

# Quadratura numerica

Alvise Sommariva

Università degli Studi di Padova  
Dipartimento di Matematica

5 aprile 2014

Un classico problema dell'analisi numerica è quello di calcolare l'*integrale definito* di una funzione  $f$  in un intervallo avente estremi di integrazione  $a, b$  (non necessariamente finiti) cioè

$$I_w(f) := I_w(f, a, b) = \int_a^b f(x) w(x) dx$$

dove  $w$  è una funzione peso in  $(a, b)$  [1, p.206, p.270].

La nostra intenzione è di approssimare  $I(f)$  come

$$I_w(f) \approx Q_N(f) := \sum_{i=1}^N w_i f(x_i) \quad (1)$$

I termini  $w_i$  e  $x_i \in [\alpha, \beta]$  sono detti rispettivamente pesi e nodi.

Sia  $(a, b)$  l'intervallo di integrazione (non necessariamente limitato),  $x_1, \dots, x_N$  un insieme di  $N$  punti a due a due distinti ed  $f \in C([a, b])$  una funzione  $w$ -integrabile cioè per cui esista finito  $I_w(f)$ . Se l'intervallo è limitato, per il teorema di Weierstrass e l'integrabilità della funzione peso, questo è vero per qualsiasi **funzione continua** in quanto

$$\left| \int_a^b f(x) w(x) dx \right| \leq \int_a^b |f(x)| w(x) dx \leq \|f\|_\infty \|w\|_1 < +\infty$$

Se

$$p_{N-1}(x) = \sum_{i=1}^N f(x_i) L_i(x)$$

è il polinomio che interpola le coppie  $(x_i, f(x_i))$  con  $i = 1, \dots, N$ , dove al solito  $L_i$  indica l' $i$ -simo polinomio di Lagrange allora

$$\int_a^b f(x) w(x) dx \approx \int_a^b p_{N-1}(x) w(x) dx$$

## Definizione

*Una formula di quadratura*

$$\int_a^b f(x)w(x)dx \approx \sum_{i=1}^N w_i f(x_i) \quad (3)$$

*per cui*

$$w_k = \int_a^b L_k(x) w(x)dx, \quad k = 1, \dots, N \quad (4)$$

*si dice **interpolatoria**.*

## Definizione

*Una formula*

$$\int_a^b f(x)w(x)dx \approx \sum_{i=1}^M w_i f(x_i)$$

*ha **grado di precisione** almeno  $N$  se e solo se è esatta per tutti i polinomi  $f$  di grado inferiore o uguale a  $N$ .*

*Ha inoltre grado di precisione  $N$  se e solo se è esatta per ogni polinomio di grado  $N$  ed esiste un polinomio di grado  $N + 1$  per cui non lo sia.*

Mostriamo ora il seguente

## Teorema

*Una formula*

$$\int_a^b f(x)w(x)dx \approx \sum_{i=1}^N w_i f(x_i)$$

*è interpolatoria se e solo se ha grado di precisione almeno  $N - 1$ .*

## Dimostrazione.

$\Rightarrow$  Se  $f = p_{N-1}$  è un polinomio di grado  $N - 1$  ovviamente corrisponde col polinomio interpolante  $p_{N-1}$  nei nodi a due a due distinti  $x_1, \dots, x_N$  e quindi la formula risulta esatta per polinomi di grado inferiore o uguale a  $N - 1$ , cioè

$$I_w(p_{N-1}) = \sum_{i=1}^N w_i f(x_i), \quad w_i = \int_a^b L_i(x) w(x) dx, \quad i = 1, \dots, N.$$

$\Leftarrow$  Viceversa se è esatta per ogni polinomio di grado  $N - 1$  allora lo è in particolare per i polinomi di Lagrange  $L_i \in \mathcal{P}_{n-1}$ , il che implica che  $w_i = \int_a^b L_i(x) w(x) dx$  e quindi i pesi sono proprio quelli della formula interpolatoria corrispondente nei nodi  $x_1, \dots, x_N$ .  $\square$

Si supponga  $[a, b]$  un intervallo chiuso e limitato di  $\mathbb{R}$  e si ponga  $w \equiv 1$ . Per semplicità di notazione, in questo caso porremo  $I := I_w = I_1$ .

Il primo esempio di formule interpolatorie che consideriamo sono le *regole* di tipo **Newton-Cotes** chiuse (cf. [8, p.336]) che si ottengono integrando l'interpolante di  $f$  in nodi equispaziati

$$x_i = a + \frac{(i-1)(b-a)}{N-1}, \quad i = 1, \dots, N.$$

# Formule di Newton-Cotes

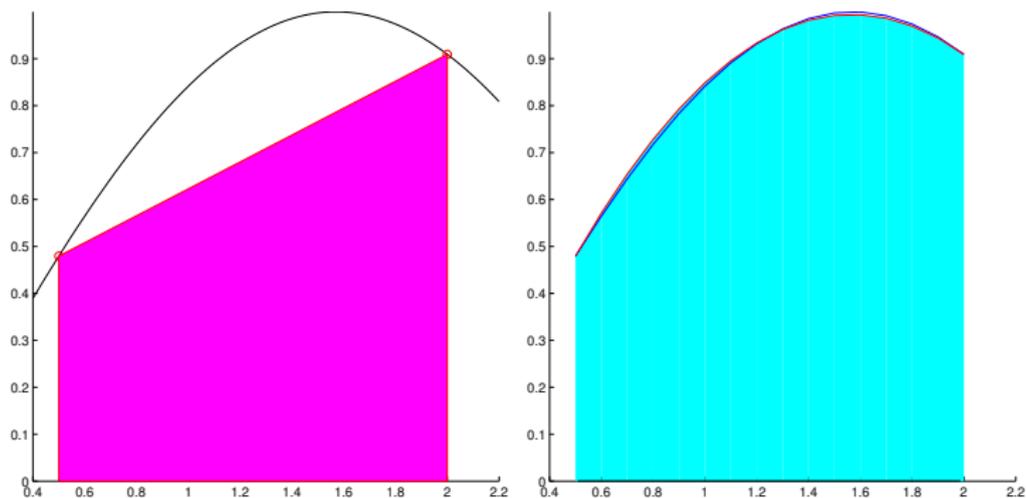


Figura : Regola del trapezio e di Cavalieri-Simpson per il calcolo di  $\int_{0.5}^2 \sin(x) dx$  (rispettivamente area in magenta e in azzurro).

Alcune classiche regole sono:

► **regola del trapezio**

$$I(f) \approx S_1(f) := S_1(f, a, b) := \frac{(b-a)(f(a) + f(b))}{2}$$

avente *grado di precisione* 1, cioè esatta per polinomi di grado inferiore o uguale a 1; si può dimostrare (con un po' di fatica) dal teorema del resto per l'interpolazione polinomiale (cf. [3, p.132]) che l'errore della regola del trapezio [14] è

$$E_1(f) := I(f) - S_1(f) = \frac{-h^3}{12} f^{(2)}(\xi)$$

per qualche  $\xi \in (a, b)$ ;

## ► regola di Cavalieri-Simpson

$$I(f) \approx S_3(f) := S_3(f, a, b) := \frac{b-a}{6} \left[ f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right]$$

avente *grado di precisione* 3, cioè esatta per polinomi di grado inferiore o uguale a 3; si può dimostrare (non facile!) che l'errore della regola di Cavalieri-Simpson [15] è

$$E_3(f) := I(f) - S_3(f) = \frac{-h^5}{90} f^{(4)}(\xi), \quad h = \frac{b-a}{2}$$

per qualche  $\xi \in (a, b)$ ;

# Formule di Newton-Cotes

Vediamo calcolando i pesi, che in effetti le due formule sono interpolatorie. Partiamo dalla regola del trapezio. Posti  $x_1 = a$ ,  $x_2 = b$  abbiamo che

$$L_1(x) = \frac{x-b}{a-b}, \quad L_2(x) = \frac{x-a}{b-a}$$

e quindi visto che  $w \equiv 1$  abbiamo

$$\begin{aligned} w_1 &= \int_a^b L_1(x) dx = \int_a^b \frac{x-b}{a-b} dx = \frac{1}{a-b} \int_a^b (x-b) dx \\ &= \frac{1}{a-b} \frac{(x-b)^2}{2} \Big|_a^b = \frac{1}{a-b} \frac{(x-b)^2}{2} \Big|_a^b \\ &= \frac{1}{a-b} \frac{-(a-b)^2}{2} = \frac{b-a}{2} \end{aligned} \tag{5}$$

Inoltre

$$\begin{aligned}w_2 &= \int_a^b L_2(x) dx = \int_a^b \frac{x-a}{b-a} dx \\&= \frac{1}{b-a} \int_a^b (x-a) dx = \frac{1}{b-a} \frac{(x-a)^2}{2} \Big|_a^b \\&= \frac{1}{b-a} \frac{(x-a)^2}{2} \Big|_a^b = \frac{1}{b-a} \frac{(b-a)^2}{2} \\&= \frac{b-a}{2}\end{aligned}\tag{6}$$

Per quanto riguarda la regola di Cavalieri-Simpson i ragionamenti sono analoghi. D'altra parte essendo quelle dei trapezi e Simpson regole rispettivamente aventi 2 e 3 punti con grado 2 e 4, allora sono entrambe interpolatorie.

Per ulteriori dettagli si confronti [1, p.252-258], [8, p.333-336].

- ▶ Qualora le funzioni da integrare non siano sufficientemente derivabili, una stima dell'errore viene fornita dalle formule dell'errore via **nucleo di Peano** ([1, p.259]).
- ▶ Ricordiamo che per  $N \geq 8$  le formule di Newton-Cotes chiuse hanno pesi di segno diverso e sono **instabili** dal punto di vista della propagazione degli errori (cf. [3, p.196]).

Si suddivide l'intervallo (chiuso e limitato)  $[a, b]$  in  $N$  subintervalli  $T_j = [x_j, x_{j+1}]$  tali che  $x_j = a + jh$  con  $h = (b - a)/N$ . Dalle proprietà dell'integrale

$$\int_a^b f(x) dx = \sum_{j=0}^{N-1} \int_{x_j}^{x_{j+1}} f(x) dx \approx \sum_{j=0}^{N-1} S(f, x_j, x_{j+1}) \quad (7)$$

dove  $S$  è una delle regole di quadratura finora esposte (ad esempio  $S_3(f)$ ). Le formule descritte in (7) sono dette **composte**.

Due casi particolari sono

- ▶ *formula composta dei trapezi*

$$S_1^{(c)} := h \left[ \frac{f(x_0)}{2} + f(x_1) + \dots + f(x_{N-1}) + \frac{f(x_N)}{2} \right] \quad (8)$$

il cui errore è

$$E_1^{(c)}(f) := I(f) - S_1^{(c)}(f) = \frac{-(b-a)}{12} h^2 f^{(2)}(\xi), \quad h = \frac{(b-a)}{N}$$

per qualche  $\xi \in (a, b)$ ;

# Implementazione Matlab di alcune formule composte

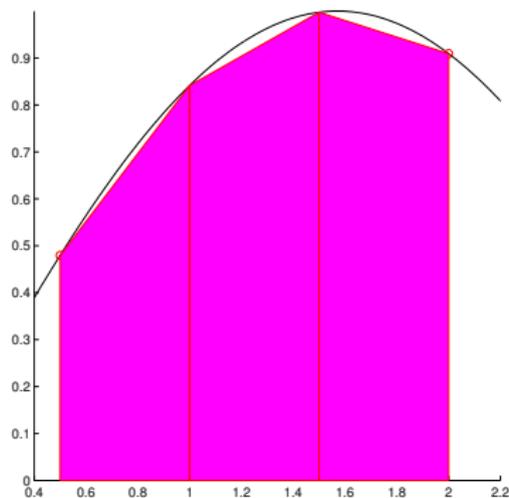


Figura : Formula dei trapezi composta per il calcolo di  $\int_{0.5}^2 \sin(x) dx$  (area in magenta).

# Formule di Newton-Cotes composte

- ▶ *formula composta di Cavalieri-Simpson* fissati il numero  $N$  di subintervalli e i punti  $x_k = a + kh/2$  dove  $h = \frac{b-a}{N}$  sia

$$I(f) \approx S_3^{(c)}(f) := \frac{h}{6} \left[ f(x_0) + 2 \sum_{r=1}^{N-1} f(x_{2r}) + 4 \sum_{s=0}^{N-1} f(x_{2s+1}) + f(x_{2N}) \right]; \quad (9)$$

il cui errore è

$$E_3^{(c)}(f) := I(f) - S_3^{(c)}(f) = \frac{-(b-a)}{180} \left( \frac{h}{2} \right)^4 f^{(4)}(\xi)$$

per qualche  $\xi \in (a, b)$ .

# Implementazione Matlab di alcune formule composte

Mostreremo di seguito un'implementazione in Matlab/Octave della formula composta dei trapezi e di Cavalieri-Simpson.

```
function [x,w]=trapezi_composta(N,a,b)

% FORMULA DEI TRAPEZI COMPOSTA.
% INPUT:
% N: NUMERO SUBINTERVALLI.
% a, b: ESTREMI DI INTEGRAZIONE.
% OUTPUT:
% x: NODI INTEGRAZIONE.
% w: PESI INTEGRAZIONE (INCLUDE IL PASSO!).

h=(b-a)/N;           % PASSO INTEGRAZIONE.
x=a:h:b; x=x';      % NODI INTEGRAZIONE.
w=ones(N+1,1);      % PESI INTEGRAZIONE.
w(1)=0.5; w(N+1)=0.5;
w=w*h;
```

# Implementazione Matlab di alcune formule composte

La funzione `trapezi_composta` appena esposta calcola i nodi e i pesi della omonima formula composta.

L'unica difficoltà del codice consiste nel calcolo dei pesi  $w$ . Essendo per loro definizione

$$I(f) \approx S_1^c(f) := \sum_{i=0}^N w_i f(x_i) \quad (10)$$

come pure per (8)

$$S_1^{(c)} := h \left[ \frac{f(x_0)}{2} + f(x_1) + \dots + f(x_{N-1}) + \frac{f(x_N)}{2} \right] \quad (11)$$

deduciamo che  $w_0 = w_N = h/2$  mentre  $w_1 = \dots = w_{N-1} = h$ , cosa che giustifica le ultime linee della function `trapezi_composta`.

# Implementazione Matlab di alcune formule composte: trapz

Si potrebbe usare il comando Matlab trapz nella sua implementazione

```
>> help trapz
TRAPZ Trapezoidal numerical integration.
  Z = TRAPZ(Y) computes an approximation of the
  integral of Y via
  the trapezoidal method (with unit spacing). To
  compute the integral
  for spacing different from one, multiply Z by the
  spacing increment.

  For vectors, TRAPZ(Y) is the integral of Y.
  ... ..
```

e sostituire la parte relativa al calcolo dei pesi con

```
I=h*trapz(fx);
```

# Implementazione Matlab di alcune formule composte

Vediamone i dettagli in Matlab (versione 6.1) per il calcolo di

$$\int_0^1 \sin(x) dx = -\cos(1) - (-\cos(0)) = -\cos(1) + 1 \\ \approx 0.45969769413186.$$

sia utilizzando la funzione `trapz` che `trapezi_composta`

```
>> format long;
>> [x,w]=trapezi_composta(10,0,1);
>> fx=sin(x);
>> I_trapezi_composta=w'*fx
I_trapezi_composta =
    0.45931454885798
>> h=(1-0)/10;
>> I_trapz=h*trapz(fx)
I_trapz =
    0.45931454885798
>>
```

## Nota.

- ▶ *Per implementare la regola è del tutto equivalente usare la function `trapezi_composta` o `trapz`.*
- ▶ *Si osserva che è sbagliato chiamare `trapz` senza il passo  $h$  (nell'esempio non si dividerebbe per 10, e invece di 0.45931454885798 avremmo 4.5931454885798).*
- ▶ *Evitiamo il diretto utilizzo di `trapz` perchè non presente in alcune vecchie versioni di Octave (ma non nella più recente 2.1.73).*

# Implementazione Matlab di alcune formule composte

Per quanto riguarda la formula di Cavalieri-Simpson composta

```
function [x,w]=simpson_composta(N,a,b)
% FORMULA DI SIMPSON COMPOSTA.
% INPUT:
% N: NUMERO SUBINTERVALLI.
% a, b: ESTREMI DI INTEGRAZIONE.
% OUTPUT:
% x: INTEGRAZIONE.
% w: PESI INTEGRAZIONE (INCLUDE IL PASSO!).
h=(b-a)/N; % AMPIEZZA INTERVALLO.
x=a:(h/2):b; x=x'; % NODI INTEGRAZIONE.
w=ones(2*N+1,1); % PESI INTEGRAZIONE.
w(3:2:2*N-1,1)=2*ones(length(3:2:2*N-1),1);
w(2:2:2*N,1)=4*ones(length(2:2:2*N),1);
w=w*h/6;
```

# Implementazione Matlab di alcune formule composte

Una volta noti il vettore (colonna)  $x$  dei nodi e  $w$  dei pesi di integrazione, se la funzione  $f$  è richiamata da un m-file  $f.m$ , basta

```
fx=f(x); % VALUT. FUNZIONE.  
I=w'*fx; % VALORE INTEGRALE.
```

per calcolare il risultato fornito dalla formula di quad. composta.

Ricordiamo che se  $\mathbf{w} = (\mathbf{w}_k)_{k=1,\dots,M}$ ,  $\mathbf{fx} = (f(x_k))_{k=1,\dots,M}$  sono due vettori colonna allora il prodotto scalare

$$\mathbf{w} * \mathbf{fx} := \sum_{k=1}^M \mathbf{w}_k \cdot \mathbf{fx}_k := \sum_{k=1}^M \mathbf{w}_k \cdot f(x_k)$$

si scrive in Matlab/Octave come  $\mathbf{w}' * \mathbf{fx}$  (dimensionalmente il prodotto di un vettore  $1 \times M$  con un vettore  $M \times 1$  dà uno scalare (cioè un vettore  $1 \times 1$ )).

# Implementazione Matlab di alcune formule composte

Applichiamo ora la formula composta di Cavalieri-Simpson all'esempio precedente in cui

$$\int_0^1 \sin(x) dx \approx 0.45969769413186$$

ottenendo

```
>> format long;
>> [x,w]=simpson_composta(5,0,1);
>> fx=sin(x);
>> I_simpson=w'*fx
I_simpson =
    0.45969794982382
>> length(x)
ans =
    11
>>
```

# Implementazione Matlab di alcune formule composte

Facciamo ora un altro esempio. Calcoliamo numericamente utilizzando le formule composte sopra citate

$$\int_{-1}^1 x^{20} dx = (1^{21}/21) - (-1)^{21}/21 = 2/21 \approx 0.09523809523810.$$

A tal proposito scriviamo il seguente codice demo\_composte.

```
a=-1; b=1; N=11;      %SCEGLIERE N DISPARI.
N_trap=N-1;
[x_trap,w_trap]=trapezi_composta(N_trap,a,b);
fx_trap=x_trap.^20;      % VALUT. FUNZIONE.
I_trap=w_trap'*fx_trap;  % TRAPEZI COMPOSTA.
N_simpson=(N-1)/2;
[x_simp,w_simp]=simpson_composta(N_simpson,a,b)
fx_simp=x_simp.^20;      % VALUT. FUNZIONE.
I_simp=w_simp'*fx_simp;  % SIMPSON COMPOSTA.
fprintf('\n\t [TPZ COMP][PTS]:%4.0f',length(x_trap));
fprintf('\n\t [TPZ COMP][RIS]:%14.14f',I_trap);
fprintf('\n\t [SIMPS.COMP][PTS]:%4.0f',length(x_simp));
fprintf('\n\t [SIMPS.COMP][RIS]:%14.14f',I_simp);
```

ottenendo

```
[TPZ COMP] [PTS]: 11  
[TPZ COMP] [RIS]: 0.20462631505024  
[SIMPS .COMP] [PTS]: 11  
[SIMPS .COMP] [RIS]: 0.13949200364447
```

Si può vedere che usando formule di tipo **gaussiano** (cf. [13]) o di tipo **Clenshaw-Curtis** (cf. [12], [10], [11]) a parità di valutazioni della funzione  $f$  avremmo ottenuto

```
[GAUSS-LEGENDRE ]: 0.095238095238095649  
[CLENISHAW-CURTIS ]: 0.094905176204004307
```

col costo aggiuntivo di dover calcolare tramite complicati algoritmi i pesi e i nodi di Gauss o i nodi di Clenshaw-Curtis via un IFFT.

Usando la formula composta dei trapezi e di Cavalieri-Simpson, con  $n = 2, 4, 8, \dots, 512$  suddivisioni equispaziate dell'intervallo di integrazione, calcolare

- ▶  $\int_0^\pi \exp(x) \cdot \cos(4x) dx = \frac{\exp(\pi)-1}{17};$
- ▶  $\int_0^1 x^{5/2} dx = \frac{2}{7};$
- ▶  $\int_{-\pi}^\pi \exp(\cos(x)) dx = 7.95492652101284;$
- ▶  $\int_0^{\pi/4} \exp(\cos(x)) dx = 1.93973485062365;$
- ▶  $\int_0^1 x^{1/2} dx = \frac{2}{3}.$

Descrivere come decresce l'errore in scala semilogaritmica e quale sia l'andamento del rapporto  $E_n(f)/E_{n+1}(f)$ , dove  $E_n = |I(f) - I_n(f)|$  con  $I(f)$  integrale esatto e  $I_n(f)$  valore fornito dalla formula di quadratura.

## Esercizio facoltativo (non facile)

Aiutandosi con quanto fornito online relativamente al metodo di Romberg, fornirne una implementazione e testarla sulle funzioni, utilizzando  $n = 2, 4, 8, \dots, 512$  suddivisioni equispaziate dell'intervallo di integrazione, calcolando

- ▶  $\int_0^\pi \exp(x) \cdot \cos(4x) dx = \frac{\exp(\pi)-1}{17}$ ;
- ▶  $\int_0^1 x^{5/2} dx = \frac{2}{7}$ ;
- ▶  $\int_{-\pi}^\pi \exp(\cos(x)) dx = 7.95492652101284$ ;
- ▶  $\int_0^{\pi/4} \exp(\cos(x)) dx = 1.93973485062365$ ;
- ▶  $\int_0^1 x^{1/2} dx = \frac{2}{3}$ .

Lo si può applicare in tutti i casi considerati?

Nelle formule interpolatorie di Newton-Cotes (come ad esempio la regola del Trapezio o di Cavalieri-Simpson) i nodi  $x_1, \dots, x_n$  sono equispaziati e il grado di precisione  $\delta$  è generalmente uguale almeno a  $n - 1$  ma in alcuni casi, come per la regola di Cavalieri-Simpson, uguale al numero di nodi  $n$ .

Consideriamo ora **formule che a parità di nodi hanno grado di precisione maggiore di  $n$** .

Sia  $w : (a, b) \rightarrow \mathbb{R}$  (non necessariamente limitato) è una funzione peso, cioè tale che (cf. [1, p.206, p.270])

1.  $w$  è nonnegativa in  $(a, b)$ ;
2.  $w$  è integrabile in  $[a, b]$ ;
3. esista e sia finito

$$\int_a^b |x|^n w(x) dx$$

per ogni  $n \in \mathbb{N}$ ;

4. se

$$\int_a^b g(x)w(x) dx$$

per una qualche funzione nonnegativa  $g$  allora  $g \equiv 0$  in  $(a, b)$ .

Tra gli esempi più noti ricordiamo

1. *Legendre*:  $w(x) \equiv 1$  in  $[a, b]$  limitato;
2. *Jacobi*:  $w(x) = (1-x)^\alpha (1+x)^\beta$  in  $(-1, 1)$  per  $\alpha, \beta \geq -1$ ;
3. *Chebyshev*:  $w(x) = \frac{1}{\sqrt{1-x^2}}$  in  $(-1, 1)$ ;
4. *Laguerre*:  $w(x) = \exp(-x)$  in  $[0, \infty)$ ;
5. *Hermite*:  $w(x) = \exp(-x^2)$  in  $(-\infty, \infty)$ ;

Si supponga ora di dover calcolare per qualche funzione

$$f : (a, b) \rightarrow \mathbb{R}$$

$$I_w(f) := \int_a^b f(x)w(x) dx.$$

**Il problema è evidentemente più generale** di quello di calcolare un integrale del tipo  $\int_a^b f(x)dx$  con  $f \in C([a, b])$ ,  $[a, b]$  limitato, visto che

- ▶ l'integranda  $fw$  non è necessariamente continua in  $[a, b]$  (si consideri ad esempio il peso di Chebyshev che ha una singolarità in  $a = -1$ ,  $b = 1$ )
- ▶ oppure può succedere che l'intervallo sia illimitato come nel caso del peso di Laguerre o Hermite.

Esistono nodi  $x_1, \dots, x_n$  e pesi  $w_1, \dots, w_n$  (detti di *Gauss-nome funzione peso*) per cui le relative formule di quadratura di tipo interpolatorio abbiano grado di precisione  $\delta = 2n - 1$ , cioè calcolino esattamente

$$\int_a^b p(x)w(x) dx$$

per ogni polinomio  $p$  il cui grado è minore o uguale a  $2n - 1$ ?  
La risposta è affermativa, come si può vedere in [1, p.272].

## Teorema

*Per ogni  $n \geq 1$  esistono e sono unici dei nodi  $x_1, \dots, x_n$  e pesi  $w_1, \dots, w_n$  per cui il grado di precisione sia almeno  $2n - 1$ . I nodi sono gli zeri del polinomio ortogonale di grado  $n$ ,*

$$\phi_n(x) = A_n \cdot (x - x_1) \cdot \dots \cdot (x - x_n)$$

*e i corrispettivi pesi sono*

$$w_i = \int_a^b L_i(x)w(x)dx = \int_a^b L_i(x)w(x)dx, \quad i = 1, \dots, n.$$

## Dimostrazione.

Tratta da [3, p.209]. Per prima cosa mostriamo che in effetti con tale scelta dei nodi la formula interpolatoria ha grado di precisione almeno  $2n - 1$ , che i pesi sono univocamente determinati e positivi. Siano  $p_{2n-1} \in \mathcal{P}_{2n-1}$  e  $q_{n-1}, r_{n-1} \in \mathcal{P}_{n-1}$  tali che

$$p_{2n-1} = q_{n-1}\phi_n + r_{n-1}.$$

- ▶  $\int_a^b q_{n-1}(x)\phi_n(x)w(x)dx = (q_{n-1}, \phi_n)_w = 0$ , poichè  $\phi_n$  è il polinomio ortogonale rispetto  $w$  di grado  $n$ ; infatti essendo

$$(\phi_k, \phi_n)_w = 0, \quad k = 0 < n$$

necessariamente da  $q_{n-1} = \sum_{k=0}^{n-1} \gamma_k \phi_k$  abbiamo

$$(q_{n-1}, \phi_n)_w = \left( \sum_{k=0}^{n-1} \gamma_k \phi_k, \phi_n \right)_w = \sum_{k=0}^{n-1} \gamma_k (\phi_k, \phi_n)_w = 0$$

## Dimostrazione.

- ▶ *la formula è interpolatoria per costruzione (vedere la definizione dei pesi!), per cui esatta per ogni polinomio di grado  $n - 1$  in quanto basata su  $n$  punti a due a due distinti;*
- ▶ *se  $x_k$  è uno zero di  $\phi_n$  allora*

$$p_{2n-1}(x_k) = q_{n-1}(x_k)\phi_n(x_k) + r_{n-1}(x_k) = r_{n-1}(x_k).$$

## Dimostrazione.

Quindi, abbiamo

$$\begin{aligned}\int_a^b p_{2n-1}(x)w(x)dx &= \int_a^b q_{n-1}(x)\phi_n(x)w(x)dx + \\ &+ \int_a^b r_{n-1}(x)w(x)dx \\ &= 0 + \int_a^b r_{n-1}(x)w(x)dx \\ &= \sum_{k=1}^n w_k r_{n-1}(x_k) = \sum_{k=1}^n w_k p_{2n-1}(x_k)\end{aligned}$$

per cui la formula di Gauss ha grado di precisione almeno  $2n - 1$ .

## Dimostrazione.

*Inoltre, come dimostrato da Stieltjes nel 1884, i pesi sono positivi, poichè in particolare la formula è esatta per ognuno dei quadrati dei polinomi di Lagrange relativo ai punti  $x_1, \dots, x_n$  per cui*

$$0 < \int_a^b L_j^2(x)w(x)dx = \sum_{k=1}^n w_k L_j^2(x_k) = w_j.$$

*in quanto  $\deg(L_j^2)=2(n-1)$  e la formula ha grado di precisione almeno  $2n-1$ .*

## Dimostrazione.

*Se esistesse un'altra formula interpolatoria con grado di precisione almeno  $2n - 1$  e avesse nodi  $\{\tilde{x}_j\}_{j=1,\dots,n}$  e pesi  $\{\tilde{w}_j\}_{j=1,\dots,n}$  per prima cosa i pesi sarebbero positivi poichè il grado di precisione è almeno  $2n - 1$  e quindi sarebbe esatta per il  $j$ -simo polinomio di Lagrange  $\tilde{L}_j$  da cui*

$$0 < \int_a^b \tilde{L}_j^2(x) w(x) dx = \sum_{k=1}^n \tilde{w}_k \tilde{L}_j^2(\tilde{x}_k) = \tilde{w}_j.$$

## Dimostrazione.

D'altra parte se  $\tilde{L}_j$  è il  $j$ -simo polinomio di Lagrange (avente grado  $n - 1$ ), poichè  $\phi_n$  è il polinomio ortogonale di grado  $n$  rispetto al peso  $w$ , e  $\tilde{w}_j > 0$  abbiamo che da

$$\begin{aligned} 0 &= (\phi_n, \tilde{L}_j)_w = \int_a^b \phi_n(x) \tilde{L}_j(x) w(x) dx = \sum_{k=1}^n \tilde{w}_k \tilde{L}_j(\tilde{x}_k) \phi_n(\tilde{x}_k) \\ &= \tilde{w}_j \cdot \phi_n(\tilde{x}_j) \end{aligned} \quad (13)$$

necessariamente  $x_j = \tilde{x}_j$  e visto che questo implica  $L_j = \tilde{L}_j$  ricaviamo anche

$$w_j = \int_a^b L_j^2(x) w(x) dx = \int_a^b \tilde{L}_j^2(x) w(x) dx = \tilde{w}_j$$

per cui la formula gaussiana cercata è unica.

□

# Sull'errore di quadratura delle formule di Newton-Cotes e di Gauss

Riguardo gli errori compiuti da alcune formule di quadratura (cf.[1, p.264])

## Teorema

Sia la regola di Newton-Cotes  $I(f) \approx I_n(f) = \sum_{i=0}^n w_{i,n} f(x_{i,n})$ .

- ▶ se **n è pari** e  $f \in C^{(n+2)}([a, b])$  allora

con 
$$I(f) - I_n(f) = C_n h^{n+3} f^{(n+2)}(\eta), \quad \eta \in (a, b)$$

$$C_n = \frac{1}{(n+2)!} \int_0^n \mu^2(\mu-1) \dots (\mu-n) d\mu;$$

- ▶ se **n è dispari** e  $f \in C^{(n+1)}([a, b])$  allora

con 
$$I(f) - I_n(f) = C_n h^{n+2} f^{(n+1)}(\eta), \quad \eta \in (a, b)$$

$$C_n = \frac{1}{(n+1)!} \int_0^n \mu(\mu-1) \dots (\mu-n) d\mu;$$

# Sull'errore di quadratura delle formule di Newton-Cotes e di Gauss

Per quanto concerne le formule gaussiane (cf.[1, p.272], cf.[3, p.264], cf.[5, p.344]) ricordiamo il seguente teorema di Markov

## Teorema

*Sia  $f \in C^{(2n)}(a, b)$  con  $(a, b)$  compatto e supponiamo*

$$I_w(f) = \int_a^b f(x)w(x)dx \approx I_n(f) = \sum_{i=1}^n w_{i,n}f(x_{i,n})$$

*sia una formula gaussiana rispetto alla funzione peso  $w$ . Allora*

$$E_n(f) := I_w(f) - I_n(f) = \frac{\gamma_n}{A_n^2(2n)!} f^{(2n)}(\eta), \quad \eta \in (a, b)$$

*dove  $A_n$  è il coefficiente di grado massimo del polinomio ortogonale  $\phi_n$  di grado  $n$ ,  $\gamma_n = \int_a^b \phi_n^2(x)w(x)dx$ .*

# Sull'errore di quadratura delle formule di Newton-Cotes e di Gauss

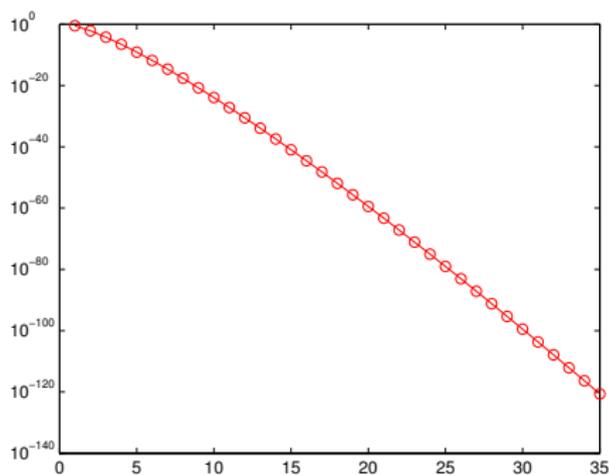


Figura : Grafico in scala semilogaritmica della funzione  $\frac{2^{2n+1}(n!)^4}{(2n+1)[(2n)!]^3}$ .

In particolare, se  $w \equiv 1$ ,  $[a, b] \equiv [-1, 1]$  allora

$$E_n(f) = \frac{2^{2n+1}(n!)^4}{(2n+1)[(2n)!]^3} f^{(2n)}(\eta), \quad \eta \in (-1, 1).$$

Sia  $(a, b)$  un intervallo non necessariamente compatto e  $w$  una funzione peso in  $(a, b)$ . Inoltre supponiamo

$$I_w(f) := \int_a^b f(x)w(x) dx \approx I_n(f) := \sum_{j=1}^{\eta_n} w_j f_j, \quad \text{con } f_j = f(x_j) \tag{14}$$

e che invece di  $\{f_j\}_j$  si disponga di una loro approssimazione  $\{\tilde{f}_j\}_j$ .

# Stabilità di una formula di quadratura

Di conseguenza al posto di  $I_n(f)$  si calcola

$$\tilde{I}_n(f) = \sum_{j=1}^{\eta_n} w_j \tilde{f}_j,$$

ed è

$$\begin{aligned} |I_n(f) - \tilde{I}_n(f)| &= \left| \sum_{j=1}^{\eta_n} w_j (f_j - \tilde{f}_j) \right| \leq \sum_{j=1}^{\eta_n} |w_j| |f_j - \tilde{f}_j| \\ &\leq \left( \sum_{j=1}^{\eta_n} |w_j| \right) \cdot \max_j |f_j - \tilde{f}_j| \end{aligned} \quad (15)$$

Quindi la quantità

$$\sum_{j=1}^{\eta_n} |w_j|$$

è un **indice di stabilità** della formula di quadratura.

# Stabilità di una formula di quadratura

Supponiamo che alcuni pesi siano negativi e che la formula abbia almeno grado di precisione 0. Siano  $\{w_j^+\}_{j=1,\dots,\eta_{n+}}$  e  $\{w_l^-\}_{l=1,\dots,\eta_{n-}}$  rispettivamente i pesi positivi e negativi. Di conseguenza

$$\int_a^b w(x) dx = \sum_{k=1}^{\eta_n} w_k = \sum_{j=1}^{\eta_{n+}} w_j^+ + \sum_{l=1}^{\eta_{n-}} w_l^- \quad (16)$$

$$\sum_{k=1}^{\eta_n} |w_k| = \sum_{j=1}^{\eta_{n+}} |w_j^+| + \sum_{l=1}^{\eta_{n-}} |w_l^-| = \sum_{j=1}^{\eta_{n+}} w_j^+ - \sum_{l=1}^{\eta_{n-}} w_l^- \quad (17)$$

da cui essendo  $w_l^- < 0$

$$\begin{aligned}\sum_{k=1}^{\eta_n} |w_k| &= \sum_{j=1}^{\eta_{n+}} w_j^+ - \sum_{l=1}^{\eta_{n-}} w_l^- \\ &= \left( \int_a^b w(x) dx - \sum_{j=1}^{\eta_{n-}} w_j^- \right) - \sum_{l=1}^{\eta_{n-}} w_l^- \\ &= \int_a^b w(x) dx - 2 \sum_{l=1}^{\eta_{n-}} w_l^- \geq \int_a^b w(x) dx \quad (18)\end{aligned}$$

mentre se i pesi fossero tutti positivi avremmo

$$\sum_{k=1}^{\eta_n} |w_k| = \sum_{j=1}^{\eta_{n+}} w_j^+ - \sum_{l=1}^{\eta_{n-}} w_l^- = \sum_{j=1}^{\eta_{n+}} w_j^+ = \int_a^b w(x) dx$$

per cui maggiore è  $-2 \sum_{l=1}^{\eta_{n-}} w_l^- > 0$  e peggiore è la stabilità della formula di quadratura.

Se  $(a, b)$  è limitato allora l'operatore lineare  $I_n : C([a, b]) \rightarrow \mathbb{R}$  è **continuo** in quanto per il teorema di Weierstrass esiste  $\|f\|_\infty$  ed è

$$\begin{aligned} |I_n(f)| &= \left| \sum_{j=1}^{\eta_n} w_j f_j \right| \leq \sum_{j=1}^{\eta_n} |w_j| |f_j| \\ &\leq \left( \sum_{j=1}^{\eta_n} |w_j| \right) \cdot \max_j |f_j| \leq \left( \sum_{j=1}^{\eta_n} |w_j| \right) \cdot \|f\|_\infty \quad (19) \end{aligned}$$

# Alcune norme di operatori

In particolare, scegliendo opportunamente  $f$  si prova che **la norma dell'operatore di quadratura**

$$\|I_n\|_\infty = \max_{f \in C([a,b]), f \neq 0} \frac{|I_n(f)|}{\|f\|_\infty}$$

**coincide con**  $\sum_{j=1}^{n_n} |w_j|$ .

Inoltre se  $(a, b)$  è limitato

$$\|I\|_\infty = \max_{f \in C([a,b]), f \neq 0} \frac{|I(f)|}{\|f\|_\infty} = \int_a^b w(x) dx = \|w\|_\infty$$

in quanto

$$\begin{aligned} |I(f)| &= \left| \int_a^b f(x)w(x)dx \right| \leq \int_a^b |f(x)|w(x)dx \\ &\leq \int_a^b w(x)dx \cdot \|f\|_\infty = \|w\|_\infty \|f\|_\infty \end{aligned} \quad (20)$$

e

$$|I(1)| = \|w\|_\infty.$$

# Teorema di Stieltjes: legame tra errore di quadratura e miglior approssimazione polinomiale

Mostriamo il seguente teorema (di Stieltjes), che lega l'errore delle formule di quadratura a quello di miglior approssimazione. Con  $\mathcal{P}_n$  denotiamo i polinomi di grado al più  $n$ .

## Teorema

Sia  $(a, b)$  un intervallo limitato e si supponga  $w : (a, b) \rightarrow \mathbb{R}$  sia una funzione peso e

$$I_n(f) = \sum_{j=1}^{\eta_n} w_j f_j, \quad \text{con } f_j = f(x_j)$$

una formula di quadratura avente grado di precisione almeno  $n$ .

Allora posto  $\mathcal{E}_n(f) = I(f) - I_n(f)$  si ha

$$|\mathcal{E}_n(f)| \leq \left( \|w\|_1 + \sum_{j=1}^{\eta_n} |w_j| \right) \cdot \min_{q_n \in \mathcal{P}_n} \|f - q_n\|_\infty. \quad (21)$$

# Teorema di Stieltjes: legame tra errore di quadratura e miglior approssimazione polinomiale

## Dimostrazione.

*Se  $q \in \mathcal{P}_n$  è un polinomio arbitrario di grado  $n$  ed  $I(q) = I_n(q)$  avendo la formula di quadratura grado di precisione almeno  $n$ .*

*Ricordiamo inoltre che gli operatori  $I$  ed  $I_n$  sono lineari e quindi*

$$I_n(f - q) = I_n(f) - I_n(q),$$

$$I(f - q) = I(f) - I(q).$$

# Teorema di Stieltjes: legame tra errore di quadratura e miglior approssimazione polinomiale

## Dimostrazione.

Per quanto visto

$|I_n(f)| \leq \|I_n\|_\infty \|f\|_\infty$ ,  $|I(f)| \leq \|f\|_\infty \|w\|_1$   
poichè per ogni polinomio di grado  $n$  si ha che  $I_n(q) = I(q)$ ,

$$\begin{aligned} |I(f) - I_n(f)| &= |I(f) - I_n(q) + I_n(q) - I_n(f)| \\ &\leq |I(f) - I_n(q)| + |I_n(q) - I_n(f)| \\ &\leq |I(f) - I(q)| + |I_n(q - f)| \\ &\leq |I(f - q)| + |I_n(f - q)| \\ &\leq \|w\|_1 \|f - q\|_\infty + \|I_n\|_\infty \|f - q\|_\infty \\ &= (\|w\|_1 + \|I_n\|_\infty) \cdot \|f - q\|_\infty \\ &= \left( \|w\|_1 + \sum_{j=1}^{\eta_n} |w_j| \right) \cdot \|f - q\|_\infty \quad (22) \end{aligned}$$

# Teorema di Stieltjes: legame tra errore di quadratura e miglior approssimazione polinomiale

L'interesse di questo teorema è il legame col polinomio di miglior approssimazione. Risulta importante osservare che in (21) contribuiscono i prodotti di due termini.

1. Il primo è dovuto alla funzione peso e alla stabilità della formula di quadratura.
2. Il secondo è dato esclusivamente dalla miglior approssimazione di  $f$  (e non  $fw$ ).

# Teorema di Stieltjes: legame tra errore di quadratura e miglior approssimazione polinomiale

Quindi se  $w$  è una funzione peso con  $fw$  **non regolare** ma  $f$  **regolare** allora l'utilizzo di formule gaussiane rispetto alla funzione peso  $w$ , come anticipato prima, offre risultati *potenzialmente migliori*, come suggerito dai teoremi di Jackson sulla miglior approssimante polinomiale di una funzione  $f$ , che forniscono stime di

$$\min_{q_n \in \mathcal{P}_n} \|f - q_n\|_\infty$$

con  $f \in C([a, b])$  (dotando  $C([a, b])$  della norma infinito).

# Teorema di Stieltjes: legame tra errore di quadratura e miglior approssimazione polinomiale

Quale esempio consideriamo una formula a pesi positivi e grado di precisione  $n \geq 0$ . Necessariamente  $\|I_n\|_\infty = \|w\|_1 = \int_a^b w(x)dx$ , in quanto la formula integra esattamente la costante 1 ed è

$$|I(f) - I_n(f)| \leq 2\|w\|_1 \min_{q_n \in \mathcal{P}_n} \|f - q_n\|_\infty.$$

Quindi, se usiamo la funzione peso di Legendre  $w \equiv 1$  nell'intervallo  $(-1, 1)$  si ha che

$$|I(f) - I_n(f)| \leq 4 \cdot \min_{q_n \in \mathcal{P}_n} \|f - q_n\|_\infty.$$

# Teorema di Stieltjes: legame tra errore di quadratura e miglior approssimazione polinomiale

**Esercizio.** Si calcoli l'integrale

$$\int_{-1}^1 \exp(x) \sqrt{1-x^2} dx$$

con la formula di Gauss-Legendre e una formula di Gauss-Jacobi con esponenti  $\alpha = 1/2$  e  $\beta = 0$ . Quale delle due sarà da usare e perchè ?

# Teorema di Polya-Steklov, sulla convergenza delle formule di quadratura

Consideriamo una formula di quadratura del tipo

$$I_w(f) := \int_a^b f(x)w(x)dx \approx I_n(f) := \sum_{i=0}^{\eta_n} w_{i,n}f(x_{i,n}) \quad (23)$$

con al solito  $w$  una funzione peso definita nell'intervallo  $(a, b)$ . Se tale intervallo è limitato ed  $f$  continua in  $[a, b]$  allora per quanto visto in precedenza si ha  $fw \in L^1(a, b)$ . Sia

$$\mathcal{E}_n(f) := \int_a^b f(x)w(x)dx - \sum_{i=0}^{\eta_n} w_{i,n}f(x_{i,n}).$$

Risulta importante sottolineare che  $n$  non è il grado di precisione della formula, bensì l'indice che contraddistingue l' $n$ -simo elemento della sequenza di formule  $\{I_n\}_n = 1, \dots$ .

# Teorema di Polya-Steklov, sulla convergenza delle formule di quadratura

Dimostriamo ora il teorema di Polya-Steklov [3, p.202].

## Teorema

*Sia un intervallo compatto  $[a, b]$ , e  $Q_n(f) = \sum_{i=0}^{\eta_n} w_{i,n} f(x_{i,n})$  una sequenza di formule di quadratura tale che  $I_w(f) \approx Q_n(f)$ .*

*Condizione necessaria e sufficiente affinché per ogni  $f \in C([a, b])$*

$$\lim_{n \rightarrow +\infty} \mathcal{E}_n(f) = 0$$

*con  $\mathcal{E}_n(f) := I_w(f) - Q_n(f)$  è che*

1. *esista  $M \in \mathbb{R}$  indipendente da  $n$  per cui si abbia*

$$\sum_{i=1}^{\eta_n} |w_{i,n}| \leq M;$$

2. *per ogni  $k \in \mathbb{N}$  si abbia*

$$\lim_{k \rightarrow +\infty} \mathcal{E}_n(x^k) = 0.$$

# Teorema di Polya-Steklov, sulla convergenza delle formule di quadratura

Dimostrazione.

⇐ *Supponiamo che*

1. *esista  $M \in \mathbb{R}$  tale che per ogni  $n$  si abbia*

$$\sum_{i=1}^{\eta_n} |w_{i,n}| \leq M;$$

2. *per ogni  $k \in \mathbb{N}$  si abbia*

$$\lim_{k \rightarrow +\infty} E_n(x^k) = 0.$$

*Per un teorema di densità dovuto a Weierstrass, per ogni  $\tau_1 > 0$  esiste un polinomio  $p$  tale che  $\|f - p\|_\infty \leq \tau_1$ .*

# Teorema di Polya-Steklov, sulla convergenza delle formule di quadratura

*Fissato  $n$ , ricordato che per la definizione di norma degli operatori, si ha che*

$$\|I\|_{\infty} = \sup_{g \in C([a,b]), g \neq 0} \frac{|I(g)|}{\|g\|_{\infty}}$$

*implica, dato che  $\|I\|_{\infty} = \|w\|_1$ , per ogni  $g \in C([a, b])$*

$$|I(g)| \leq \|I\|_{\infty} \|g\|_{\infty} = \|w\|_1 \|g\|_{\infty} \quad (24)$$

# Teorema di Polya-Steklov, sulla convergenza delle formule di quadratura

*Similmente*

$$\|I_n\|_\infty = \sup_{g \in C([a,b]), g \neq 0} \frac{|I_n(g)|}{\|g\|_\infty}$$

*implica, ricordando che*  $\|I_n\|_\infty = \sum_{i=1}^{\eta_n} |w_{i,n}|$ ,

$$|I_n(g)| \leq \|I_n\|_\infty \|g\|_\infty = \sum_{i=1}^{\eta_n} |w_{i,n}| \|g\|_\infty. \quad (25)$$

# Teorema di Polya-Steklov, sulla convergenza delle formule di quadratura

Quindi, per  $g = f - p$  in (24) e (25)

$$\begin{aligned} |\mathcal{E}_n(f - p)| &= |I(f - p) - I_n(f - p)| \leq |I(f - p)| + |I_n(f - p)| \\ &\leq \|w\|_1 \|f - p\|_\infty + \sum_{i=1}^{\eta_n} |w_{i,n}| \|f - p\|_\infty \\ &= \left( \|w\|_1 + \sum_{i=1}^{\eta_n} |w_{i,n}| \right) \cdot \|f - p\|_\infty \\ &\leq (\|w\|_1 + M) \cdot \tau_1. \end{aligned} \tag{26}$$

Di conseguenza  $|\mathcal{E}_n(f - p)| \leq (\|w\|_1 + M) \cdot \tau_1$ , per ogni  $n \in \mathbb{N}$ . Si osservi che il secondo membro della precedente disuguaglianza non dipende da  $n$ .

# Teorema di Polya-Steklov, sulla convergenza delle formule di quadratura

Poichè  $\lim_{n \rightarrow \infty} \mathcal{E}_n(p) = 0$ , per ogni  $\tau_2 > 0$  esiste  $\tilde{N}(\tau_2) \in \mathbb{N}$  tale che se  $n \geq \tilde{N}(\tau_2)$  allora  $|\mathcal{E}_n(p)| \leq \tau_2$ .

Di conseguenza fissato  $\epsilon > 0$  e posto

$$\blacktriangleright \tau_1 = \epsilon / (2 \cdot (\|w\|_1 + M)),$$

$$\blacktriangleright \tau_2 = \epsilon / 2,$$

abbiamo che esiste  $N(\epsilon) = \tilde{N}(\epsilon/2)$  tale che se  $n \geq N(\epsilon)$  allora

$$|\mathcal{E}_n(f - p)| \leq (\|w\|_1 + M) \cdot \tau_1 = \frac{(\|w\|_1 + M) \cdot \epsilon}{2 \cdot (\|w\|_1 + M)} = \epsilon/2$$

e

$$|\mathcal{E}_n(p)| \leq \tau_2 = \epsilon/2.$$

# Teorema di Polya-Steklov, sulla convergenza delle formule di quadratura

Quindi, dalla linearità di  $\mathcal{E}_n$ , per  $n \geq N(\epsilon) = \tilde{N}(\tau_2)$

$$\begin{aligned} |\mathcal{E}_n(f)| &\leq |\mathcal{E}_n(f) - \mathcal{E}_n(p)| + |\mathcal{E}_n(p)| = |\mathcal{E}_n(f - p)| + |\mathcal{E}_n(p)| \\ &\leq \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon \end{aligned} \quad (27)$$

che permette di concludere dalla definizione di limite che  $\lim_n \mathcal{E}_n(f) = 0$ .

# Teorema di Polya-Steklov, sulla convergenza delle formule di quadratura

*⇒ Mostriamo il primo punto dell'asserto. Supponiamo che per ogni  $f \in C([a, b])$  sia  $\lim_n \mathcal{E}_n(f) = 0$ . Per definizione di  $\mathcal{E}_n(f)$  abbiamo  $I_n(f) = I(f) - \mathcal{E}_n(f)$  e quindi per la disuguaglianza triangolare*

$$|I_n(f)| \leq |I(f)| + |\mathcal{E}_n(f)| \leq \|w\|_1 \|f\|_\infty + |\mathcal{E}_n(f)|.$$

*Poichè  $\lim_n \mathcal{E}_n(f) = 0$  necessariamente  $\lim_n |\mathcal{E}_n(f)| = 0$  e quindi, dalla definizione di limite, segue facilmente che esiste  $M(f) \in \mathbb{R}$  (indipendente da  $n$ , ma dipendente da  $f$ ) tale che  $I_n(f) \leq M(f) < \infty$ .*

# Teorema di Polya-Steklov, sulla convergenza delle formule di quadratura

*Il teorema di uniforme limitatezza (talvolta citato come di Banach-Steinhaus) [2, p.58] stabilisce che*

- ▶ *se  $L_n$  è una sequenza di operatori lineari limitati da uno spazio di Banach  $V$  a uno spazio di Banach  $W$ ,*
- ▶ *per ogni  $v \in V$  la sequenza  $\{L_n(v)\}_n$  è limitata,*

*allora*

$$\sup_n \|L_n\| < +\infty.$$

# Teorema di Polya-Steklov, sulla convergenza delle formule di quadratura

*Nel nostro caso*

- ▶  $V \equiv (C([a, b]), \|\cdot\|_\infty)$ ,  $W \equiv \mathbb{R}$  sono spazi di Banach,
- ▶ posto  $L_n \equiv I_n$ , operatore lineare limitato con norma  $\|I_n\|_\infty = \sum_{i=0}^{\eta_n} |w_{i,n}|$ , se  $f \in C([a, b])$  abbiamo che la sequenza  $\{I_n(f)\}_n$  è limitata in quanto esiste  $M(f) < \infty$  indipendente da  $n$  tale che  $I_n(f) \leq M(f) < \infty$ .

# Teorema di Polya-Steklov, sulla convergenza delle formule di quadratura

*Conseguentemente per il teorema di Banach-Steinhaus, ricordando che*

$$\|I_n\|_\infty = \sum_{i=0}^{\eta_n} |w_{i,n}|$$

*si ha*

$$\sup_n \left( \sum_{i=0}^{\eta_n} |w_{i,n}| \right) = \sup_n \|I_n\|_\infty < +\infty.$$

*e quindi esiste  $M$  finito tale che*

$$\sum_{i=0}^{\eta_n} |w_{i,n}| \leq M < +\infty, \quad \forall n \in \mathbb{N}.$$

# Alcune conseguenze e osservazioni sul Teorema di Polya-Steklov

- ▶ L'intervallo  $[a, b]$  è limitato per cui il teorema di Polya **non è applicabile per funzioni peso quali Gauss-Laguerre e Gauss-Hermite**.
- ▶ Si osservi che in generale le formule di errore introdotte nei capitoli precedenti, implicavano la convergenza in caso l'integranda  $f$  fosse sufficientemente regolare. Nel teorema di Polya-Steklov si chiede **esclusivamente che  $f \in C([a, b])$** , senza però offrire stime dell'errore compiuto.

# Alcune conseguenze e osservazioni sul Teorema di Polya-Steklov

## Teorema

*Consideriamo una formula su un dominio limitato, con i pesi  $w_{i,n}$  positivi, e g.d.p. almeno 0. Essa è convergente per ogni  $f \in C([a, b])$  se e solo se è convergente per ogni polinomio  $p$ .*

## Dimostrazione.

Infatti se è convergente per polinomi in particolare lo è per  $f(x) = 1$ , cioè  $\lim_n E_n(1) = 0$ . Visto che

$$E_n(1) = \int_a^b w(x) dx - \sum_{k=1}^{\eta_n} w_k \rightarrow 0$$

necessariamente  $\sum_{k=1}^{\eta_n} w_k \leq M < +\infty$ .  $\square$

# Alcune conseguenze e osservazioni sul Teorema di Polya-Steklov

## Teorema

*Una formula gaussiana su un dominio limitato è convergente.*

## Dimostrazione.

Sia una **formula di Gauss**, su un dominio limitato, con  $n$  nodi  $\{w_{i,n}\}_{i=1,\dots,n}$  positivi. Per quanto detto, è convergente per ogni  $f \in C([a, b])$  se e solo se è convergente per ogni polinomio  $p$ . Ma ciò è verificato banalmente in quanto essendo il grado di precisione almeno  $2n - 1$ , fissato  $k$ , per  $n \geq \text{ceil}((k + 1)/2)$  si ha  $E_n(x^k) = 0$ .

Quindi, essendo tutti gli zeri contenuti in  $(a, b)$ , possiamo applicare il teorema di Polya-Steklov e dedurre che al crescere del numero di punti  $n$  della formula gaussiana si ha che

$$\lim_{n \rightarrow +\infty} E_n(f) = 0$$

# Alcune conseguenze e osservazioni sul Teorema di Polya-Steklov

## Teorema

*Una sequenza di formule composte, basate su regole a pesi positivi e g.d.p.  $n \geq 0$ , risulta convergente qualora l'ampiezza delle suddivisioni tenda a 0.*

## Dimostrazione.

*Si consideri la suddivisione  $\Delta_m = \{\tau_i\}_{i=0,\dots,m}$  dell'intervallo  $(a, b)$  con*

$$\tau_i < \tau_{i+1}, \quad \tau_0 = a, \quad \tau_m = b$$

*nodi della formula in questione.*

*Visto che la formula composta ha pesi positivi, risulta convergente per ogni  $f \in C([a, b])$  se e solo se è convergente per ogni polinomio  $p$ .*

# Alcune conseguenze e osservazioni sul Teorema di Polya-Steklov

*Mostriamo che è convergente per qualsiasi polinomio  $p$ .*

*Osserviamo che tale formula composta integra esattamente ogni funzione polinomiale a tratti di grado  $n$  su  $\Delta$  e che se  $s_{\Delta_m, n}$  è l'interpolante polinomiale a tratti di grado  $n$  della funzione  $f$  relativamente alla suddivisione  $\Delta_m$  e ai nodi di quadratura,*

$$\begin{aligned}\int_a^b f(x) dx &= \int_a^b (f(x) - s_{\Delta_m, n}) dx + \int_a^b s_{\Delta_m, n} dx \\ &= \int_a^b (f(x) - s_{\Delta_m, n}) dx + \sum_{i=0}^{\eta_n} w_{i, n} f(x_{i, n}) dx \quad (28)\end{aligned}$$

# Alcune conseguenze e osservazioni sul Teorema di Polya-Steklov

da cui se  $f \in C^{(n+1)}([a, b])$

$$\begin{aligned} |\mathcal{E}_n(f)| &= \left| \int_a^b f(x) dx - \sum_{i=1}^{\eta_n} w_{i,n} f(x_{i,n}) \right| \\ &= \left| \int_a^b (f(x) - s_{\Delta_{m,n}}) dx \right| \\ &\leq \left| \int_a^b 1 \cdot dx \right| \cdot \|f - s_{\Delta_{m,n}}\|_{\infty} \\ &\leq (b - a) \cdot \|f - s_{\Delta_{m,n}}\|_{\infty} \\ &\leq (b - a) \cdot \frac{h^{(n+1)} \|f^{(n+1)}\|_{\infty}}{(n + 1)!} \end{aligned}$$

(29)

# Alcune conseguenze e osservazioni sul Teorema di Polya-Steklov

*Siccome un polinomio  $f(x) = x^k$  è infinitamente derivabile,  $f^{(n+1)}$  è continua in  $[a, b]$  e quindi per il teorema di Weierstrass  $\|f^{(n+1)}\|_\infty$  è finito.*

*Di conseguenza, se la successione di formule composte è tale che la massima ampiezza  $h$  della suddivisione tende a 0 allora  $\mathcal{E}_n(x^k) \rightarrow 0$ . Per il teorema di Polya-Steklov si ha così che la formula composta a pesi positivi*

$$\int_a^b f(x)w(x)dx \approx \sum_{i=1}^{\eta_n} w_{i,n}f(x_{i,n}) \quad (30)$$

*è tale che qualsiasi sia la funzione continua  $f$ ,  $\mathcal{E}_n(f) \rightarrow 0$  quando la massima ampiezza dei subintervalli tende a 0.*

# Alcune conseguenze e osservazioni sul Teorema di Polya-Steklov (facoltativo)

**[Facoltativo]** Una formula  $I_n(f) \approx \int_a^b f(x)w(x)dx := I(f)$  a pesi positivi convergente sui polinomi e avente grado di precisione almeno 0 risulta convergente sulle funzioni continue a tratti in  $[a, b]$ . Se  $f$  è tale funzione, si dimostra che per ogni  $\epsilon > 0$  esistono due funzioni  $f_1, f_2 \in C([a, b])$  tali che  $f_1 \leq f \leq f_2$  e

$$\|f_1 - f_2\|_\infty \leq \epsilon.$$

Osserviamo che

- ▶ Poichè la formula è a pesi positivi,  $f_1 \leq f \leq f_2$  implica che  $I_n(f_1) \leq I_n(f) \leq I_n(f_2)$ .
- ▶ Essendo  $f_1 \leq f \leq f_2$ ,

$$\int_a^b f_1(x)w(x)dx \leq \int_a^b f(x)w(x)dx \leq \int_a^b f_2(x)w(x)dx.$$

- ▶ Per il teorema di Polya-Steklov abbiamo inoltre che essendo  $f_1, f_2 \in C([a, b])$  allora  $\lim_n \mathcal{E}_n(f_1) = 0$  e  $\lim_n \mathcal{E}_n(f_2) = 0$ .

# Alcune conseguenze e osservazioni sul Teorema di Polya-Steklov (facoltativo)

Ora notiamo che

$$I(f_1) - I_n(f_2) \leq I(f) - I_n(f) \leq I(f_2) - I_n(f_1). \quad (31)$$

Per la linearità degli operatori  $I, I_n$

$$\mathcal{E}_n(f_1) + I_n(f_1 - f_2) = (I(f_1) - I_n(f_1)) + I_n(f_1 - f_2) = I(f_1) - I_n(f_2) \quad (32)$$

$$\mathcal{E}_n(f_2) + I_n(f_2 - f_1) = (I(f_2) - I_n(f_2)) + I_n(f_2 - f_1) = I(f_2) - I_n(f_1) \quad (33)$$

Inoltre

$$I_n(f_2 - f_1) = \sum_{k=1}^{\eta_n} w_k (f_2(x_k) - f_1(x_k)) \leq \sum_{k=1}^{\eta_n} w_k \|f_2 - f_1\|_{\infty} \quad (34)$$

$$I_n(f_1 - f_2) = -I_n(f_2 - f_1) \geq -\sum_{k=1}^{\eta_n} w_k \|f_2 - f_1\|_{\infty} \quad (35)$$

# Alcune conseguenze e osservazioni sul Teorema di Polya-Steklov (facoltativo)

Quindi da (31), in virtù di (33), (34)

$$\mathcal{E}_n(f) \leq I(f_2) - I_n(f_1) = \mathcal{E}_n(f_2) + I_n(f_2 - f_1) \leq \mathcal{E}_n(f_2) + \sum_{k=1}^{\eta_n} w_k \|f_2 - f_1\|_\infty$$

mentre da (31), in virtù di (32), (35)

$$\mathcal{E}_n(f_1) - \sum_{k=1}^{\eta_n} w_k \|f_2 - f_1\|_\infty \leq \mathcal{E}_n(f_1) + I_n(f_1 - f_2) = I(f_1) - I_n(f_2) \leq \mathcal{E}_n(f)$$

cioè

$$\mathcal{E}_n(f_1) - \sum_{k=1}^{\eta_n} w_k \|f_2 - f_1\|_\infty \leq \mathcal{E}_n(f) \leq \mathcal{E}_n(f_2) + \sum_{k=1}^{\eta_n} w_k \|f_2 - f_1\|_\infty$$

Dal fatto che  $\sum_{k=1}^{\eta_n} w_k = \int_a^b w(x) dx < +\infty$ ,  
 $\lim_n \mathcal{E}_n(f_1) = \lim_n \mathcal{E}_n(f_2) = 0$  e  $\|f_2 - f_1\|_\infty \leq \epsilon$ , dall'arbitrarietà di  $\epsilon$   
deduciamo che  $\mathcal{E}_n(f) \rightarrow 0$ .

Eccetto che in pochi casi (come ad esempio per la funzione peso di Chebyshev), non esistono formule esplicite per l'individuazione di tali nodi e pesi.

Una volta venivano tabulati, oggi si consiglia di applicare del software che si può trovare ad esempio nella pagina di Walter Gautschi

```
http://www.cs.purdue.edu/people/wxg
```

Fissato un peso, ad esempio quello di Jacobi, si cercano per prima cosa i *coefficienti di ricorrenza*, che possono essere calcolati con l' m-file *r\_jacobi.m* nel sito di W. Gautschi.

La sintassi è la seguente

```
ab=r_jacobi(n,a,b)
```

Come si vede dai commenti al file, genera i primi  $n$  coeff. di ricorrenza dei polinomi ortogonali monici relativamente alla funzione peso di Jacobi

$$w(x) = (1 - x)^a (1 + x)^b.$$

Quindi

- ▶  $a, b$  corrispondono rispettivamente all' $\alpha$  e  $\beta$  delle formule di Jacobi (e non agli estremi di integrazione!).
- ▶ I coefficienti di ricorrenza sono immagazzinati nella variabile  $ab$ .

Nota.

*Ricordiamo che se  $\{\phi_n(x)\}$  è una famiglia di polinomi ortogonali su  $[a, b]$ , rispetto ad una funzione peso  $w(x)$ , allora per  $n \geq 1$*

$$\phi_{n+1}(x) = \alpha_n(x - \beta_n)\phi_n(x) - \gamma_n\phi_{n-1}(x)$$

*ove, detto  $A_n$  il coefficiente di grado  $n$  in  $\phi_n$ , si ha  $\alpha_n = A_{n+1}/A_n$ ,*

$$\beta_n = \frac{(x\phi_n, \phi_n)}{\|\phi_n\|^2}, \quad \gamma_n = \frac{(\phi_n, x\phi_{n-1})}{\|\phi_n\|^2},$$

*con  $(f, g) = \int_a^b f(x)g(x) dx$ .*

*Nel caso di `r_jacobi` si usa la formula ricorsiva dei polinomi ortogonali (monici!) [7, p.216]*

$$\begin{aligned}\phi_{n+1}(x) &= (x - \bar{\alpha}_n)\phi_n(x) - \bar{\beta}_n\phi_{n-1}(x), \quad k = 1, 2, \dots \\ \phi_{-1}(t) &= 0, \quad \phi_0(t) = 1.\end{aligned}\tag{36}$$

*Osserviamo che,*

- ▶ *essendo i polinomi ortogonali definiti a meno di una costante moltiplicativa non nulla (se  $(\phi_k, \phi_j) = 0$  per ogni  $j < k$  pure  $(\tau\phi_k, \phi_j) = 0$  per ogni  $j < k$ , per ogni  $\tau \neq 0$ ), si può richiedere che la famiglia triangolare di polinomi ortogonali sia di polinomi monici (cioè con coefficienti di grado massimo uguali a 1).*
- ▶ *Il vettore `ab` ha quale prima colonna il vettore dei coefficienti  $\{\bar{\alpha}_k\}$  e quale seconda colonna il vettore dei coefficienti  $\{\bar{\beta}_k\}$ .*

# Formule gaussiane in Matlab

A questo punto si chiama la funzione *gauss.m* (reperibile nuovamente presso lo stesso sito di W. Gautschi)

```
xw=gauss(N,ab)
```

definita da

```
function xw=gauss(N,ab)
N0=size(ab,1); if N0<N, error('input array ab too short
    '), end
J=zeros(N);
for n=1:N, J(n,n)=ab(n,1); end
for n=2:N
    J(n,n-1)=sqrt(ab(n,2));
    J(n-1,n)=J(n,n-1);
end
[V,D]=eig(J);
[D,I]=sort(diag(D));
V=V(:,I);
xw=[D ab(1,2)*V(1,:)'.^2];
```

- ▶ Tale routine, che per un certo  $N$ , uguale al massimo al numero di righe della matrice di coefficienti di ricorrenza  $ab$ , fornisce  **$N$  nodi  $x$  e i corrispondenti pesi  $w$**  immagazzinati in una matrice  $xw$  che ha quale prima colonna  $x$  e quale seconda colonna  $w$ .

Osserviamo che in tale codice i polinomi ortogonali sono monici e quindi compatibili con quelli *forniti* da `r_jacobi`.

- ▶ La descrizione di perchè tale software fornisca il risultato desiderato è complicata ma può essere trovata nella monografia di W. Gautschi sui polinomi ortogonali, per Acta Numerica.

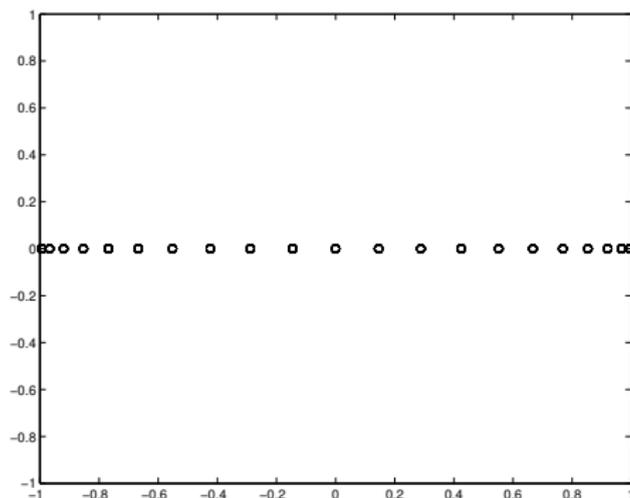
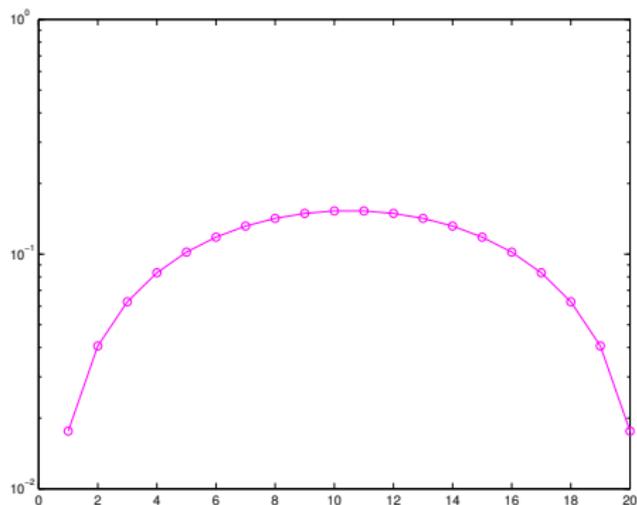


Figura : Grafico che illustra la distribuzione dei 20 nodi di Gauss-Legendre nell'intervallo  $[-1, 1]$



**Figura :** Grafico che illustra la distribuzione dei 20 pesi di Gauss-Legendre nell'intervallo  $[1, 1]$

# Una demo di Gauss-Jacobi

Salviamo nel file `integrazione_gauss_jacobi.m`

```
function [I,x,w]=integrazione_gauss_jacobi(N,alpha,beta
,a,b,f)
% INPUT:
% N: NUMERO NODI.
% alpha,beta:  $w(x)=((1-x)^\alpha)*((1+x)^\beta)$ 
% a,b: SE alpha=0, beta=0, SONO ESTREMI DI INTEGRAZIONE
%       ,
%       ALTRIMENTI a=-1, b=1.
% f: INTEGRANDA (RISPETTO w).
% OUTPUT:
% I: APPROSSIMAZIONE DI  $\int_a^b f(x) w(x) dx$ 
% x,w: NODI/PESI UTILIZZATI.
```

# Una demo di Gauss-Jacobi

```
ab=r_jacobi(N,alpha,beta); %TERM. RICORSIVI
xw=gauss(N,ab); %NODI/PESI IN MATRICE .
x=xw(:,1);w=xw(:,2);%NODI/PESI GAUSS JAC.[-1,1].
if (alpha == 0) & (beta == 0) & ((a~-1)|(b~=1))
    x=((a+b)/2)+((b-a)/2)*x;%NODI GAUSS-LEG.[a,b].
    w=((b-a)/2)*w;%PESI GAUSS-LEG.[a,b].
end
fx=feval(f,x);%VALUTAZIONE FUNZIONE.
I=w'*fx;%VALORE INTEGRALE.
```

# Una demo di Gauss-Jacobi

e nel file *f.m* delle funzioni su cui effettueremo dei test. Un esempio è

```
function fx=f(x)
fx=x.^20;
% ALCUNE FUNZIONI TRATTE DA
% "IS GAUSS QUADRATURE BETTER THAN CLENSHAW-CURTIS?"
% DI L.N. TREFETHEN.
% fx=exp(x);
% fx=exp(-x.^2);
% fx=1./(1+16*(x.^2));
% fx=exp(-x.^(-2));
% fx=abs(x); fx=fx.^3;
% fx=x.^(0.5);
% fx=exp(x).*(sqrt(1-x));
```

# Una demo di Gauss-Jacobi

## Le funzioni test [10]

1.  $f(x) = x^{20}$ ;
2.  $f(x) = \exp(x)$ ;
3.  $f(x) = \exp(-x^2)$ ;
4.  $f(x) = 1./(1 + 16 * x^2)$ ;
5.  $f(x) = \exp(-x^{-2})$ ;
6.  $f(x) = |x|^3$ ;
7.  $f(x) = x^{0.5}$ ;
8.  $f(x) = \exp(x) * (\sqrt{1-x})$ ;

con  $a = -1$ ,  $b = 1$  possono essere valutate a scelta cancellando % riponendolo altrove.

Scaricando dalla pagina web del corso

```
integrazione_gauss_jacobi.m, f.m
```

e da quella di W. Gautschi

```
r_jacobi.m, gauss.m
```

possiamo fare quindi alcuni esperimenti.

# Una demo di Gauss-Jacobi: scalatura

L'idea della scavatatura consiste nel modificare la formula di quadratura relativa ad un intervallo  $[a, b]$  di riferimento così da poterla utilizzare in un intervallo generico  $[\alpha, \beta]$ , cambiando esclusivamente nodi e pesi.

Si osservi che per ottenere i nodi e pesi di Gauss-Legendre in  $(a, b)$  da quelli di Gauss-Legendre in  $(-1, 1)$  abbiamo effettuato uno **scalatura**.

- ▶ se  $x_{i,[-1,1]}$  sono i nodi di Gauss-Legendre in  $[-1, 1]$  allora i nodi di Gauss-Legendre  $x_{i,[a,b]}$  in  $[a, b]$  sono

$$x_{i,[a,b]} = ((a + b)/2) + ((b - a)/2) \cdot x_{i,[-1,1]}$$

- ▶ se  $w_{i,[-1,1]}$  sono i pesi di Gauss-Jacobi in  $[-1, 1]$  allora i pesi di Gauss-Jacobi  $w_{i,[a,b]}$  in  $[a, b]$  sono

$$w_{i,[a,b]} = ((b - a)/2) \cdot w_{i,[-1,1]}$$

# Formule gaussiane in Matlab: scalatura

Specialmente per le formule con peso  $w(x) \equiv 1$  si ha spesso da effettuare l'integrale  $\int_a^b f(x)dx$  con  $(a, b)$  diverso da  $(-1, 1)$ .

In tal caso si osserva che se  $\gamma(t) = (a(1-t)/2) + (b(1+t)/2)$ , per  $\{t_k\}$  e  $\{w_k\}$  nodi e pesi della formula di Gauss-Legendre,

$$\int_a^b f(x)dx = \int_{-1}^1 \frac{b-a}{2} f(\gamma(t))dt \approx \sum_{k=1}^n \frac{b-a}{2} w_k f(\gamma(t_k))$$

e quindi

$$\int_a^b f(x)dx \approx \sum_{k=1}^n w_k^* f(x_k^*)$$

con  $w_k^* = \frac{b-a}{2} w_k$ ,  $x_k = \gamma(t_k) = (a(1-t_k)/2) + (b(1+t_k)/2)$ .

Questo processo si chiama *scalatura* dei nodi e dei pesi.

# Esempio 1

Se ora testiamo il primo esempio, cioè il calcolo di

$$\int_{-1}^1 x^{20} dx \approx 0.09523809523809523300$$

otteniamo

```
>> format long;
>> N=11; ajac=0; bjac=0; a=-1; b=1;
>> [I_jac,x_jac,w_jac]=integrazione_gauss_jacobi(N,ajac
    ,bjac,a,b,@f);
>> I_jac
I_jac =
    0.09523809523810
>> length(x_jac)
ans =
    11
>> fprintf('\n \t [GAUSS-LEGENDRE]: \t%15.20f',I_jac);
    [GAUSS-LEGENDRE]:  0.09523809523809564900
>> 0.09523809523809523300 - 0.09523809523809564900
ans =
-4.163336342344337e-016
```

## Esempio 2

Consideriamo ora l'integrale

$$\int_{-1}^1 \exp(x) \sqrt{1-x} dx = 1.7791436546919097925911790299941. \quad (37)$$

Tale risultato è stato ottenuto usando il comando simbolico di Matlab 6.5 (non funziona in Octave, vedere in alternativa il programma Maxima!!)

```
>> syms x
>> int(' (exp(x)) * ( (1-x)^(0.5) )', -1,1)
ans =
1.7791436546919097925911790299941
```

## Esempio 2

Si capisce che

1. `syms x` rende la variabile `x` di tipo simbolico (e non numerico!);
2. il termine

```
int('exp(x) * ((1-x)^(0.5))', -1, 1)
```

calcola simbolicamente l'integrale

$$\int_{-1}^1 \exp(x) \sqrt{1-x} dx.$$

## Esempio 2

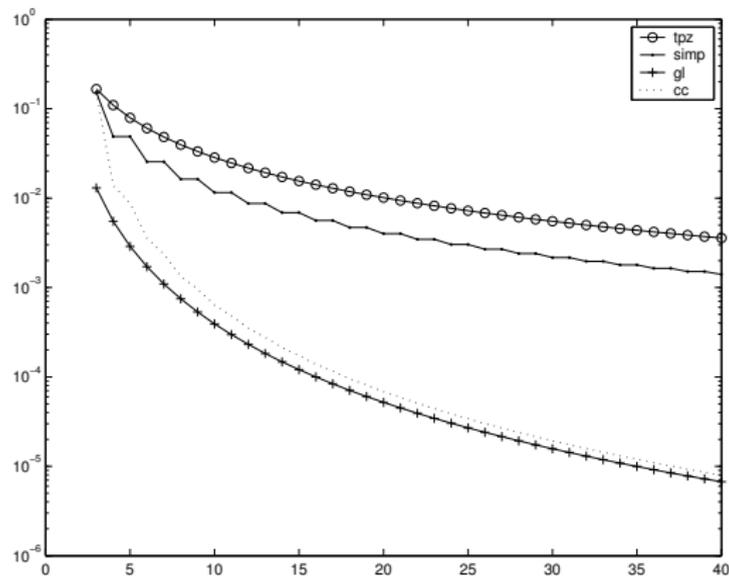


Figura : Grafico che illustra l'errore delle formule di quad. dei trap. composta, Cav.-Simpson comp., Gauss-Legendre e Clenshaw-Curtis relativamente a  $\int_{-1}^1 \exp(x) \sqrt{1-x} dx$ .

E' immediato osservare che  $w(x) = \sqrt{1-x}$  è un peso di Gauss-Jacobi

$$w(t) = (1-t)^\alpha(1+t)^\beta$$

per  $\alpha = 1/2$  e  $\beta = 0$ .

Infatti se  $w(x) = \sqrt{1-x}$ , allora  $g(x) = \exp(x) w(x)$  il che corrisponde a usare Gauss-Jacobi con  $f(x) = \exp(x)$ . Quindi paragoniamo le formule gaussiane (il codice funziona in Matlab 6.1 come pure nella release 2.1.73 di Octave).

## Esempio 2

```
>> a=-1; b=1;
>> [I_jac , x_jac , w_jac]=integrazione_gauss_jacobi
    (10,1/2,0,a,b,@exp);
>> fprintf('\n \t [GAUSS-JACOBI]: %15.20f',I_jac);
    [GAUSS-JACOBI]: 1.77914365469190930000
>> 1.77914365469190979259117902999 -1.779143654691909300
ans = 4.440892098500626e-016
>> length(x_jac)
ans = 10
>> [I_jac , x_jac , w_jac]=...
    integrazione_gauss_jacobi(10,0,0,-1,1,inline('exp(x).*
    sqrt(1-x)'));
>> fprintf('\n \t [GAUSS-LEGENDRE]: %15.20f',I_jac)
    [GAUSS-LEGENDRE]: 1.77984112101478020000
>> 1.77914365469190979259117902999 -1.779841121014780200
ans = -6.9747e-004
>> length(x_jac)
ans = 10
>>
```

## Esempio 2

Entrambe le formule hanno lo stesso numero di nodi (e pesi), come si vede dalla riga di comando

```
>> length(x_jac)
ans =
    10
```

ma offrono risultati diversi, con un errore assoluto di circa  $4.44 \cdot 10^{-16}$  per Gauss-Jacobi con  $a = 1/2$ ,  $b = 0$  e di  $2.52 \cdot 10^{-3}$  per Gauss-Legendre (cioè Gauss-Jacobi con  $a = 0$ ,  $b = 0$ ).

Usando una opportuna formula gaussiana, con  $n = 10, 20, 30$  punti, calcolare

- ▶  $\int_0^{\pi} \exp(x) \cdot \cos(4x) dx = \frac{\exp(\pi)-1}{17};$
- ▶  $\int_0^1 x^{5/2} dx = \frac{2}{7};$
- ▶  $\int_{-\pi}^{\pi} \exp(\cos(x)) dx = 7.95492652101284;$
- ▶  $\int_0^{\pi/4} \exp(\cos(x)) dx = 1.93973485062365;$
- ▶  $\int_0^1 x^{1/2} dx = \frac{2}{3}.$

Descrivere come decresce l'errore in scala semilogaritmica.

**NB:** Non serve vedere quale sia l'andamento del rapporto  $E_n(f)/E_{n+1}(f)$ , dove  $E_n = |I(f) - I_n(f)|$  con  $I(f)$  integrale esatto e  $I_n(f)$  valore fornito dalla formula di quadratura.

## Esercizio (facoltativo)

Si calcolino per  $N = 10, 20$  con la formula composta dei trapezi, di Cav.-Simpson e un'appropriata formula gaussiana

$$\int_{-1}^1 x^{20} dx = 2/21 \approx 0.095238095238096801 \quad (38)$$

$$\int_{-1}^1 e^x dx = e - e^{-1} \approx 2.3504023872876032 \quad (39)$$

$$\int_{-1}^1 e^{-x^2} dx = \operatorname{erf}(1) \cdot \sqrt{\pi} \approx 1.4936482656248538 \quad (40)$$

$$\int_{-1}^1 1/(1 + 16x^2) dx = 1/2 \cdot \operatorname{atan}(4) \approx 0.66290883183401628 \quad (41)$$

$$\int_{-1}^1 e^{-1/x^2} dx \approx 0.17814771178156086 \quad (42)$$

$$\int_{-1}^1 |x|^3 dx = 1/2 \quad (43)$$

$$\int_0^1 \sqrt{x} dx = 2/3 \quad (44)$$

$$\int_{-1}^1 e^x \sqrt{1-x} dx \approx 1.7791436546919095 \quad (45)$$

## Esercizio (facoltativo)

1. Quali delle due formule ha errori relativi inferiori?
2. Quali funzioni risultano più difficili da integrare numericamente?
3. La scelta  $N = 10, 20$  nei singoli codici a quanti nodi corrisponde?
4. Suggerimento: ricordarsi che il peso  $w(x) \equiv 1$  corrisponde ad un'opportuna scelta del peso di Jacobi.

## Esercizio (facoltativo)

1. Quali delle due formule ha errori relativi inferiori?
2. Quali funzioni risultano più difficili da integrare numericamente?
3. La scelta  $N = 10, 20$  nei singoli codici a quanti nodi corrisponde?
4. Suggerimento: ricordarsi che il peso  $w(x) \equiv 1$  corrisponde ad un'opportuna scelta del peso di Jacobi.

## Esercizio (facoltativo)

Calcolare con una opportuna formula gaussiana basata su 11 nodi:

1.

$$\int_{-1}^1 x^{20} \sqrt{1-x^2} dx$$

2.

$$\int_{-1}^1 \exp(x^2) \cdot (1-x)^{0.5} \cdot (1+x)^{1.5} dx$$

Il primo integrale, può essere calcolato *esattamente* con Mathematica, Maple (da Matlab, aiutarsi con Google) o quadgk di Matlab, una toolbox di Matlab specifica per l'integrazione numerica, con un errore assoluto di  $10^{-15}$

```
>> Q = quadgk(@(x)(x.^20).*(1-x.^2).^0.5), -1, 1, 'AbsTol', 10^(-15))
Q =
    0.025160880188796
>>
```

## Esercizio (facoltativo)

In alternativa usando online il Wolfram Integrator, e ricordando che devo integrare in  $(-1, 1)$ , dal teorema fondamentale del calcolo si vede che si tratta di valutare

$$\frac{14549535}{1816657920} \cdot (\arcsin(1) - \arcsin(-1)).$$

Utilizzando la shell di Matlab

```
>> c=14549535/1816657920;
>> s=asin(1)-asin(-1);
>> I=s*c;
>> format long;
>> I
I =
    0.025160880188796
>>
```

## Esercizio (facoltativo)

Il secondo integrale non è integrabile col Wolfram integrator e nemmeno col calcolo simbolico di Matlab.

```
>> syms x
>> int((exp(-x.^2)).*((1-x).^0.5)).*((1+x).^1.5))
    , -1,1)
Warning: Explicit integral could not be found.
> In sym.int at 58
ans =
int(exp(-x^2)*(1-x)^(1/2)*(1+x)^(3/2), x = -1 .. 1)
```

## Esercizio (facoltativo)

Utilizzando la shell di Matlab e `quadgk` con un errore assoluto di  $10^{-15}$

```
>> format long;
>> I=quadgk(@(x)(exp(x.^2)).*((1-x).^0.5)).*((1+x)
    .^(1.5)), -1,1,...
    'AbsTol',10^(-15))
I =
    2.086318843895086
>>
```

# Bibliografia



K. Atkinson, *Introduction to Numerical Analysis*, Wiley, 1989.



K. Atkinson e W. Han, *Theoretical Numerical Analysis*, Springer, 2001.



V. Comincioli, *Analisi Numerica, metodi modelli applicazioni*, Mc Graw-Hill, 1990.



S.D. Conte e C. de Boor, *Elementary Numerical Analysis, 3rd Edition*, Mc Graw-Hill, 1980.



P.J. Davis, *Interpolation and Approximation*, Dover, 1975.



W. Gautschi, personal homepage, <http://www.cs.purdue.edu/people/wxg>.



W. Gautschi, , *Orthogonal polynomials (in Matlab)*, JCAM 178 (2005) 215–234



A. Quarteroni e F. Saleri, *Introduzione al calcolo scientifico*, Springer Verlag, 2006.



A. Suli e D. Mayers, *An Introduction to Numerical Analysis*, Cambridge University Press, 2003.



L.N. Trefethen, *Is Gauss quadrature better than Clenshaw-Curtis?*, SIAM Reviews, (2007).



J. Waldvogel, *Fast Construction of the Fejer and Clenshaw-Curtis Quadrature Rules*, BIT 46 (2006),no. 1,195–202.



Wikipedia, Clenshaw-Curtis quadrature, [http://en.wikipedia.org/wiki/Clenshaw-Curtis\\_quadrature](http://en.wikipedia.org/wiki/Clenshaw-Curtis_quadrature).



Wikipedia, Quadratura gaussiana, [http://it.wikipedia.org/wiki/Quadratura\\_di\\_Gauss](http://it.wikipedia.org/wiki/Quadratura_di_Gauss).



Wikipedia, Regola del trapezio, [http://it.wikipedia.org/wiki/Regola\\_del\\_trapezio](http://it.wikipedia.org/wiki/Regola_del_trapezio).



Wikipedia, Regola di Cavalieri-Simpson, [http://it.wikipedia.org/wiki/Regola\\_di\\_Cavalieri-Simpson](http://it.wikipedia.org/wiki/Regola_di_Cavalieri-Simpson).