

# Serie 6 - Soluzione

## Integrazione Numerica

---

©2021 - Questo testo (compresi i quesiti ed il loro svolgimento) è coperto da diritto d'autore. Non può essere sfruttato a fini commerciali o di pubblicazione editoriale. Non possono essere ricavati lavori derivati. Ogni abuso sarà punito a termine di legge dal titolare del diritto. This text is licensed to the public under the Creative Commons Attribution-NonCommercial-NoDerivs2.5 License (<http://creativecommons.org/licenses/by-nc-nd/2.5/>)

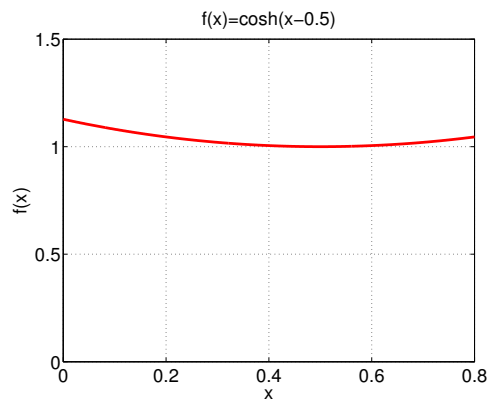
---

### Esercizio 1

1. Disegnare il grafico della funzione  $f(x)$  nell'intervallo  $[0, 0.8]$ .

Tramite i seguenti comandi Matlab<sup>®</sup> si ottiene il grafico in Figura 1:

```
a = 0; b = 0.8;
x_dis = linspace(a,b,1000);
f = @(x) cosh(x-0.5);
y_dis=f(x_dis);
plot(x_dis,y_dis, 'r' )
```



**Figura 1:** Grafico rappresentante la sezione del solido di rotazione nel piano  $(x, y)$ .

La funzione  $f$  è continua e derivabile infinite volte nell'intervallo  $[0, 0.8]$ .

2. Scrivere la funzione `pmedcomp.m` che implementa la formula di quadratura del punto medio composita su intervalli equispaziati. La funzione `pmedcomp.m` prende in input:

- gli estremi di integrazione  $a, b$ ;
- il numero di sottointervalli  $N$  in cui si vuole suddividere il dominio di integrazione;
- la funzione `fun` da integrare (function handle);

e restituisce in uscita il valore approssimato dell'integrale.

```
function I = pmedcomp(a, b, N, f)

% I = pmedcomp(a, b, N, f)
%
% Formula del punto medio composita:
```

```

% Inputs:
%     a,b: estremi di integrazione,
%     N: numero di sottointervalli (N=1 formula di integrazione semplice)
%     f: funzione da integrare definita come inline o anonymous
% Output:
%     I: integrale calcolato

h = ( b - a ) / N;

x = [ a + h / 2 : h : b - h / 2 ]; % vettore dei punti medi

I = h * sum( f(x) );

```

3. *Scrivere la funzione `trapcomp.m` che implementa la formula di quadratura del trapezio composta su intervalli equispaziati.*

```

function I = trapcomp(a, b, N, f)

% I = trapcomp(a, b, N, f)
%
% Formula del punto medio composta:
% Inputs:
% Formula del trapezio composta:
%     a,b: estremi di integrazione,
%     N: numero di sottointervalli (m=1 formula di integrazione semplice)
%     f: funzione da integrare definita come inline o anonymous
% Output:
%     I: integrale calcolato

h = ( b - a ) / N;

x = [ a : h : b ]; % vettore dei nodi di integrazione

y = f( x );

I = h * ( 0.5 * y( 1 ) + sum( y( 2 : N ) ) + 0.5 * y( N + 1 ) );

```

4. *Scrivere la funzione `simpcomp.m` che implementa la formula di quadratura di Simpson composta su intervalli equispaziati.*

```

function I = simpcomp(a, b, N, f)

% I = simpcomp(a, b, N, f)
%
% Formula di Simpson composta:
% Inputs:
%     a,b: estremi di integrazione,
%     N: numero di sottointervalli (N=1 formula di integrazione semplice)
%     f: funzione da integrare definita come inline o anonymous
% Output:
%     I: integrale calcolato

h = ( b - a ) / N;

```

```

x = [ a : h / 2 : b ]; % vettore dei nodi di integrazione

y = f( x );

I = ( h / 6 ) * ( y( 1 ) + ...
                2 * sum( y( 3 : 2 : 2 * N - 1 ) ) + ...
                4 * sum( y( 2 : 2 : 2 * N ) ) + ...
                y( 2 * N + 1 ) );

```

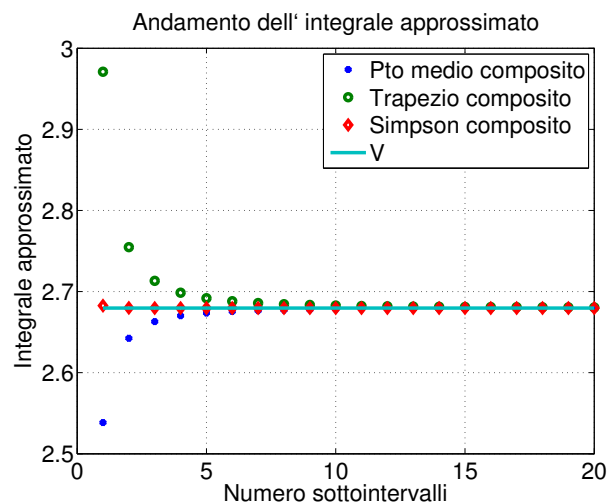
5. Utilizzando le tre funzioni scritte ai punti precedenti, calcolare il valore approssimato  $V_N$  del volume del solido di rotazione per  $N = 1, \dots, 20$  suddivisioni uniformi dell'intervallo  $[0, 0.8]$ . Riportare su un grafico l'andamento del valore dell'integrale approssimato  $V_N$  in funzione di  $N$ .

Con i seguenti comandi si ottiene il grafico in figura 2.

```

f_init = @(x) pi*(cosh(x-0.5)).^2;
N = [1:20];
V_ex = pi*((sinh(1)+sinh(3/5))/4+2/5);
V_PMC = zeros(1,N(end));
V_TC = zeros(1,N(end));
V_SC = zeros(1,N(end));
for i = N
    V_PMC(i) = pmedcomp(a,b,i,f_init);
    V_TC(i) = trapcomp(a,b,i,f_init);
    V_SC(i) = simpcomp(a,b,i,f_init);
end
figure(2);
plot(N,V_PMC,'*',N,V_TC,'o',N,V_SC,'d',N, V_ex * ones(1,20))
grid on
legend('Pto medio composito','Trapezio composito','Simpson composito','V', 1)

```



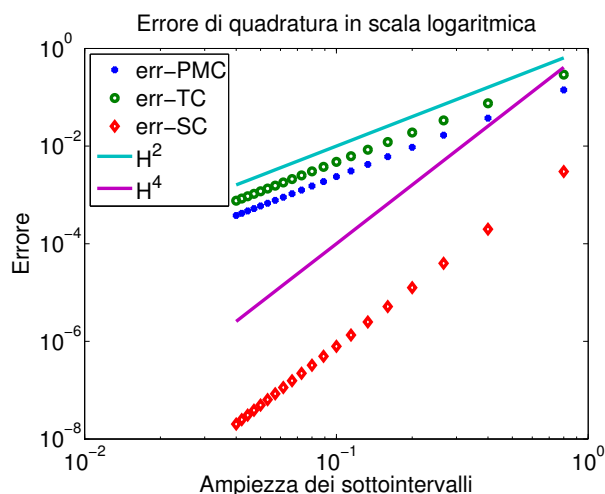
**Figura 2:** Andamento dell'integrale approssimato in funzione del numero di sottointervalli. Punto medio composito, Trapezio composito e Simpson composito.

Si nota che per tutti e tre i metodi si ha convergenza del valore approssimato dell'integrale verso il valore esatto  $V = 2.679664474685$  all'aumentare del numero di sottointervalli. Si può osservare che, in particolare, il metodo di Simpson composto converge più velocemente rispetto agli altri due.

6. *Conoscendo il valore esatto  $V$  del volume, si indichi con  $E_N = |V - V_N|$  il valore assoluto dell'errore nel calcolo dell'integrale. Partendo dai valori di  $V_N$  trovati al punto precedente al variare dell'ampiezza dei sottointervalli, riportare su un grafico, con entrambi gli assi in scala logaritmica, l'andamento dell'errore utilizzando le formule di quadratura implementate. Verificare che ci sia accordo con i risultati teorici per ciascuno dei tre metodi utilizzati.*

Con i seguenti comandi si ottiene il grafico in Figura 3.

```
err_PMC = abs(V_ex - V_PMC);
err_TC  = abs(V_ex - V_TC);
err_SC  = abs(V_ex - V_SC);
H = (b-a)./N;
figure(3);
loglog(H,err_PMC,'*',H,err_TC,'o',H,err_SC,'d',H,H.^2,H,H.^4)
legend('err-PMC','err-TC','err-SC','H^2','H^4', 2)
```



**Figura 3:** Errore di quadratura in scala logaritmica in funzione dell'ampiezza dei sottointervalli. PMC = Punto medio composto, TC = Trapezio composto e SC = Simpson composto.

Essendo  $H$  il vettore delle ampiezze dei sottointervalli utilizzati, si osserva dalla Figura 3 che gli errori per i metodi del punto medio composto e del trapezio composto hanno un andamento simile ad  $H^2$ . Invece per il metodo di Simpson composto l'errore decresce come  $H^4$ . Osserviamo che l'errore calcolato per la formula composta del punto medio risulta circa pari alla metà di quello calcolato per la formula composta del trapezio, come si vede eseguendo le istruzioni:

```
rapp = err_PMC ./ err_TC;
disp(rapp(end-4:end))
```

che restituisce a video gli ultimi 5 valori del rapporto fra i due errori:

```
0.4999    0.4999    0.5000    0.5000    0.5000
```

Si ha dunque perfetto accordo con i risultati teorici.

7. Per ciascuna delle tre formule considerate, fornire una stima del numero minimo  $N$  di sottointervalli per cui  $E_N \leq \text{toll}$ , con  $\text{toll} = 10^{-5}$ , e verificare sperimentalmente tali risultati.

Per ciascuno dei tre metodi il minimo valore intero  $N$  tale per cui risulta  $|V - V_N| \leq \text{toll}$ . Poichè in generale non si conosce il valore dei punti  $\xi$ ,  $\delta$  e  $\eta$ , siamo costretti a maggiore le quantità  $f''(\xi)$ ,  $f''(\delta)$  e  $f^{(4)}(\eta)$  con il massimo del valore assoluto di  $f''(x)$  e di  $f^{(4)}(x)$  sull'intervallo  $[a, b]$ . Indichiamo tali valori rispettivamente con

$$K2 = \max_{x \in [a, b]} |f''(x)| \quad \text{e} \quad K4 = \max_{x \in [a, b]} |f^{(4)}(x)|.$$

Ricordando che nel nostro caso  $a = 0$ ,  $b = 0.8$ ,  $H = (b - a)/N$ ,  $f''(x) = 2\pi(2 \cosh^2(x - \frac{1}{2}) - 1)$  e  $f^{(4)}(x) = 8\pi(2 \cosh^2(x - \frac{1}{2}) - 1)$ , perché venga soddisfatta la tolleranza è sufficiente che valgano le maggiorazioni:

$$\begin{aligned} |I(f) - I_{pm}^c(f)| &\leq \frac{b-a}{24} \left( \frac{b-a}{N} \right)^2 \cdot K2 \leq \text{toll} \\ |I(f) - I_t^c(f)| &\leq \frac{b-a}{12} \left( \frac{b-a}{N} \right)^2 \cdot K2 \leq \text{toll} \\ |I(f) - I_s^c(f)| &\leq \frac{(b-a)}{16 \cdot 180} \left( \frac{b-a}{N} \right)^4 \cdot K4 \leq \text{toll} \end{aligned}$$

Risolvendo rispetto a  $N$  si ha nel caso del punto medio composito:  $N \geq \sqrt{\frac{(b-a)^3 K2}{24 \text{toll}}}$ , nel caso del trapezio composito:  $N \geq \sqrt{\frac{(b-a)^3 K2}{12 \text{toll}}}$ , mentre nel caso di Simpson composito:  $N \geq \left( \frac{(b-a)^5 K4}{2880 \text{toll}} \right)^{1/4}$ . Si sceglierà come  $N$  l'intero più piccolo che soddisfa le disuguaglianze di cui sopra. Implementiamo in Matlab® :

```
toll=1e-5;
d2f_int = @(x) 2*pi*(2*(cosh(x - 0.5)).^2 -1);
d4f_int = @(x) 8*pi*(2*(cosh(x - 0.5)).^2 -1);
K2 = max(abs(d2f_int(x_dis)));
K4 = max(abs(d4f_int(x_dis)));
Npmc = ceil(sqrt( ((b-a)^3*K2)/(24*toll) ));
Ntc = ceil(sqrt( ((b-a)^3*K2)/(12*toll) ));
Nsc = ceil( ( ((b-a)^5*K4)/(2880*toll) )^(1/4) )
```

I valori stampati a schermo sono:

```

Npmc =
    144
Ntc =
    204
Nsc =
     5

```

Verifichiamo qual è l'errore effettivamente commesso utilizzando rispettivamente  $N = N_{pmc}$ ,  $N = N_{tc}$  e  $N = N_{sc}$ :

```

VPM = pmedcomp(a,b,Npmc,f_int);
errPMCvero = abs(V_ex - VPM)
VTC = trapcomp(a,b,Ntc,f_int);
errTCvero = abs(V_ex - VTC)
VSC = simpcomp(a,b,Nsc,f_int);
errSCvero = abs(V_ex - VSC)

```

Gli errori stampati a schermo sono:

```

errPMCvero =
    7.3201e-06
errTCvero =
    7.2948e-06
errSCvero =
    5.1653e-06

```

da cui si evince che la stima ricavata è pessimistica (ovvero sarebbe bastato un numero inferiore di sottointervalli per soddisfare la tolleranza richiesta), cosa giustificabile dal fatto che la derivata seconda e la derivata quarta sono state maggiorate con il loro massimo!

8. *Scrivere la funzione `gausscomp.m` che implementa la formula di quadratura di Gauss(-Legendre) composta a due punti.*

L'implementazione della funzione richiesta è la seguente:

```

function I = gausscomp(a, b, N, f)

% I = gausscomp(a, b, N, f)
%
% Formula di Gauss composta a due nodi:
% Inputs:
%   a,b: estremi di integrazione,
%   N: numero di sottointervalli (m=1) formula di integrazione semplice
%   f: funzione da integrare definita come inline o anonymous
% Output:
%   I: integrale calcolato

h = (b-a) / N;

x = [ a : h : b ]; % vettore degli estremi dei sottointervalli

xm = [ a+h/2 : h : b ]; % vettore dei punti medi dei sottointervalli

```

```

x1 = xm - h/( 2*sqrt(3) ); % vettore contenente una parte dei
                             % nodi di integrazione

y1 = f( x1 );

I  = h/2 * sum(y1); % prima parte dell'integrale

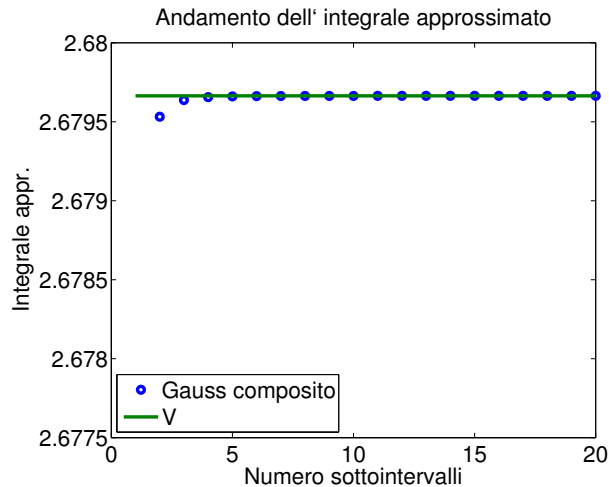
x2 = xm + h/( 2*sqrt(3) ); % vettore contenente la parte
                             % restante dei nodi di integrazione

y2 = f( x2 );

I  = I + h/2 * sum(y2);

```

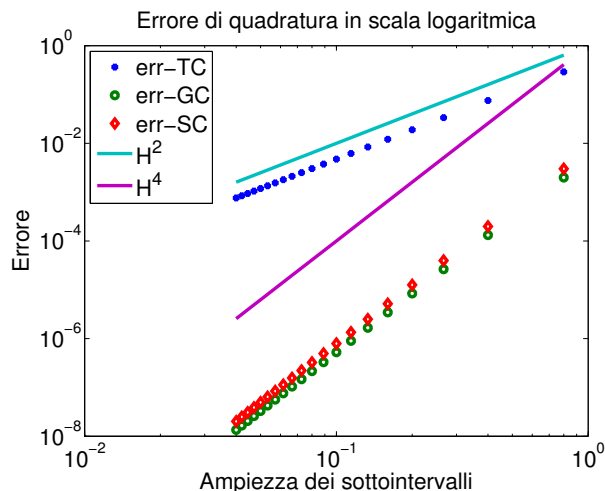
9. Ripetere i Punti 5 e 6 utilizzando questa formula di quadratura e confrontare l'ordine di convergenza con quello ottenuto applicando la formula del trapezio composta (anch'essa formula di quadratura a due punti) e quella di Simpson composta.



**Figura 4:** Confronto tra il valore esatto  $V$  dell'integrale e quello calcolato con il metodo di Gauss composto, al variare del numero di sottointervalli.

Si noti dalla Figura 4 che per Gauss composto si ha convergenza del valore approssimato dell'integrale verso il valore esatto  $V$ , all'aumentare del numero di sottointervalli.

Si osserva dalla Figura 5 che la pendenza della curva dell'errore relativa a Gauss composto e a Simpson composto è pari a 4, ma l'errore per Gauss composto risulta comunque inferiore rispetto a quello di Simpson composto; eppure la formula di quadratura di Simpson è a tre punti, mentre quella di Gauss è a soli due punti! Il metodo dei trapezi composto (a due punti come Gauss) decresce invece solo quadraticamente.



**Figura 5:** Errore di quadratura in scala logaritmica in funzione dell'ampiezza dei sottointervalli. TC = Trapezi Composito, GC = Gauss composito e SC = Simpson Composito.

## Esercizio 2

a) Poniamo  $g_0(t) = t^0 \equiv 1$ ,  $g_1(t) = t$ ,  $g_2(t) = t^2$ . Deve, dunque, essere

$$I_2(g_0) = w_0 + w_1 + w_2 = \int_{-1}^1 1 dt = 2,$$

$$I_2(g_1) = -w_0 + \alpha w_1 + w_2 = \int_{-1}^1 t dt = 0,$$

$$I_2(g_2) = w_0 + \alpha^2 w_1 + w_2 = \int_{-1}^1 t^2 dt = 2/3,$$

da cui

$$w_0 = \frac{1+3\alpha}{3(1+\alpha)}, \quad w_1 = \frac{4}{3(1-\alpha^2)}, \quad w_2 = \frac{1-3\alpha}{3(1-\alpha)}.$$

b) In questo caso deve valere anche

$$I_2(t^3) = -w_0 + \alpha^3 w_1 + w_2 = \int_{-1}^1 t^3 dt = 0$$

da cui si ottiene l'equazione  $4\alpha(\alpha^2 - 1) = 0$ ; poiché deve essere  $|\alpha| < 1$ , la soluzione accettabile è  $\alpha = 0$ , e quindi

$$w_0 = w_2 = 1/3, \quad w_1 = 4/3.$$

c) Introduciamo la trasformazione  $T : [-1, 1] \rightarrow [a, b]$ ,  $T(t) = \frac{1}{2}[(a+b) + t(b-a)]$ . Data una funzione  $f \in C([a, b])$ , poniamo  $g(t) = f(T(t))$ . Grazie al teorema del cambio di



variabili di integrazione, abbiamo

$$\begin{aligned}\tilde{I}(f) &= \int_a^b f(x) dx = \int_{-1}^1 f(T(t)) |T'(t)| dt \\ &= \frac{b-a}{2} \int_{-1}^1 g(t) dt = \frac{b-a}{2} I(g).\end{aligned}$$

La formula di quadratura “mappata” sul nuovo intervallo  $[a, b]$  si scriverà allora

$$\tilde{I}_2(f) = \frac{b-a}{2} I_2(g) = \frac{b-a}{2} \left[ w_0 f(a) + w_1 f\left(\frac{a+b}{2}\right) + w_2 f(b) \right],$$

ovvero

$$\tilde{I}_2(f) = \frac{b-a}{6} \left[ f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right].$$

Osserviamo che questa è precisamente la formula di quadratura di Simpson.

Inoltre, questo esercizio mostra che ogni formula di quadratura interpolatoria  $I_n$  su un intervallo di riferimento  $[a_0, b_0]$ , come ad esempio  $[-1, 1]$ , può estendersi ad una formula  $\tilde{I}_n$  su un intervallo arbitrario  $[a, b]$  tramite una mappa affine  $T$ , con i nodi di  $\tilde{I}_n$  dati da  $\tilde{x}_i = T(x_i)$ , e i pesi dati dai  $\tilde{w}_i = J w_i$ ,  $J = |T'| = \frac{b-a}{b_0-a_0}$ . In tale trasformazione, sia l'integrale esatto che la formula di quadratura vengono semplicemente moltiplicati per il coefficiente  $J$ , dunque anche l'errore di quadratura si trasforma di conseguenza, e la formula  $\tilde{I}_n$  avrà lo stesso grado di esattezza di  $I_n$ .

### Esercizio 3

*Si considerino le funzioni*

$$p_1(x) = x^4 - 2x + 1, \quad p_2(x) = 3x^9 - 5x^4 + 1, \quad g(x) = \frac{10}{x+2}, \quad z(x) = \sqrt{x}, \quad 0 \leq x \leq 1.$$

1. *Calcolarne gli integrali utilizzando la formula di quadratura di Gauss–Legendre semplice di ordine  $n \geq 0$*

$$I_{G,n}(f) = \sum_{i=0}^n \omega_i f(y_i),$$

*per  $n = 0, \dots, 7$ .*

2. *Tracciare un grafico in scala semilogaritmica degli errori di quadratura al variare di  $n = 0, \dots, 7$ ; commentare il risultato alla luce della teoria.*

Consideriamo i seguenti comandi Matlab<sup>®</sup> :

```
x_a = 0;          x_b = 1;
p1 = @(x) x.^4 - 2 * x + 1;
p2 = @(x) 3 * x.^9 - 5 * x.^4 + 1;
g = @(x) 10 ./ ( x + 2 );
z = @(x) sqrt( x );
Iex_p1 = 1 / 5;    Iex_p2 = 3 / 10;
```

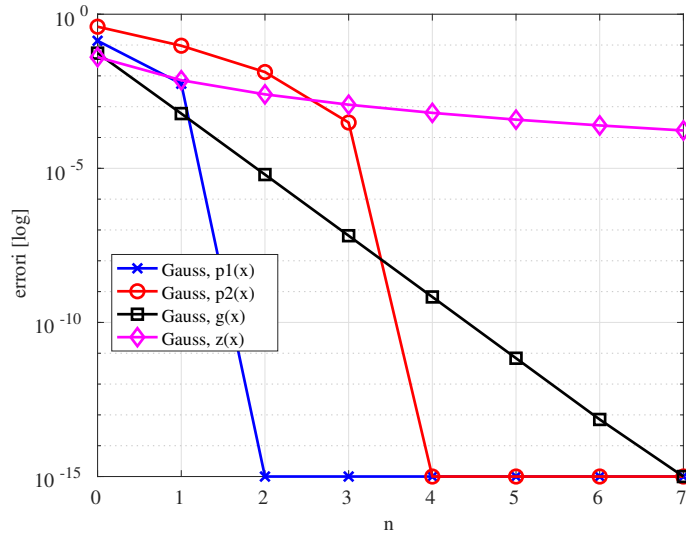
```

Iex_g = 10 * log( 3 / 2 );      Iex_z = 2 / 3;
nv = 0 : 7;
Ep1_gauss = [ ];      Ep2_gauss = [ ];
Eg_gauss = [ ];      Ez_gauss = [ ];

for n = nv
    % nodi di quadratura di Gauss
    [ xi, wi ] = zplege( n, x_a, x_b );
    % integrali approssimati ed errori commesso
    I = sum( wi .* p1( xi ) );
    Ep1_gauss = [ Ep1_gauss, abs( I - Iex_p1 ) ];
    I = sum( wi .* p2( xi ) );
    Ep2_gauss = [ Ep2_gauss, abs( I - Iex_p2 ) ];
    I = sum( wi .* g( xi ) );
    Eg_gauss = [ Eg_gauss, abs( I - Iex_g ) ];
    I = sum( wi .* z( xi ) );
    Ez_gauss = [ Ez_gauss, abs( I - Iex_z ) ];
end

semilogy( nv, max( Ep1_gauss, 1e-15 ), 'bx-', nv, max( Ep2_gauss, 1e-15 ), 'ro-', ...
           nv, max( Eg_gauss, 1e-15 ), 'ks-', nv, max( Ez_gauss, 1e-15 ), 'md-', ...
           'Linewidth', 2, 'MarkerSize', 10);
xlabel('n');      ylabel('errori [log]')
grid on;
legend('Gauss, p1(x)', 'Gauss, p2(x)', 'Gauss, g(x)', 'Gauss, z(x)', ...
       'Location', 'best');

```



**Figura 6:** Errori di quadratura per le formule gaussiane vs.  $n$  in scala semilogaritmica applicate agli integrali delle funzioni  $p_1(x)$ ,  $p_2(x)$ ,  $g(x)$  e  $z(x)$ .

Otteniamo il grafico riportato in Figura 6. Ricordiamo che le formule di quadratura di Gauss–Legendre presentano un grado di esattezza pari a  $2n + 1$ , ovvero integrano esattamente i polinomi di grado minore o uguale a  $2n + 1$ . Verifichiamo dunque che il polinomio  $p_1$ , di grado 4, è integrato esattamente dalle formule con  $n \geq 2$  (numero di

nodi maggiore o uguale a 3), infatti  $2n + 1 \geq 4$  per  $n \geq 2$ ; per tali valori di  $n$  infatti osserviamo un errore nullo (in aritmetica esatta). In maniera simile, il polinomio  $p_2(x)$ , di grado 9, è integrato esattamente per  $n \geq 4$ , come atteso dalla teoria ( $2n + 1 \geq 9$ , da cui  $n \geq 4$ ); osserviamo peraltro una rapida convergenza dell'errore per  $n < 4$ . La funzione  $g(x)$  invece non è un polinomio, ma è una funzione di classe  $C^\infty([0, 1])$ ; non è pertanto pensabile ottenere un errore nullo per qualche  $n \geq 0$ , anche se notiamo una convergenza molto rapida dell'errore al crescere di  $n$  (la convergenza osservata in questo grafico in scala semilogaritmica è di tipo esponenziale rispetto a  $n$ ). Infine, la funzione  $z(x) = \sqrt{x}$  è solamente continua in  $[0, 1]$ , ma non è derivabile con continuità in  $x = 0$  (non è di classe  $C^1([0, 1])$ ); le formule gaussiane esibiscono dunque una convergenza più lenta che nel caso precedente rispetto a  $n$  (in questo grafico in scala semilogaritmica si evidenzia una convergenza di tipo algebrico, inferiore a quella di tipo esponenziale).

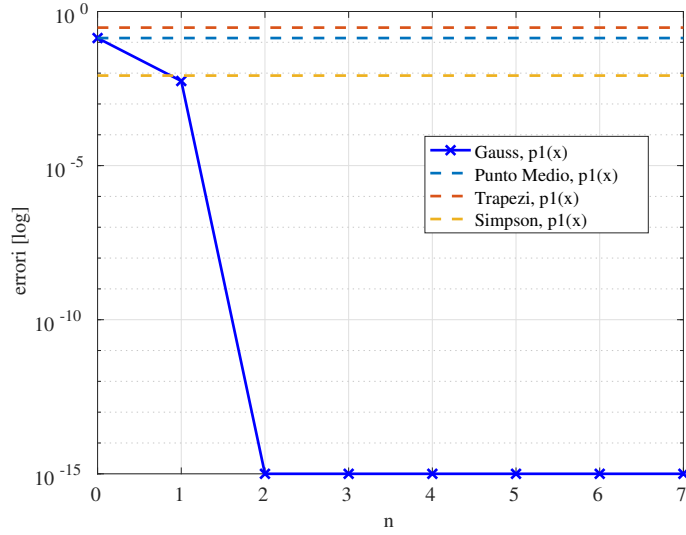
3. *Si confrontino gli errori di quadratura precedentemente ottenuti con quelli associati alle formule del punto medio, trapezi e Simpson semplici.*

Consideriamo per esempio i seguenti comandi Matlab<sup>®</sup> applicati alla funzione  $p_1(x)$  (per le altre funzioni  $p_2(x)$ ,  $g(x)$  e  $z(x)$  possiamo considerare comandi analoghi).

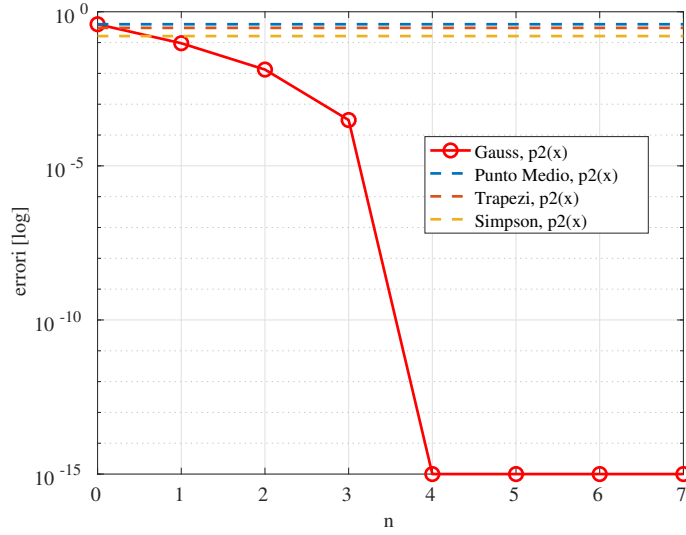
```
Ipm = pmedcomp( x_a, x_b, 1, p1 );
Epm = abs( Ipm - Iex_p1 );
It = trapcomp( x_a, x_b, 1, p1 );
Et = abs( It - Iex_p1 );
Is = simpcomp( x_a, x_b, 1, p1 );
Es = abs( Is - Iex_p1 );

figure( 2 );
semilogy( nv, max( Epl_gauss, 1e-15 ), 'bx-', ...
           nv, max( Epm + 0 * nv, 1e-15 ), '—', ...
           nv, max( Et + 0 * nv, 1e-15 ), '—', ...
           nv, max( Es + 0 * nv, 1e-15 ), '—', ...
           'Linewidth', 2, 'MarkerSize', 10);
xlabel('n');          ylabel('errori [log]')
grid on;
legend('Gauss, p1(x)', 'Punto Medio, p1(x)', 'Trapezi, p1(x)', 'Simpson, p1(x)', ...
       'Location', 'best');
```

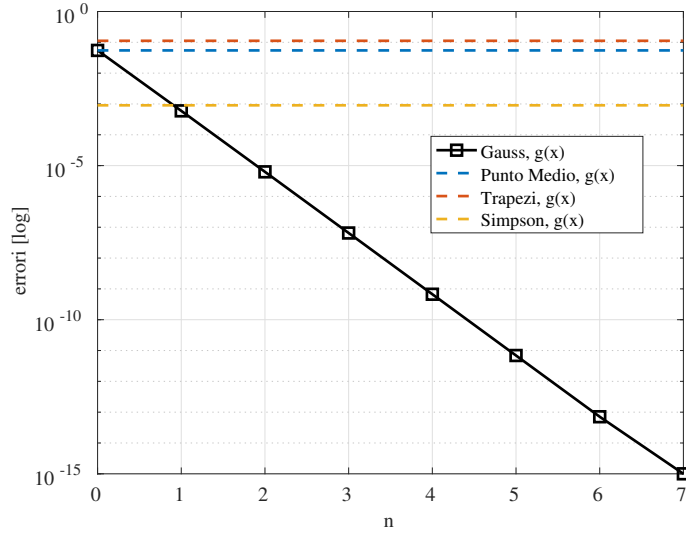
Otteniamo il grafico di Figura 7. Ricordiamo che la formula di Gauss–Legendre per  $n = 0$  coincide con la formula del punto medio; inoltre la formula del trapezio è una formula a due punti ( $n = 1$ ), mentre quella di Simpson a tre punti ( $n = 2$ ). Le formule del punto medio, trapezi e Simpson presentano gradi di esattezza rispettivamente pari a 1, 1 e 3. Osserviamo dunque che, per tutte le funzioni in considerazione (Figure 7–10), le formule gaussiane forniscono errori più bassi di quelle del punto medio, trapezi e Simpson per  $n \geq 1$ . La formula di Simpson si rivela più accurata di quella gaussiana solo per  $n = 0$ . Qualora volessimo migliorare l'accuratezza delle formule del punto medio, trapezi e Simpson, dovremmo utilizzare la loro versione composita, dato che queste operano a un valore di  $n$  fissato.



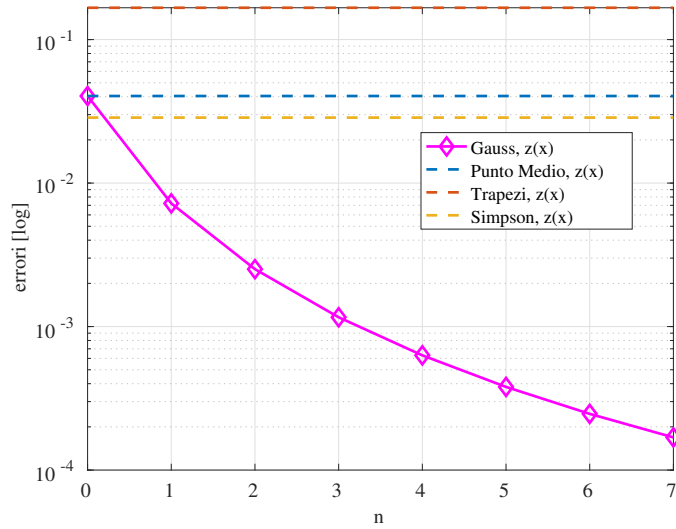
**Figura 7:** Errori di quadratura per le formule gaussiane vs.  $n$  in scala semilogaritmica, del punto medio, trapezi e Simpson applicate agli integrali della funzione  $p_1(x)$  .



**Figura 8:** Errori di quadratura per le formule gaussiane vs.  $n$  in scala semilogaritmica, del punto medio, trapezi e Simpson applicate agli integrali della funzione  $p_2(x)$  .



**Figura 9:** Errori di quadratura per le formule gaussiane vs.  $n$  in scala semilogaritmica, del punto medio, trapezi e Simpson applicate agli integrali della funzione  $g(x)$  .



**Figura 10:** Errori di quadratura per le formule gaussiane vs.  $n$  in scala semilogaritmica, del punto medio, trapezi e Simpson applicate agli integrali della funzione  $z(x)$  .