GaHe.m`	Formule di Gauss-Hermite	GHe.dat
`GaLa.m`	Formule di Gauss-Laguerre	GLa.dat
`GaLe.m`	Formule di Gauss-Legendre	GLe.dat
`Lobatto.m`	Formule di Lobatto	Lobatto.dat
`Radau.m`	Formule di Radau	Radau.dat

Tabella 5.5 Elenco degli script MATLAB per il calcolo di nodi e pesi delle formule di quadratura di Gauss più comuni.

Nel precedente script abbiamo utilizzato le funzioni del *Symbolic Mathematics toolbox* `expand` che scrive un'espressione simbolica in forma estesa, ad esempio `expand('(x+1)^2')` fornisce `'x^2+2*x+1'`, e la funzione `sym2poly` che converte un polinomio scritto in forma simbolica in un vettore di coefficienti adatto ad essere utilizzato con la funzione `roots`. Infine la funzione `diff` esegue la derivata di una funzione in modo simbolico, ad esempio `diff('sin(x^2)')` restituisce `'2*cos(x^2)*x'`, oppure numerico tramite una approssimazione alle differenze finite se l'argomento è un vettore. Si consulti l'help di MATLAB per maggiori informazioni.

Osservazione 5.2 Osserviamo che la formula del punto medio, che può essere vista come formula di Newton-Cotes aperta con $n = 0$, è anche la formula di Gauss-Legendre con $n = 0$. La formula dei Trapezi è la formula di Lobatto con $n = 1$, e la formula di Simpson è la formula di Lobatto con $n = 2$. ■

Osservazione 5.3 Nodi e pesi delle formule di quadratura sono stati ricavati per intervalli ben determinati. Per calcolare integrali in intervalli arbitrari è sufficiente effettuare un cambiamento di variabili che riporti l'intervallo di integrazione nell'intervallo standard. Ad esempio, per calcolare

$$\int_a^b f(t)\,dt \tag{5.18}$$

mediante una formula di quadratura con nodi x_i e pesi w_i ottenuti per approssimare un integrale nell'intervallo [−1,1], cerchiamo una trasformazione $t = \alpha x + \beta$, e imponiamo $a = -\alpha + \beta$ e $b = \alpha + \beta$. Si ottiene

$$\alpha = \frac{b-a}{2}, \quad \beta = \frac{b+a}{2}.$$

Applicando la trasformazione all'integrale (5.18) si ha

$$\int_a^b f(t)\,dt = \alpha \int_{-1}^{1} f(\alpha x + \beta)\,dx \approx \alpha \sum_{i=0}^{n} w_i f(\alpha x_i + \beta) = \sum_{i=0}^{n} \tilde{w}_i f(t_i),$$

pertanto i nodi e i pesi della formula di quadratura nell'intervallo $[a,b]$ sono dati da

$$t_i = \frac{b-a}{2}x_i + \frac{b+a}{2}, \quad \tilde{w}_i = \frac{b-a}{2}w_i.$$

Analoghe trasformazioni permettono di applicare, ad esempio, le formule di Gauss-Laguerre al calcolo di integrali del tipo

$$\int_a^\infty \exp(-\lambda t) f(t)\, dt,$$

o le formule di Gauss-Hermite ad integrali del tipo

$$\int_{-\infty}^\infty \exp(-(t-t_0)^2/\sigma^2) f(t)\, dt.$$

■

5.4 Formule composite

Come abbiamo visto nel capitolo riguardante l'interpolazione, per migliorare la approssimazione uniforme di una funzione raramente si utilizzano polinomi di grado molto elevato. Piuttosto si preferisce suddividere l'intervallo in un certo numero di sottointervalli, ed utilizzare una approssimazione polinomiale a tratti. Una simile strategia si adopera anche per l'approssimazione di integrali.

A questo scopo consideriamo l'integrale

$$I[f] = \int_a^b f(x)\, dx.$$

Suddividiamo l'intervallo $[a,b]$ in un certo numero N di intervalli mediante i punti $a = x_0 < x_1 < \ldots < x_{N-1} < x_N = b$. Chiamiamo Δ tale suddivisione. Riscriviamo l'integrale come somma di integrali nei vari intervalli.

$$I[f] = \sum_{i=1}^N I^{(i)}[f] = \sum_{i=1}^N \int_{x_{i-1}}^{x_i} f(x)\, dx,$$

e per ciascuno di questi utilizziamo una delle formule I_n viste nella sezione precedente. Si ottiene così una *formula di quadratura composita*. Scegliendo, ad esempio, la formula dei trapezi per ciascuno degli intervalli, si ottiene la formula dei trapezi composita

$$T_\Delta[f] = \sum_{i=1}^N T^{(i)}[f] = \sum_{i=1}^N \frac{1}{2}(f(x_{i-1}) + f(x_i))(x_i - x_{i-1}). \tag{5.19}$$

Tale formula si semplifica nel caso di divisione uniforme dell'intervallo

$$T_h[f] = h\left(\frac{f(a)+f(b)}{2} + \sum_{i=1}^{N-1} f(a+ih)\right), \tag{5.20}$$

dove $h = (b-a)/N$.

La formula di Simpson composita diventa

$$S_\Delta[f] = \sum_{i=1}^{N} S^{(i)}[f] = \sum_{i=1}^{N} \frac{1}{6}(f(x_{i-1}) + 4f(\frac{x_{i-1}+x_i}{2}) + f(x_i))(x_i - x_{i-1}). \quad (5.21)$$

Nel caso di suddivisione uniforme dell'intervallo si ha

$$S_h[f] = \frac{h}{6}\left(f(a) + f(b) + 2\sum_{i=1}^{N-1} f(a+ih) + 4\sum_{i=1}^{N} f(a+i\frac{h}{2})\right). \quad (5.22)$$

Stime dell'errore per tali formule composite si deducono immediatamente dall'espressione delle stime dell'errore per le formule semplici.

$$I[f] = T_h[f] + R_T(h),$$

$$I[f] = S_h[f] + R_S(h),$$

dove

$$R_T(h) = -\frac{(b-a)}{12}h^2 f''(\xi), \quad R_S(h) = -\frac{(b-a)}{2880}h^4 f^{(4)}(\xi)h^4.$$

Si vede quindi che la formula dei trapezi composita ha una accuratezza del secondo ordine in h, mentre la formula di Simpson è del quarto ordine.

Osservazione 5.4 Si noti che nella formula dei trapezi composita nel caso di suddivisione uniforme (5.20) se consideriamo una sequenza di passi del tipo $h_j = (b-a)/2^j$, $j = 0,1,\ldots$, la valutazione della sommatoria non richiede che si ricalcolino tutti i valori $f(x_i)$. Indichiamo con $T(f,j)$ le formule dei trapezi composite per tale sequenza di passi. In Figura 5.7 si mostra come esempio il caso dei punti coinvolti nella formula $T(f,3)$. Alcuni di questi punti sono stati utilizzati nella formula $T(f,2)$: x_0, x_2, x_4, x_6, x_8. Infatti $h_3 = (b-a)/2^3 = h_2/2$ e quindi i punti di indice pari sono i vecchi punti della formula con passo doppio rispetto al passo attuale. In generale, posto $2^j = 2m$ si ottiene la relazione ricorrente

$$T(f,j) = \frac{T(f,j-1)}{2} + h_j \sum_{k=1}^{m} f(x_{2k-1}), \quad T(f,0) = \frac{(b-a)}{2}(f(a)+f(b)).$$

Il denominatore 2 è dovuto al fattore di variazione del passo. Le formule gaussiane non permettono questa gerarchia di griglie perch'e le radici di successivi polinomi ortogonali sono sempre diverse[6]. ■

[6] Un compromesso interessante è rappresentato dalle formule composite di *Gauss-Kronrod* costituite da una formula di tipo gaussiano G_n con n punti e una formula di Kronrod K_{2n+1} con $(2n+1)$ punti scelta in modo tale che tutti i punti della formula gaussiana sono utilizzati. I restanti nodi sono scelti in modo tale da massimizzare l'ordine della formula totale risultante. In molte librerie matematiche la coppia utilizzata maggiormente è la coppia (G_7, K_{15}).

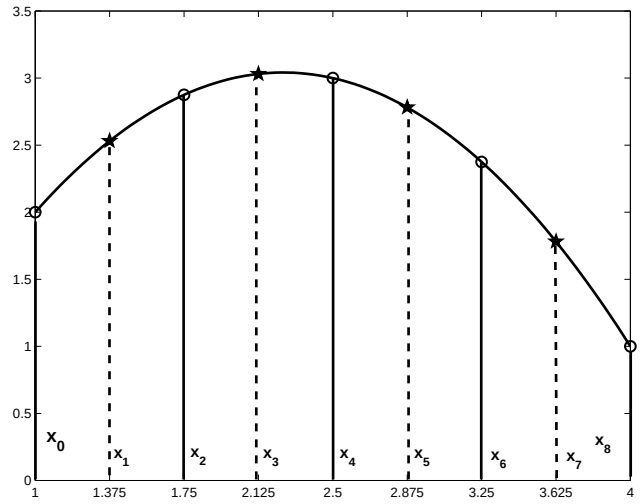

Figura 5.7 Formula composita dei trapezi, —o vecchi punti, $--\star$, nuovi punti.

5.4.1 Procedimento di estrapolazione di Richardson

La conoscenza dell'ordine di accuratezza di una formula di quadratura ci permette di costruire una nuova formula di quadratura di ordine più elevato. Supponiamo, ad esempio, che una formula di quadratura composita, che denotiamo genericamente con $F(h)$, abbia una accuratezza di ordine p, cioè che si possa scrivere

$$I[f] = F(h) + Ch^p + o(h^p), \tag{5.23}$$

dove con $o(h^p)$ denotiamo un infinitesimo in h di ordine superiore a p. Usando un passo qh si ottiene

$$I[f] = F(qh) + C(qh)^p + o(h^p). \tag{5.24}$$

Moltiplicando la prima espressione per q^p, sottraendone la seconda e risolvendo per $I[f]$ si ottiene

$$I[f] = F_1(h) + o(h^p), \quad \text{con} \quad F_1(h) \equiv \frac{q^p F(h) - F(qh)}{q^p - 1}. \tag{5.25}$$

La formula così ottenuta è di ordine più elevato della formula di partenza. Il procedimento può essere iterato e applicato alla formula F_1, a patto di conoscere l'ordine di infinitesimo del resto. Supponiamo che la formula di partenza $F(h)$ ammetta uno sviluppo dell'errore del tipo

$$I[f] = F(h) + c_1 h^{p_1} + c_2 h^{p_2} + \ldots + c_k h^{p_k} + \ldots \tag{5.26}$$

con $p_1 < p_2 < \cdots < p_k < \cdots$, si può allora costruire una successione di formule di quadratura, iterando il procedimento

$$F_{k+1}(h) = \frac{q^{p_k} F_k(h) - F_k(qh)}{q^{p_k} - 1}, \tag{5.27}$$

con un errore $O(h^{p_{k+1}})$. Il procedimento descritto è noto come procedimento di *estrapolazione di Richardson* e trova applicazione non solo per le formule di quadratura ma anche in ambiti differenti. In pratica per ottenere una $F_k(h)$ si ricorre ad una tabella triangolare simile a quella utilizzata nella formula di Newton delle differenze divise.

Integrazione di Romberg Il metodo di integrazione di Romberg consiste nell'applicazione della formula di estrapolazione di Richardson alla formula dei trapezi composita. Esso si basa su una notevole rappresentazione dell'errore della formula dei trapezi nota come formula di Eulero-Mc Laurin. Secondo tale formula, per una funzione $f \in C^\infty([a,b])$, cioè che ha derivate continue di qualsiasi ordine, vale il seguente sviluppo asintotico

$$\int_a^b f(x)\,dx = \frac{h}{2}\left(f(a)+f(b)+2\sum_{i=1}^{N-1} f(a+ih)\right) + c_1 h^2(f'(b)-f'(a)) \\ +c_2 h^4(f'''(b)-f'''(a)) + \cdots + c_k h^{2k}(f^{2k-1}(b)-f^{2k-1}(a)) + O(h^{2k+2}). \tag{5.28}$$

Le costanti c_k possono calcolarsi esplicitamente. Le prime valgono

$$c_1 = -\frac{1}{12}, \quad c_2 = \frac{1}{720}, \quad c_3 = -\frac{1}{30240}.$$

Lo sviluppo dell'errore è del tipo richiesto per l'estrapolazione di Richardson. Partendo dall'Osservazione 5.4 e con riferimento alle notazioni di (5.27) si scelgono $p_k = 2k$ e $q = 2$. Di seguito mostriamo una semplice implementazione ricorsiva in MATLAB del metodo di Romberg. La funzione fa uso della funzione `Trapz` che implementa la formula dei trapezi.

```
function R = Romb(fname,a,b,L,N)
% Sintassi R = Romb(fname,a,b,L,N)
%
%  Calcola l'integrale mediante la formula di quadratura di Romberg
%  L'intervallo viene diviso in N intervalli, ed in ciascuno
%  viene effettuato il calcolo mediante la formula di Romberg a L
%  livelli. La funzione Romb viene definita ricorsivamente
if (L==0)
 R = Trapz(fname,a,b,N);
else
 R = (2^(2*L)*Romb(fname,a,b,L-1,2*N)-...
     Romb(fname,a,b,L-1,N))/(2^(2*L)-1);
end

function T=Trapz(fname,a,b,N);
% Sintassi T = Trapz(fname,a,b,N)
%
%   Calcola l'integrale della funzione definita dal file funz.m
%   con il metodo dei trapezi nell'intervallo [a,b],
%   suddividendo l'intervallo in N intervalli uguali.
x = linspace(a,b,N+1);
F=feval(fname,x);
T=(sum(F(2:N))+0.5*(F(1)+F(N+1)))*(b-a)/N;
```

Si noti che tale implementazione richiede la possibilità di far uso di chiamate ricorsive di funzioni [7].

5.4.2 Quadratura adattiva

Negli esempi visti finora, l'utente deve scegliere il numero di punti in cui suddividere l'intervallo per il calcolo dell'integrale. Sarebbe tuttavia molto utile una routine che fornisca il calcolo di un integrale con una precisione assegnata in maniera automatica, cosicchè l'utente non debba effettuare controlli *a posteriori*.

Le routine di quadratura automatica utilizzano formule composite con ampiezza del passo di integrazione variabile, che si adatta all'andamento locale della funzione integranda. La scelta dei passi di integrazione viene fatta seguendo l'obiettivo di distribuire l'errore in maniera uniforme su tutto l'intervallo di integrazione.

Supponiamo di voler approssimare l'integrale

$$I[f] = \int_a^b f(x)\,dx,$$

mediante una formula composita del tipo

$$F(\Delta_N; f) = \sum_{i=1}^{N} I_n^{(i)}[f],$$

dove Δ_N indica una decomposizione dell'intervallo $[a,b]$ e di imporre che l'errore commesso dalla formula di quadratura sia inferiore ad una prefissata tolleranza ε. Questo obiettivo si ottiene imponendo che in ciascun intervallo della decomposizione Δ_N dell'intervallo $[a,b]$ l'errore sia inferiore ad una frazione di ε proporzionale alla lunghezza dell'intervallo stesso, cioè che valga

$$\left| \int_{x_{i-1}}^{x_i} f(x)\,dx - I_n^{(i)}[f] \right| < \frac{\varepsilon}{b-a}(x_i - x_{i-1}). \tag{5.29}$$

Allora l'errore totale soddisfa la disuguaglianza

$$\begin{aligned} \left| \int_a^b f(x)\,dx - F(\Delta_N; f) \right| &\leq \sum_{i=1}^{N} \left| \int_{x_{i-1}}^{x_i} f(x)\,dx - I_n^{(i)}[f] \right| \\ &\leq \sum_{i=1}^{N} \frac{\varepsilon}{b-a}(x_i - x_{i-1}) = \varepsilon. \end{aligned} \tag{5.30}$$

L'obiettivo (5.29) si ottiene come segue. Si parte da una suddivisione iniziale uniforme Δ_{N_0} dell'intervallo $[a,b]$, e si stima l'errore in ciascun sottointervallo. Sulla base di tale stima si effettua il test (5.29) su ciascun intervallo. Gli intervalli

[7] La maggior parte dei linguaggi di programmazione utilizzati nel calcolo scientifico (MATLAB, C, Pascal, Fortran90) consente di effettuare chiamate di questo tipo. Linguaggi come il Fortran77 non consentono invece chiamate ricorsive. In tal caso la routine di quadratura adattiva deve essere implementata utilizzando un procedimento diverso (di tipo iterativo)

per i quali tale test non risulta verificato vengono ulteriormente suddivisi, e su di essi viene applicato nuovamente il test. Si procede così ricorsivamente fino a quando il test (5.29) risulta soddisfatto in tutti gli intervalli.

L'errore locale può essere stimato confrontando il risultato con quello ottenuto mediante una formula di quadratura di ordine più elevato, oppure utilizzando il procedimento di estrapolazione di Richardson.

L'affidabilità dei risultati ottenuti dipenderà dalla bontà della stima utilizzata. Questa a sua volta sarà tanto migliore quanto più piccoli sono gli intervalli. Per questo motivo non conviene partire da una suddivisione iniziale Δ_{N_0} troppo grossolana.

Una semplice funzione ricorsiva che implementa la precedente idea utilizzando il metodo dei trapezi è la seguente

```
function [di,ct] = quadtra(fname,a,b,epsilon,fa,fb)
% Sintassi Q = quadtra(fname,a,b,epsilon)
%          [Q,Nf] = quadtra(fname,a,b,epsilon)
%
%   Calcola in maniera ricorsiva l'integrale tra a e b
%   della funzione definita dalla stringa fname con
%   precisione assoluta inferiore ad epsilon*(b-a)
%   La quantit Nf rappresenta il numero di valutazioni di funzione
%
global ct;

c = (a+b)/2;
if (nargin<5)
    fa=feval(fname,a);
    fb=feval(fname,b);
    ct=2;
end
fc=feval(fname,c);
ct=ct+1;
err = (2*fc-fa-fb)/3;
if abs(err)<epsilon
   di = (fa+4*fc+fb)*(b-a)/6;
else
   di = quadtra(fname,a,c,epsilon,fa,fc) + ...
        quadtra(fname,c,b,epsilon,fc,fb);
end
```

La variabile globale `ct` consente di calcolare il numero totale di valutazioni della funzione utilizzate.

L'esempio che segue confronta il risultato ottenuto con la precedente funzione con il risultato ottenuto utilizzando le funzioni di libreria MATLAB `quad` (o `quad8`)[8]. Entrambe realizzano una quadratura di tipo adattivo. La prima è basata sul metodo di Simpson e la seconda sulla formula chiusa di Newton-Cotes per $n = 8$. La sintassi è analoga, ad esempio per la funzione `quad`

[*Risultato*, *Valutazioni*]= `quad(`*Funzione*, *a*, *b*, *Tolleranza*, *Traccia*`)`,

[8] In MATLAB versione 6.0 e seguenti è disponibile anche la funzione *quadl*.

dove *Traccia* è un parametro opzionale che se posto diverso da zero visualizza i risultati intermedi del calcolo, quali la posizione dei punti di suddivisione, durante il procedimento ricorsivo.

Esempio 5.6 [Confronto tra diverse formule di quadratura adattiva] Consideriamo il problema del calcolo di

$$\int_a^b \cos(10x)\exp(\sin(x))\,dx,$$

con $a = 1$ e $b = 10$.

Definendo la funzione MATLAB

```
function f = funz(x)
% sintassi f=funz(x)
% esempio di funzione rapidamente oscillante
%
f = cos(10*x).*exp(sin(x));
```

all'interno dell'ambiente principale di MATLAB possiamo scrivere

```
>> a = 1; b = 10;
>> epsilon = 1e-4;
>> [I1,n] = quadtra('funz',a,b,epsilon)
I1 =
    0.1046
n =
    6389
```

Utilizzando le funzioni di libreria MATLAB abbiamo invece

```
>> [I2,n] = quad('funz',a,b,epsilon)
Recursion level limit reached in quad.  Singularity likely.
I2 =
    0.1046
n=
    2041
>> [I2,n] = quad8('funz',a,b,epsilon)
I2 =
    0.1046
n =
    513
```

Le routines forniscono lo stesso risultato con 4 cifre significative. La maggiore efficienza delle funzioni MATLAB è evidente dal minor numero di valutazioni della funzione. In Figura 5.8 sono riportate le valutazioni della funzione effettuate dalle funzioni `quad` e `quad8`. Si noti inoltre come la funzione `quad` segnala la presenza di un numero eccessivo di chiamate ricorsive della funzione.

■

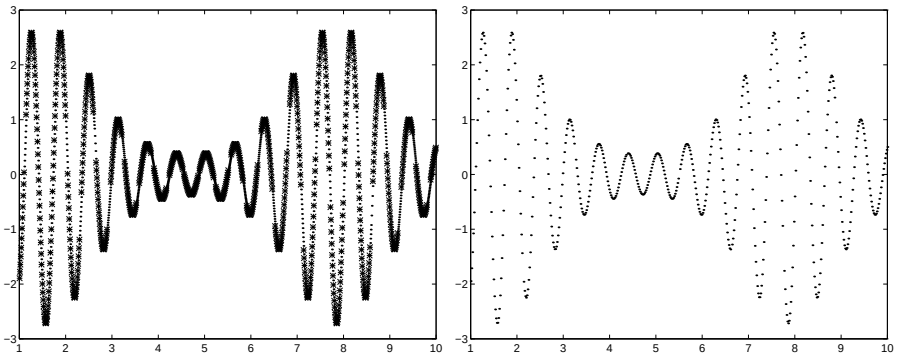

Figura 5.8 Valutazioni della funzione e distribuzione dei punti nell'intervallo ottenuta con le funzioni `quad` (sinistra) e `quad8` (destra) nell'Esempio 5.6.

5.5 Funzioni rapidamente oscillanti

In questa sezione consideriamo il calcolo di integrali di funzioni rapidamente oscillanti. Supponiamo di voler calcolare integrali del tipo

$$\mathrm{IS}_k[f] = \int_a^b f(x) \sin kx \, dx, \quad \mathrm{IC}_k[f] = \int_a^b f(x) \cos kx \, dx. \tag{5.31}$$

Integrali di questo tipo intervengono di frequente qualora si vogliano calcolare i coefficienti di Fourier della funzione f.

Il calcolo di tali integrali diventa particolarmente oneroso per elevati valori di k. Infatti, se si utilizza un'usuale formula di quadratura composita, occorre suddividere l'intervallo di integrazione in un numero di sottointervalli molto superiore a k, per poter risolvere accuratamente la piccola scala dell'oscillazione (si vedano a riguardo i risultati ottenuti nell'Esempio 5.6).

La tecnica che si utilizza per la costruzione di formule di quadratura accurate per tali funzioni è quella di imporre che le formule siano esatte per funzioni f particolarmente semplici. Imponendo, ad esempio, che le formule di quadratura per il calcolo di IC_k e ed IS_k siano esatte per funzioni polinomiali di secondo grado, $f \in \mathbb{P}_2$, si ottiene l'analogo della formula di Simpson per il calcolo di funzioni rapidamente oscillanti. Le formule così ottenute sono chiamate formule di Filon[9]. Esse si costruiscono come segue: suddividiamo l'intervallo $[a,b]$ in $2N$ sottointervalli di ampiezza $h = (b-a)/(2N)$ mediante i punti $x_j = a + jh$, $j = 0, \ldots, 2N$. Definiamo le quantità

$$\begin{aligned} C_p &= \frac{1}{2} f_0 \cos kx_0 + f_2 \cos kx_2 + f_4 \cos kx_4 + \ldots \\ &\quad + f_{2N-2} \cos kx_{2N-2} + \frac{1}{2} f_{2N} \cos kx_{2N}, \end{aligned} \tag{5.32}$$

[9] Introdotte nell'articolo di L.N.G.Filon, *On a quadrature formula for trigonometric integrals*, Proc. Roy. Soc. Edinburgh **49** (1928-29), pp.38-47.

$$C_d \quad = \quad f_1 \cos kx_1 + f_3 \cos kx_3 + \ldots + f_{2N-1} \cos kx_{2N-1}, \tag{5.33}$$

dove $f_j \equiv f(x_j)$, $j = 0, \ldots, 2N$, C_p e C_d sono calcolate, rispettivamente, utilizzando i nodi pari e i nodi dispari. Utilizzando le abbreviazioni

$$\vartheta = kh, \quad \alpha(\vartheta) = \frac{\vartheta^2 + \frac{1}{2}\vartheta \sin 2\vartheta - 2\sin^2 \vartheta}{\vartheta^3}, \tag{5.34}$$

$$\beta(\vartheta) = 2\frac{\vartheta(1 + \cos^2 \vartheta) - \sin 2\vartheta}{\vartheta^3}, \quad \gamma(\vartheta) = 4\frac{\sin \vartheta - \vartheta \cos \vartheta}{\vartheta^3}, \tag{5.35}$$

la formula di Filon per il calcolo di integrali di tipo coseno può essere scritta nella forma

$$\int_a^b f(x) \cos kx \, dx = h[(f_{2N} \sin kx_{2N} - f_0 \sin kx_0)\alpha + C_p\beta + C_d\gamma] + E, \tag{5.36}$$

dove l'errore E, che si annulla per $f \in \mathbb{P}_2$, ammette una stima della forma

$$|E_C| \leq \frac{b-a}{180} h^4 (M_4 + 4kM_3), \tag{5.37}$$

dove M_3 ed M_4 sono stime delle norme delle derivate corrispondenti:

$$|f'''(x)| \leq M_3, \quad |f^{(4)}(x)| \leq M_4.$$

Si osservi come per $k = 0$ la formula e la relativa stima dell'errore si riducano alla formula di Simpson composita. Questo si può osservare direttamente utilizzando l'espansione della formula e dei coefficienti per piccoli valori del parametro ϑ. Osserviamo, inoltre, che per valori di ϑ molto piccoli è più conveniente utilizzare l'espansione di Taylor dei coefficienti, per evitare problemi di cancellazione numerica o divisioni.

Analoghe formule valgono per il calcolo di integrali del tipo seno

$$\int_a^b f(x) \sin kx \, dx = h[-(f_{2N} \cos kx_{2N} - f_0 \cos kx_0)\alpha + S_p\beta + S_d\gamma], \tag{5.38}$$

dove le quantità S_p ed S_d sono definite in maniera analoga alle corrispondenti quantità C_p e C_d. Per tale formula vale la stima dell'errore (5.37) che può essere scritta nella forma

$$|E_S| \leq \frac{b-a}{45} h^3 \left(\vartheta M_3 + \frac{h}{4} M_4\right).$$

La stima dell'errore cresce linearmente con ϑ. Sarebbe auspicabile utilizzare queste formule con un valore di ϑ dell'ordine dell'unità.

I codici MATLAB `filonc` e `filons` disponibili in rete implementano le formule di Filon di tipo coseno e seno rispettivamente. Un esempio di applicazione di tali funzioni è riportato in Tabella 5.6 assieme ad un confronto con altre formule.

5.6 Integrazione in più dimensioni

L'integrazione di funzioni in più dimensioni risulta alquanto più complessa del calcolo di integrali in una dimensione. I motivi principali sono i seguenti

Calcolo di $\int_0^{2\pi} x \cos x \sin kx \, dx$					
k	Esatto	Gauss32	Filons32	Simpson128	Simpson256
10	-0.63466518,	-0.63466400	-0.63466454	-0.63467833	-0.63466600
20	-0.31494663,	-1.21274306	-0.31494759	-0.31505179	-0.31495306
30	-0.20967248,	-1.57913198	-0.20973796	-0.21003931	-0.20969428
40	-0.15717787,	-0.95576051	-0.15718011	-0.15809425	-0.15723013
50	-0.12571399,	1.05161961	-0.12571407	-0.12763780	-0.12581771
60	-0.10474885,	-2.09557307	-0.10474965	-0.10840350	-0.10493173
70	-0.08977811,	-2.06116054	-0.08977906	-0.09633620	-0.09007573
80	-0.07855209,	-0.42121765	-0.07855214	-0.09001668	-0.07900944
90	-0.06982179,	1.17434994	-0.06982173	-0.08991873	-0.07049535
100	-0.06283814,	-1.13390359	-0.06283892	-0.09952174	-0.06379871
Calcolo di $\int_0^{2\pi} x \sin x \cos kx \, dx$					
k	Esatto	Gauss32	Filonc32	Simpson128	Simpson256
10	0.06346652,	0.06346782	0.06345213	0.06346266	0.06346628
20	0.01574733,	-0.36174065	0.01572646	0.01573136	0.01574637
30	0.00698908,	-1.48776701	0.00685606	0.00695087	0.00698689
40	0.00392945,	-0.24386080	0.00394955	0.00385510	0.00392546
50	0.00251428,	2.15897383	0.00252039	0.00238303	0.00250787
60	0.00174581,	-1.30995425	0.00174385	0.00152400	0.00173626
70	0.00128254,	1.67197330	0.00128926	0.00091110	0.00126900
80	0.00098190,	1.40438304	0.00098491	0.00034658	0.00096332
90	0.00077580,	-1.67864469	0.00077694	-0.00037520	0.00075091
100	0.00062838,	1.05715404	0.00063203	-0.00171505	0.00059558

Tabella 5.6 Confronto fra formule di Filon, formula di Gauss-Legendre a 32 nodi, formule di Simpson composita con, rispettivamente, 128 e 256 intervalli

1. il numero di punti necessario per "approssimare" una funzione cresce molto rapidamente al crescere della dimensione d (se occorrono N punti per interpolare una funzione $f(x)$ regolare in un dominio, diciamo [0,1], con un errore minore di ε, ne occorreranno N^d per ottenere una analoga approssimazione di una funzione $f(x_1, \ldots, x_d)$, $d > 1$);

2. mentre in una dimensione il dominio di integrazione è sempre un intervallo (finito o infinito), in più dimensioni il dominio può variare con una casistica molto più ampia, e nella scelta della formula di quadratura occorre tenere presente non solo la regolarità della funzione integranda, ma anche la regolarità del dominio di integrazione.

Un motivo legato ai due precedenti è che al crescere della dimensione il rapporto superfice/volume diventa sempre più alto. Potremmo dire, pittoricamente, che per dimensioni molto alte un dominio generico "è quasi tutto bordo!". Per quantificare questo effetto facciamo la seguente considerazione, che ci fa capire quale possa essere la difficoltà del calcolo di un integrale di una funzione in più dimensioni, per dimensioni molto alte. Supponiamo di voler effettuare il calcolo

di un integrale di una funzione in un dominio $B \subset \mathbb{R}^d$, e sia V il volume di tale regione. Supponiamo di dividere tale volume in un certo numero N di regioni cubiche di lato h. Sia N_i il numero di regioni cubiche interne al dominio B e N_b il numero delle regioni cubiche che contengono almeno un punto della frontiera (Si veda la Figura 5.9 nel caso bidimensionale $d = 2$).

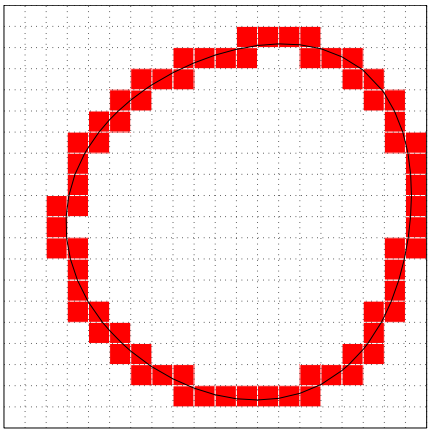

Figura 5.9 Suddivisione di un dominio bidimensionale in quadratini. La linea continua indica il bordo del dominio. I quadratini in grigio contengono almeno un punto della frontiera.

Supponiamo di voler approssimare l'integrale con i contributi ottenuti integrando la funzione solo sull'unione dei cubi interni. Anche ammettendo di calcolare accuratamente il contributo all'integrale di ciascuna regione cubica, ci aspettiamo che una stima dell'errore relativo nel calcolo dell'integrale sia dell'ordine del rapporto N_b/N_i. Il legame tra il numero di regioni interne e il volume è $V = N_i h^d$. Se S è la superficie del bordo di B, il volume della regione vicina alla frontiera è $Sh \approx N_b h^d$. Il rapporto N_b/N_i è dunque circa

$$\frac{N_b}{N_i} \approx \frac{Sh}{V} \sim \frac{S}{V}\left(\frac{V}{N_i}\right)^{1/d} \sim \frac{1}{N_i^{1/d}}.$$

Da questa stima (fin troppo pessimistica) appare evidente che il numero di cubi in cui è necessario suddividere il dominio cresce molto rapidamente all'aumentare della dimensione.

In questa sezione ci limiteremo principalmente a considerare integrali in due dimensioni, e diamo solo qualche cenno ai casi di dimensione più elevata.

Come per integrali in una dimensione, ci proponiamo di calcolare integrali del tipo

$$I[f] = \int_B w(x) f(x)\, dx, \tag{5.39}$$

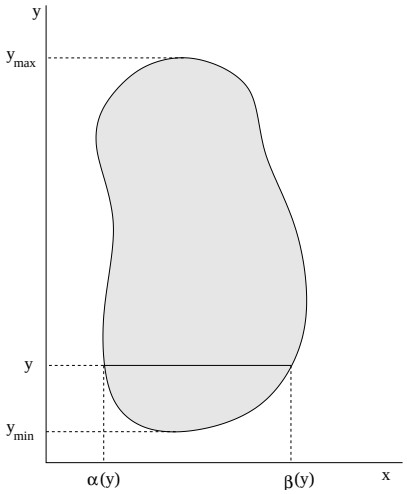

Figura 5.10 Dominio normale rispetto all'asse y, e decomposizione dell'integrale mediante il Teorema di Fubini.

dove $w : \mathbb{R}^d \to \mathbb{R}_+$ denota una funzione peso, ed $f : \mathbb{R}^d \to \mathbb{R}$ indica la funzione da integrare. Il nostro obiettivo è quello di approssimare tale integrale mediante una formula di quadratura del tipo

$$I_n[f] = \sum_{i=1}^{n} w_i f(x_i), \tag{5.40}$$

dove $x_i \in \mathbb{R}^d$, $i = 1, \ldots, n$ sono i nodi e w_i sono i pesi di tale formula.

Senza addentrarci troppo nei dettagli del problema della integrazione in più dimensioni, considereremo i seguenti casi

1. il dominio di integrazione $B \subset \mathbb{R}^d$ sia il prodotto cartesiano di domini di dimensione inferiore, $B = B_1 \otimes B_2$, $B_1 \subset \mathbb{R}^{d_1}$, $B_2 \subset \mathbb{R}^{d_2}$, $d_1 + d_2 = d$;

2. il dominio di integrazione $B \subset \mathbb{R}^2$ sia normale rispetto ad uno degli assi, ad esempio rispetto all'asse y (vedi Figura 5.10);

3. la regione B sia scomponibile o approssimabile mediante unione disgiunta di regioni più semplici (ipercubi o tetraedri d-dimensionali). Si veda la Figura 5.11 per un esempio in due dimensioni, che mostra come la regione Ω a forma di "8", che è una regione non semplicemente connessa e certamente non normale, si possa scomporre in due regioni, Ω_1 ed Ω_2 normali rispetto all'asse y.

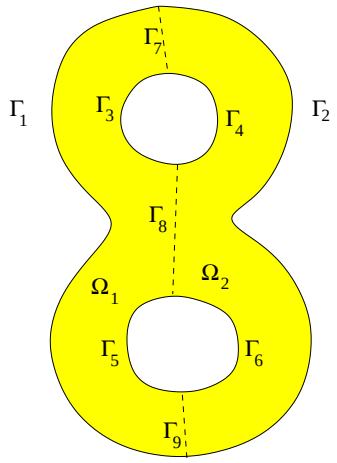

Figura 5.11 Decomposizione di un dominio normale nella unione di due domini normali. La regione Ω in grigio, la cui frontiera è data da $\partial\Omega = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4 \cup \Gamma_5 \cup \Gamma_6$, viene decomposta nell'unione di due domini, Ω_1 e Ω_2, dai tagli individuati da Γ_7, Γ_8 e Γ_9. Si ha così: $\partial\Omega_1 = \Gamma_1 \cup \Gamma_3 \cup \Gamma_5 \cup \Gamma_7 \cup \Gamma_8 \cup \Gamma_9$ e $\partial\Omega_2 = \Gamma_2 \cup \Gamma_4 \cup \Gamma_6 \cup \Gamma_7 \cup \Gamma_8 \cup \Gamma_9$.

Prodotto cartesiano e regole prodotto Consideriamo il caso in cui il dominio di integrazione B sia un prodotto cartesiano di domini di dimensione inferiore. In due dimensioni si ha $B = [a,b] \times [c,d]$. In tal caso se indichiamo con I_n una formula di quadratura per l'intervallo $[a,b]$ e con $\tilde{I}_m$ una formula di quadratura per l'intervallo $[c,d]$,

$$I_n[f] = \sum_{i=1}^{n} w_i f(x_i), \quad \tilde{I}_m[f] = \sum_{i=1}^{m} \tilde{w}_i f(y_i),$$

allora una formula di quadratura per l'integrale di una funzione di due variabili in B si ottiene come prodotto cartesiano delle due formule di quadratura suddette

$$\int_B f(x,y)\,dx\,dy \approx I_n \times \tilde{I}_m[f] = \sum_{i=1}^{n}\sum_{j=1}^{m} w_i \tilde{w}_j f(x_i,y_j). \tag{5.41}$$

Se la formula di quadratura I_n è esatta per la funzione $f(x)$ e la formula $\tilde{I}_m$ è esatta per la funzione $g(y)$, allora la formula prodotto risulta esatta per la funzione $f(x)g(y)$.

Esempio 5.7 [Quadratura gaussiana su una regione quadrata]
Consideriamo una formula per il calcolo di un integrale nel quadrato $[-1,1]^2$ ottenuta come prodotto cartesiano di due formule di quadratura di Gauss-Legendre a due nodi. Si ottiene

$$\int_{-1}^{1}\int_{-1}^{1} f(x,y)\,dx\,y \approx f(\bar{x},\bar{x}) + f(-\bar{x},\bar{x}) + f(\bar{x},-\bar{x}) + f(-\bar{x},-\bar{x}),$$

con $\bar{x} = 1/\sqrt{3}$. La formula ottenuta risulta esatta per funzioni del tipo $f(x,y) = p(x)q(y)$, con $p,q \in \mathbb{P}_3$ e per combinazioni lineari di tali funzioni. In particolare, la formula risulta esatta per tutti i polinomi algebrici di grado minore o uguale a tre nelle due variabili x e y. ■

Esempio 5.8 [Quadratura gaussiana su una regione circolare]
Come secondo esempio consideriamo il caso in cui la regione B sia il cerchio unitario. In tal caso, utilizzando coordinate polari, l'integrale si può scrivere

$$I = \int_{D_1} f(x,y)\,dx\,dy = \int_0^{2\pi} d\vartheta \int_0^1 rf(r\cos\vartheta, r\sin\vartheta)\,dr.$$

In r utilizziamo in nodi $r_1,r_2,\ldots,r_n$ di una formula gaussiana con n nodi e funzione peso r, e in ϑ utilizziamo la formula dei trapezi con m nodi. Si ottiene così la formula

$$I \approx \sum_{k=0}^{m-1}\sum_{i=1}^{n} w_i f(r_i\cos\vartheta_k, r_i\sin\vartheta_k), \quad \vartheta_k \equiv \frac{2\pi k}{m}.$$

Un esempio di nodi e pesi per la funzione peso r in $[0,1]$ può essere calcolata con la funzione `Gauss` riportata in precedenza. ■

Osservazione 5.5 Anche nel caso di formule di quadratura in due dimensioni, si può utilizzare un cambiamento di variabili analogo quello usato nel caso monodimensionale, per riportare il calcolo di un integrale nella regione B al calcolo di un integrale nel quadrato $[-1,1]^2$. Il cambiamento di coordinate in questo caso è dato da

$$\begin{aligned} x &= a(1-\xi)/2 + b(1+\xi)/2, \\ y &= c(1-\eta)/2 + d(1-\eta)/2. \end{aligned}$$

In questo modo, al variare di (ξ,η) in $[-1,1]\times[-1,1]$, il punto (x,y) varia in $[a,b]\times[c,d]$. Tale cambiamento di coordinate può essere generalizzato per ricavare una formula di quadratura per un quadrilatero arbitrario, a partire da una formula di quadratura per il quadrato $[-1,1]\times[-1,1]$.

In questo caso il cambiamento di coordinate è una trasformazione bilineare del tipo

$$x = \sum_{i=1}^{4} C_i(\xi,\eta)x_i, \quad y = \sum_{i=1}^{4} C_i(\xi,\eta)y_i, \tag{5.42}$$

dove

$$\begin{aligned} C_1 &= \frac{(1-\xi)(1-\eta)}{4}, \quad C_2 = \frac{(1+\xi)(1-\eta)}{4}, \\ C_3 &= \frac{(1+\xi)(1+\eta)}{4}, \quad C_4 = \frac{(1-\xi)(1+\eta)}{4}. \end{aligned} \tag{5.43}$$

Tali relazioni si ricavano in maniera elementare dalla seguente considerazione geometrica. I punti P_A e P_B corrispondenti ad un certo valore della coordinata ξ, si ottengono dalla combinazione convessa, rispettivamente dei punti P_1, P_2 e dei punti P_4, P_5

$$P_A = \frac{1-\xi}{2}P_1 + \frac{1+\xi}{2}P_2, \quad P_B = \frac{1-\xi}{2}P_4 + \frac{1+\xi}{2}P_3.$$

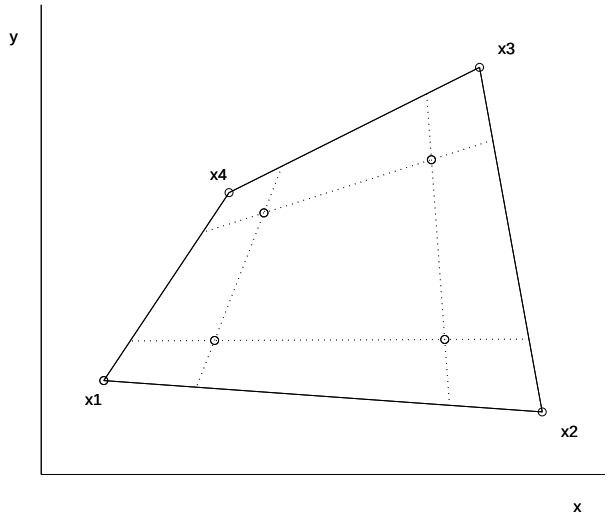

Figura 5.12 Formula di quadratura per un quadrilatero, a partire da una formula di quadratura nel quadrato $[-1,1]^2$. I punti mostrati nell'esempio forniscono i nodi per una formula prodotto di Gauss-Legendre, che risulta esatta per polinomi in x,y di primo grado.

Il generico punto P corrispondente alle coordinate (ξ,η) si ottiene dalla combinazione convessa dei punti P_A e P_B

$$P = \frac{1-\eta}{2} P_A + \frac{1+\eta}{2} P_B.$$

Riordinando i termini si ottengono le relazioni (5.42) e (5.43).

L'integrale di una funzione sul quadrilatero B si trasforma come segue

$$\int_B f(x,y)\,dx\,dy = \int_{-1}^{1}\int_{-1}^{1} f(x(\xi,\eta),y(\xi,\eta))J(\xi,\eta)\,d\xi\,d\eta,$$

dove $x(\xi,\eta)$ e $y(\xi,\eta)$ sono date dalle relazioni (5.42) e (5.43), e lo Jacobiano della trasformazione J è dato da

$$J = \begin{vmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{vmatrix} = \alpha\xi + \beta\eta + \gamma,$$

con

$$\begin{aligned}
\alpha &= -\frac{1}{8}((x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)), \\
\beta &= -\frac{1}{8}((y_1 - y_4)(x_3 - x_2) - (x_1 - x_4)(y_3 - y_2)), \\
\gamma &= \frac{1}{8}\sum_{i=1}^{4} x_i y_{i+1} - x_{i+1} y_i, \quad \text{con} \quad x_5 \equiv x_1,\, y_5 \equiv y_1.
\end{aligned}$$

La Figura 5.12 mostra i nodi della formula di quadratura ottenuta come prodotto cartesiano della formula di quadratura di Gauss-Legendre a due nodi con se stessa. Notiamo che tale formula applicata al quadrilatero in Figura non risulta più esatta per polinomi di grado 3, come nel caso di dominio rettangolare, e neanche per polinomi di secondo grado, poiché se f è un polinomio di secondo grado in x e y, la funzione integranda $f(x(\xi,\eta),y(\xi,\eta)J(\xi,\eta)$ risulta un polinomio di quinto grado in (ξ,η). Tale formula risulta esatta per polinomi di grado 1 in (x,y), poiché in questo caso la funzione integranda risulta un polinomio di terzo grado in (ξ,η). ■

5.6.1 Domini normali

Supponiamo che il dominio di integrazione sia normale rispetto ad uno degli assi. La Figura 5.10 mostra un dominio normale rispetto all'asse y. In tal caso la frontiera del dominio $\Gamma = \partial B$ è data dall'unione di due contributi, $\Gamma = \Gamma_1 \cup \Gamma_2$, ciascuno dei quali rappresenta il grafico di una funzione in una variabile

$$\Gamma_1 = \{(\alpha(y),y), y \in [y_{\min},y_{\max}]\}, \quad \Gamma_2 = \{(\beta(y),y), y \in [y_{\min},y_{\max}]\}.$$

In tal caso l'integrale della funzione si può decomporre, graze al Teorema di Fubini [23], come segue

$$\int_B f(x,y)\,dx\,dy = \int_{y_{\min}}^{y_{\max}} dyF(y), \tag{5.44}$$

dove

$$F(y) \equiv \int_{\alpha(y)}^{\beta(y)} dxf(x,y).$$

In questo modo il problema viene ricondotto al calcolo di diversi integrali in una dimensione. Si ha infatti

$$I_B[f] \equiv \int_B f(x,y)\,dx\,dy \approx \sum_{i=1}^{n} w_iF(y_i),$$

dove w_i e y_i sono nodi e pesi relativi ad una formula di quadratura per una funzione nell'intervallo $[y_{\min},y_{\max}]$. Ciascuno dei termini $F(y_i)$, a sua volta, richiede l'approssimazione di un integrale in una dimensione. Una strategia di tipo adattivo può essere utilizzata nel calcolo dei vari termini $F(y_i)$ e dell'integrale $I_B[f]$.

5.6.2 Formule composite

Come nel caso monodimensionale, anche per il calcolo di integrali multidimensionali si fa spesso ricorso a formule composite, come ad esempio nel caso in cui si voglia utilizzare una formula adattiva per l'integrazione su domini normali visto nel paragrafo precedente. Se il dominio non è normale rispetto ad alcuno degli assi, allora si può decomporre il dominio di integrazione nella unione di domini normali disgiunti. L'integrale verrà quindi approssimato come somma degli integrali nei vari domini. Una strategia comune per il calcolo di integrali multipli è

quello di approssimare il dominio di integrazione e di suddividerlo mediante una griglia. Tale griglia è di solito costituita da una triangolazione oppure da una griglia rettangolare. La Figura 5.13 mostra ad esempio la approssimazione di un dominio mediante una regione poligonale, e la suddivisione della regione mediante una triangolazione.

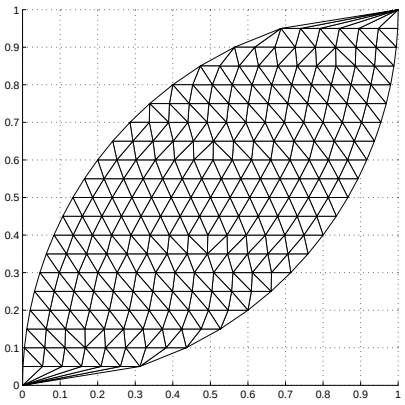

Figura 5.13 Esempio di triangolazione di un dominio "a foglia".

L'integrale su un triangolo di vertici $(x_i,y_i), i = 1,2,3$ può essere approssimato dalla formula

$$\int_T f(x,y)\,dx\,dy \approx \frac{A_T}{3}\sum_{i=1}^{3} f(x_i,y_i).$$

Tale formula rappresenta l'analogo bidimensionale della formula dei trapezi, e viene utilizzata per calcolare l'approssimazione di un integrale mediante una triangolazione.

Dato un insieme di punti del piano $\mathcal{P} = \{(x_i,y_i), \quad i = 1,\ldots,N\}$, una triangolazione è individuata assegnando N_T terne di indici $T_{j,k}, \quad j = 1,\ldots,N_T, k = 1,\ldots,3$. Il numero N_T rappresenta il numero di triangoli, e per ogni triangolo j, i tre indici $T_{j,k}, k = 1,2,3$ individuano i tre vertici del triangolo. Più precisamente, le coordinate dei vertici del triangolo j sono date da $(x_{T_{j,k}},y_{T_{j,k}}), k = 1,2,3$.

Per un insieme di punti $\mathcal{P}$ dato esistono diversi modi in cui tali punti si possono connettere mediante una triangolazione. Una *triangolazione di Delaunay* ha la proprietà caratteristica che il cerchio circoscritto ad ogni triangolo non contiene al suo interno alcun punto dell'insieme $\mathcal{P}$. A parte casi degeneri (in cui un punto di $\mathcal{P}$ può appartenere alla circonferenza circoscritta ad un triangolo), la triangolazione di Delaunay è univoca. MATLAB è in grado di generare automaticamente triangolazioni di Delaunay associate ad una distribuzione di punti. Per dettagli su tale triangolazione e sul suo grafo duale, il *diagramma di Voronoi*, si veda l'help del

MATLAB (`help delaunay` o `help voronoi`), oppure, per esempio, la referenza [43].

Il seguente esempio mostra come costruire una triangolazione di una regione del piano individuata da due curve $x_1(y)$ e $x_2(y)$, e come calcolare una approssimazione dell'integrale, una volta che sia nota la triangolazione.

Esempio 5.9 [Calcolo di integrali su un dominio "a foglia"]
Si voglia calcolare l'integrale della funzione

$$f(x,y) = \cos(10(x^2+y))\exp(y^2+x)$$

nella regione B individuata dalle due curve

$$x_1(y) = 1-\sqrt{1-y^2}, \quad x_2(y) = \sqrt{1-(y-1)^2}.$$

La regione ha la forma "a foglia" riportata congiuntamente ad una sua triangolazione in Figura 5.13. Per il calcolo utilizzeremo diversi metodi.

Il primo metodo è basato sulla scrittura dell'integrale nella forma (5.44). Il calcolo dell'integrale può essere effettuato in MATLAB utilizzando le funzioni `quadtra` costruita in precedenza e la funzione MATLAB `quad8`. Avremo quindi

```
> I1 = quadtra('int1',0,1,1e-4)
I1 =
    1.4131
```

dove la funzione `int1` è definita come

```
function I = int1(y)
I = quad8('fun2',x1(y),x2(y),1e-6,0,y);
```

mentre le funzioni `fun2`, `x1`, e `x2` sono date da

```
function z = fun2(x,y)
z = (1+cos(10*(x.^2+y))).*exp(y.^2+x);

function x = x1(y)
x = 1-sqrt(1-y.^2);

function x = x2(y)
x = sqrt(1-(y-1).^2);
```

Il secondo metodo utilizza il seguente codice MATLAB che calcola una triangolazione del dominio, e poi sfrutta la triangolazione per l'approssimazione numerica dell'integrale.

```
function z=Triregion(dx,dy)
% Sintassi IT=Triregion(dx,dy)
%
% Costruzione di una triangolazione caratterizzata da dx e dy
% nella regione definita da x1 e x2 e calcolo dell'integrale
% mediante la formula dei trapezi su tale triangolazione
%
X = [];
Y = [];
for y=0:dy:1
    Xnew = linspace(x1(y),x2(y),floor((x2(y)-x1(y))/dx+1));
    Ynew = y*ones(size(Xnew));
```

```
    X = [X,Xnew];
    Y = [Y,Ynew];
end
tri = delaunay(X,Y);
nt = length(tri);
s = 0;
At = abs(((X(tri(:,2))-X(tri(:,1))).*(Y(tri(:,3))-Y(tri(:,1))) - ...
    (X(tri(:,3))-X(tri(:,1))).*(Y(tri(:,2))-Y(tri(:,1))))/2;
for i=1:3
    s = s+sum(fun2(X(tri(:,i)),Y(tri(:,i))).*At)/3;
end
z = s;
```

La precedente funzione fornisce il risultato

```
>> I1=Triregion(0.03,0.03)
I1 =
     1.3987
```

La triangolazione qui utilizzata è quella riportata in Figura 5.13. MATLAB consente anche di effettuare il grafico di una funzione calcolata su una triangolazione tramite la funzione `trimesh`. Sarà sufficiente aggiungere alla fine della funzione `Triregion` le seguenti due linee

```
Z = fun2(X,Y);
trimesh(tri,X,Y,Z);
```

per ottenere la Figura 5.14 (dove è stato cambiato manualmente il punto di vista). Si consulti l'help in linea per ulteriori informazioni sulla funzione `trimesh`. Raffinando la griglia, cioè aumentando i valori di $nx = 1/dx$ e $ny = 1/dy$ si ot-

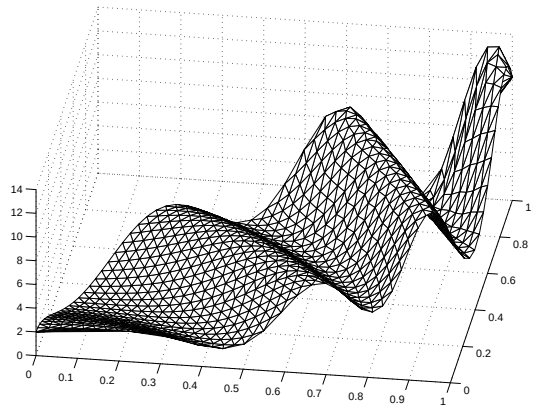

Figura 5.14 La funzione `fun2` ricostruita su una triangolazione

tengono migliori approssimazioni dell'integrale, come riportato nella tabella sotto riportata

nx = ny	IT
20	1.3797
40	1.3966
80	1.4068
160	1.4108
320	1.4123

Come si vede dalla tabella, la convergenza è piuttosto lenta. Questo è dovuto sia al fatto che il metodo dei trapezi non è molto accurato, sia, soprattutto, al fatto che la triangolazione utilizzata non è particolarmente buona. L'errore relativo nell'integrale su ogni triangolo è infatti quadratico nella lunghezza del lato maggiore del triangolo, che non decresce molto rapidamente all'aumentare di nx nella triangolazione su costruita. Una migliore triangolazione porterebbe a dei valori più accurati dell'integrale.

Un terzo metodo è basato su una diversa suddivisione ottenuta tassellando il dominio B mediante dei quadrilateri. Questo può essere ottenuto con il seguente script

```
function IQ=Quadreg(nx,ny,iload)
% Sintassi  IQ=Quadreg(nx,ny,iload)
%
% Costruzione di una quadrangolazione nella regione
% definita da x1(y) e x2(y)
% iload stabilisce se caricare i punti da file (1) o calcolarli (0)
% Per usare la opzione 1 deve esistere un file chiamato XYnxny.dat
% dove nx ed ny sono le dimensioni della griglia
% Una buona distribuzione di punti pu essere ottenuta con la
% function Betterquad
%
dx = 1/nx; dy = 1/ny;
if iload==1       % Carica la griglia da file
    eval(['load XY' num2str(nx) num2str(ny) '.dat'])
    eval(['XY = XY' num2str(nx) num2str(ny) ';']);
    X = reshape(XY(:,1),nx+1,ny+1);
    Y = reshape(XY(:,2),nx+1,ny+1);
else     % Definisce la distribuzione dei punti griglia
    X = [];
    Y = [];
    for y=0:dy:1
        Xnew = linspace(x1(y),x2(y),nx+1)';
        Ynew = y*ones(size(Xnew));
        X = [X,Xnew];
        Ynew = y*ones(size(Xnew));
        Y = [Y,Ynew];
    end
end
% Definisce le matrici per il calcolo di nodi
% e pesi in ciascun quadrilatero
c1 = (2+sqrt(3))/6;
c2 = 1/6;
c3 = (2-sqrt(3))/6;
c4 = 1/6;
C = [c1 c2 c3 c4  0  0  0
      0 c1 c2 c3 c4  0  0
      0  0 c1 c2 c3 c4  0
```

```
      0  0  0 c1 c2 c3 c4];
M = [-1 1 1 -1 ; -1 -1 1 1 ];
M = [M/sqrt(3) ; 1 1 1 1];
% Somma il contributo dei vari quadrilateri
s = 0;
for i=1:nx
  for j=1:ny
    xp = [X(i,j);X(i+1,j);X(i+1,j+1);X(i,j+1)];
    xp = [xp;xp(1:3)];
    yp = [Y(i,j);Y(i+1,j);Y(i+1,j+1);Y(i,j+1)];
    yp = [yp;yp(1:3)];
    xt = C*xp;
    yt = C*yp;
    alpha = ((xp(1)-xp(2))*(yp(3)-yp(4))-(yp(1)-yp(2))*( ...
           xp(3)-xp(4)))/8;
    beta = ((yp(1)-yp(4))*(xp(3)-xp(2))-(xp(1)-xp(4))*( ...
             yp(3)-yp(2)))/8;
    gamma = sum(xp(1:4).*yp(2:5)-xp(2:5).*yp(1:4))/8;
    J = [alpha,beta,gamma]*M;
    s = s + sum(fun2(xt,yt).*J');
  end
end
IQ = s;
```

Il programma costruisce dapprima una suddivisione della regione in quadrilateri, e poi applica la tecnica descritta nell'Osservazione 5.5 basata sulle formule di Gauss-Lobatto. Si noti l'uso della funzione `eval(`*Stringa*`)` che consente di eseguire un'istruzione MATLAB espressa in forma di stringa.

I valori dell'integrale risultano in questo caso

nx = ny	IQ
20	1.41760497412
40	1.40711449885
80	1.40883638537
160	1.41113509238
320	1.41231280471

Un esempio di suddivisione in quadrilateri è riportata in Figura 5.15. Per ottenere tale figura è sufficiente aggiungere alla funzione `Quadreg` le seguenti due righe

```
Z = fun2(X,Y);
mesh(X,Y,Z);
```

Come si vede dalla figura alcuni dei quadrangoli risultano molto distorti. Ci aspettiamo che il metodo descritto sia più accurato se si utilizza una suddivisione più regolare.

A tale scopo possiamo utilizzare il seguente script[10].

```
% Betterquad.m
%
% Migliore distribuzione di punti per una quadrangolazione della
```

[10]Lo script in pratica utilizza il metodo di Gauss-Seidel per punti per la risoluzione dell'equazione di Laplace nel quadrato unitario.

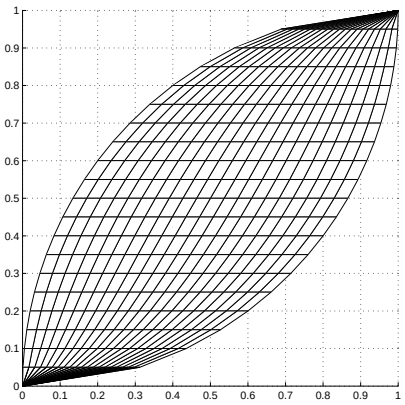

Figura 5.15 Suddivisione della regione B in domini quadrangolari per $nx = ny = 20$.

```
% regione B
% Il programma produce una distribuzione di punti regolari e salva
% la configurazione di punti nel file XYnn.dat
%
niter = 1000;          % Parametro di iterazione
nx = 20; ny = nx;      % Definisce la griglia in x e y
dx = 1/nx; dy = 1/ny;
% calcola la griglia iniziale, come nel codice quadreg.m
X = [];
Y = [];
for y=0:dy:1
    Xnew = linspace(x1(y),x2(y),nx+1)';
    Ynew = y*ones(size(Xnew));
    X = [X,Xnew];
    Ynew = y*ones(size(Xnew));
    Y = [Y,Ynew];
end
% Distribuisce punti uniformemente spaziati lungo i bordi
theta = linspace(0,pi/2,ny+1);
Y(1,:) = sin(theta);
X(1,:) = x1(Y(1,:));
Y(nx+1,:) = 1-cos(theta);
X(nx+1,:) = x2(Y(nx+1,:));
% Rilassa i punti, in modo che le coordinate di ciascun punto siano
% la media delle coordinate dei punti vicini
i = 2:nx; j = 2:ny;
for k=1:niter
            X(i,j) = (X(i+1,j)+X(i,j+1)+X(i-1,j)+X(i,j-1))/4;
            Y(i,j) = (Y(i+1,j)+Y(i,j+1)+Y(i-1,j)+Y(i,j-1))/4;
        end
XY = [X(:),Y(:)];
% Salva i dati in un file
eval(['save XY' num2str(nx) '.dat XY -ascii -double'])
```