

Progetto Analisi Numerica

Condurachi Alexei

17 luglio 2020

Introduzione

La risoluzione di equazioni differenziali ordinarie, note anche come ODE nella terminologia inglese, interessa vari campi negli ambiti scientifici. Spesso si conoscono informazioni che riguardano lo sviluppo di un fenomeno nel tempo o nello spazio e si cerca un modello matematico per poter prevedere a partire da queste informazioni il comportamento generale del fenomeno. Tra i vari metodi per le risoluzioni di ODE prendiamo in esame il metodo di Galërkin.

Metodo di Galërkin

Dato un problema definito su uno spazio di Hilbert H , data una forma bilineare $a(\cdot, \cdot) : H \times H \rightarrow \mathbb{R}$ ed una forma lineare $l(\cdot)$, si vuole risolvere l'equazione

$$a(u, v) = l(v) \quad \forall v \in H \quad (1)$$

Il metodo di Galërkin prevede di effettuare la discretizzazione del problema per trovare una soluzione approssimata su una sequenza di sottospazi $H_n \subset H$ tale che:

$$\bigcup_{n=1}^{\infty} H_n = H$$

Su ciascuno di questi sottospazi il problema è risolvibile in modo esatto. Il nuovo problema derivato dalla discretizzazione del dominio, richiede la determinazione della soluzione u_n tale che

$$a(u_n, v) = l(v) \quad \forall v \in H_n$$

Essendo H_n di dimensione finita è possibile determinare una base finita $\{v_i\}_{i=1}^{N_n}$. Poichè u_n appartiene a H_n , si può scrivere u_n come combinazione lineare degli elementi della base di H_n :

$$u_n = \sum_{j=1}^{N_n} c_j v_j \quad (2)$$

Sostituendo (2) in (1) e sfruttando la linearità di a otteniamo la seguente:

$$\sum_{j=1}^{N_n} a(v_j, v) \cdot c_j = l(v) \quad \forall v \in H_n$$

Poichè vale $\forall v \in H_n$, sarà vera anche per gli elementi della base v_i con $i = 1, \dots, N_n$

$$\sum_{j=1}^{N_n} a(v_j, v_i) \cdot c_j = l(v_i) \quad i = 1, \dots, N_n$$

Questo risultato ci permette di dire che, fissata una base $\{v_i\}_{i=1}^{N_n}$ la determinazione della soluzione discretizzata u_n si riduce alla risoluzione del sistema lineare in forma matriciale

$$\mathbf{K}_n \cdot \mathbf{X}_n = \mathbf{F}_n$$

con la seguente notazione:

- $\mathbf{K}_n = \{k_{ij}\}_{i,j=1}^{N_n}$ con $k_{ij} = a(v_j, v_i)$ *(matrice di rigidità)*
- $\mathbf{F}_n = \{f_i\}_{i=1}^{N_n}$ con $f_i = l(v_i)$ *(vettore dei termini noti)*
- $\mathbf{X}_n = \{c_i\}_{i=1}^{N_n}$ *(vettore dei coefficienti)*

Ortogonalità di Galërkin

Notiamo che la differenza $u - u_n$ soddisfa la proprietà nota come *ortogonalità di Galërkin*:

$$a(u - u_n, v_n) = a(u, v_n) - a(u_n, v_n) = l(v_n) - l(v_n) = 0 \quad \forall v_n \in H_n$$

Ovvero il residuo $u - u_n$ è ortogonale al sottospazio $H_n \subset H$.

Problemi differenziali ai limiti

I *problemi differenziali ai limiti* sono quelli nei quali all'equazione differenziale definita in un insieme $[a, b]$, vengono assegnate delle condizioni sulla soluzione agli estremi dell'intervallo (detti anche *limiti*), note in letteratura come *condizioni al contorno di Dirichlet*.

N.B.: poichè stiamo considerando equazioni differenziali lineari non è restrittivo considerare condizioni omogenee. Infatti dato il problema

$$\begin{cases} Lu(x) = f(x) & \forall x \in [a, b] \\ u(a) = \alpha, u(b) = \beta \end{cases}$$

dove L è un operatore differenziale lineare, possiamo facilmente passare al problema omogeneo, considerando

$$y(x) = u(x) - \left(\alpha \frac{b-x}{b-a} + \beta \frac{x-a}{b-a} \right)$$

Infatti

$$\begin{cases} Ly(x) = f(x) + \alpha \frac{b-x}{b-a} + \beta \frac{x-a}{b-a} + \frac{\beta - \alpha}{b-a} =: g(x) & \forall x \in [a, b] \\ y(a) = u(a) - \alpha = 0 & y(b) = u(b) - \beta = 0 \end{cases}$$

Consideriamo dunque da questo punto in poi, il solo caso dei problemi ai limiti con condizioni omogenee.

Parte I

Applicazione per problemi differenziali ai limiti

In questo elaborato si considerano equazioni differenziali del tipo

$$\begin{cases} -u''(x) + u(x) = f(x) & x \in [a, b] \\ u(a) = 0, u(b) = 0 \end{cases} \quad (3)$$

ovvero equazioni differenziali del secondo ordine con condizioni omogenee. Per semplificare la notazione consideriamo $Lu(x) = -u''(x) + u(x)$,

1 Risoluzione tramite metodo di Galërkin

Poniamoci nello spazio $H_0^1([a, b])$, dove con H^1 si indica lo spazio di Sobolev $W^{1,2}([a, b])$ (che in particolare è spazio di Hilbert) e con H_0 si indica il fatto che la soluzione ha traccia nulla agli estremi.

Passiamo ora al problema debole di (3) proiettando tramite il prodotto scalare in L^2

$$(f, g)_{L^2} = \int_a^b f \cdot g$$

Abbiamo così la forma bilineare $a(\cdot, \cdot)$ e $l(\cdot)$

$$a(Lu, v) = (Lu, v)_{L^2} \quad l(v) = \int_a^b f \cdot v$$

in uno spazio di Hilbert.

Consideriamo $H_n \subset H$ lo spazio delle *B-splines lineari* su una mesh creata dalla suddivisione di $[a, b]$ in n intervalli $[t_i, t_i + h]$, $h = \frac{b-a}{n}$ con $i = 0, \dots, n-1$.

Dunque gli elementi della base di H_n , sono le splines lineari φ_i con $i = 0, \dots, n$. Dovremo quindi trovare la soluzione discretizzata $u_n(x)$, combinazione lineare degli elementi della base H_n .

Imponendo però le condizioni al bordo, che u_n dovrà soddisfare per essere soluzione, sappiamo a priori che i contributi di φ_0 e φ_n saranno nulli. Dunque possiamo limitarci a considerare le sole φ_i $i = 1, \dots, n-1$.

Definiamo il residuo $Res(x) := Lu(x) - Lu_n(x)$. Applicando l'ortogonalità di Galërkin al residuo rispetto alle φ_i , otteniamo

$$(Res, \varphi_i) = \int_a^b (Lu(x) - Lu_n(x)) \cdot \varphi_i(x) dx = \int_a^b (f(x) - Lu_n(x)) \cdot \varphi_i(x) dx = 0 \quad i = 1, \dots, n-1$$

Ovvero

$$\int_a^b Lu_n(x) \cdot \varphi_i(x) dx = \int_a^b f(x) \cdot \varphi_i(x) dx \quad i = 1, \dots, n-1$$

Sostituendo l'operatore $Lu_n(x)$

$$\int_a^b (-u_n''(x) + u_n(x)) \cdot \varphi_i(x) dx = \int_a^b f(x) \cdot \varphi_i(x) dx \quad i = 1, \dots, n-1 \quad (4)$$

integrando per parti il membro di sinistra

$$-u_n'(x) \cdot \varphi_i'(x) \Big|_a^b + \int_a^b u_n'(x) \cdot \varphi_i'(x) dx + \int_a^b u_n(x) \cdot \varphi_i(x) dx = \int_a^b f(x) \cdot \varphi_i(x) dx \quad i = 1, \dots, n-1$$

per la struttura delle splines (che sono note)

$$\int_a^b u'_n(x) \cdot \varphi'_i(x) + u_n(x) \cdot \varphi_i(x) dx = \int_a^b f(x) \cdot \varphi_i(x) dx \quad i = 1, \dots, n-1 \quad (5)$$

Poichè $u_n \in H_n$, $u_n = \sum_{j=1}^{n-1} c_j \cdot \varphi_j(x)$. Quindi

$$\sum_{j=1}^{n-1} c_j \int_a^b \varphi'_j(x) \cdot \varphi'_i(x) + \varphi_j(x) \cdot \varphi_i(x) dx = \int_a^b f(x) \cdot \varphi_i(x) dx \quad i = 1, \dots, n-1 \quad (6)$$

Per la struttura delle splines, la (6) si riduce a

$$\sum_{j=i-1}^{i+1} c_j \int_a^b \varphi'_j(x) \cdot \varphi'_i(x) + \varphi_j(x) \cdot \varphi_i(x) dx = \int_a^b f(x) \cdot \varphi_i(x) dx \quad i = 1, \dots, n-1$$

Poichè le splines lineari sono state costruite su una mesh a nodi equispaziati con passo h , esse possono essere viste come ottenute da traslazioni di un'unica spline. Questa osservazione ci permette di dire che è sufficiente il calcolo dell'integrale per $i = 1$. I calcoli svolti sono visibili nell'Appendice (vedere (12))

Sostituendo nella (6) i risultati ottenuti

$$c_{i-1}(\frac{h}{6} - \frac{1}{h}) + c_i(\frac{2}{h} + \frac{2}{3}h) + c_{i+1}(\frac{h}{6} - \frac{1}{h}) = \int_a^b f(x) \cdot \varphi_i(x) dx \quad i = 1, \dots, n-1$$

Moltiplicando entrambi i membri per h

$$c_{i-1}(\frac{h^2}{6} - 1) + c_i(2 + \frac{2}{3}h^2) + c_{i+1}(\frac{h^2}{6} - 1) = h \int_a^b f(x) \cdot \varphi_i(x) dx \quad i = 1, \dots, n-1 \quad (7)$$

Scriviamo il risultato in forma matriciale

$$\mathbf{A} \cdot \mathbf{c} = \mathbf{b}$$

$$\mathbf{A} = \begin{bmatrix} (2 + \frac{2}{3}h^2) & (\frac{h^2}{6} - 1) & 0 & \dots & \dots & 0 \\ (\frac{h^2}{6} - 1) & (2 + \frac{2}{3}h^2) & (\frac{h^2}{6} - 1) & 0 & \ddots & \vdots \\ 0 & (\frac{h^2}{6} - 1) & (2 + \frac{2}{3}h^2) & (\frac{h^2}{6} - 1) & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & (\frac{h^2}{6} - 1) & (2 + \frac{2}{3}h^2) & (\frac{h^2}{6} - 1) \\ 0 & \dots & \dots & 0 & (\frac{h^2}{6} - 1) & (2 + \frac{2}{3}h^2) \end{bmatrix}, \mathbf{b} = \begin{bmatrix} h \int_a^b f(x) \cdot \varphi_1(x) dx \\ \vdots \\ h \int_a^b f(x) \cdot \varphi_{n-1}(x) dx \end{bmatrix}$$

Risolviendo il sistema, ricaviamo il vettore \mathbf{c} che ci permette di determinare la soluzione discreta

$$u_n = \sum_{i=1}^{n-1} c_i \cdot \varphi_i.$$

Parte II

Codice MatLab

Procediamo col mostrare il codice MatLab che implementa la risoluzione del problema (3) tenendo conto delle considerazioni fatte sulla matrice **A** e di conseguenza sul vettore **b** in quanto entrambi i membri sono stati moltiplicati per h . Per il calcolo dei vari integrali si è utilizzata la function predefinita *trapz.m*.

Metodo di Galerkin con calcolo analitico della matrice A

```

1 function [u]=Galerkin_limiti(a,b,f,N,xx)
2 %Metodo di Galerkin per la risoluzione di equazioni differenziali del
3 %secondo ordine del tipo -u''+u=f, con u(a)=u(b)=0
4 %Il calcolo dell'integrale e' stato effettuato in modo analitico
5 %INPUT: a,b estremi
6 %      f membro di destra dell'ODE
7 %      N numero di intervalli in cui viene diviso [a,b]
8 %      xx nodi in cui viene calcolato il valore di u
9 %OUTPUT: u calcolato nei nodi xx
10
11 %Vettore dei nodi che divide [a,b] in N intervalli
12 x=linspace(a,b,N+1);
13 %Matrice delle splines lineari sulla mesh x, calcolate in xx
14 YY=Base_Splines_Lineari(x,xx);
15 %Viene calcolato il valore di f sul vettore xx
16 yy_es=f(xx);
17 %Calcolo del valore del passo h
18 h=(b-a)/N;
19
20 %Creiamo alfa e beta, che costituiscono le diagonali di A tridiagonale
21 %simmetrica
22 beta=(-1+h^2/6)*ones(1,N-2);
23 alfa=(2+2/3*h^2)*ones(1,N-1);
24
25 A=diag(alfa)+diag(beta,-1)+diag(beta,1);
26
27 d=zeros(1,N-1);
28 for i=1:N-1
29     %Integrale di f per le splines lineari moltiplicato per h
30     d(i)=h*trapz(xx,yy_es.*YY(i+1,:));
31 end
32
33 %RisolviAMO il sistema lineare
34 c=A\d';
35 %Ricostruiamo il vettore dei coefficienti aggiungendo c_0 e c_N per ipotesi
36 %nulli
37 Coeff=[0, c', 0];
38 u=zeros(1,length(xx));
39 %Calcolo della soluzione approssimata
40 for i=1:(N+1)
41     u=u+Coeff(i)*YY(i,:);
42 end

```

2 Esempio

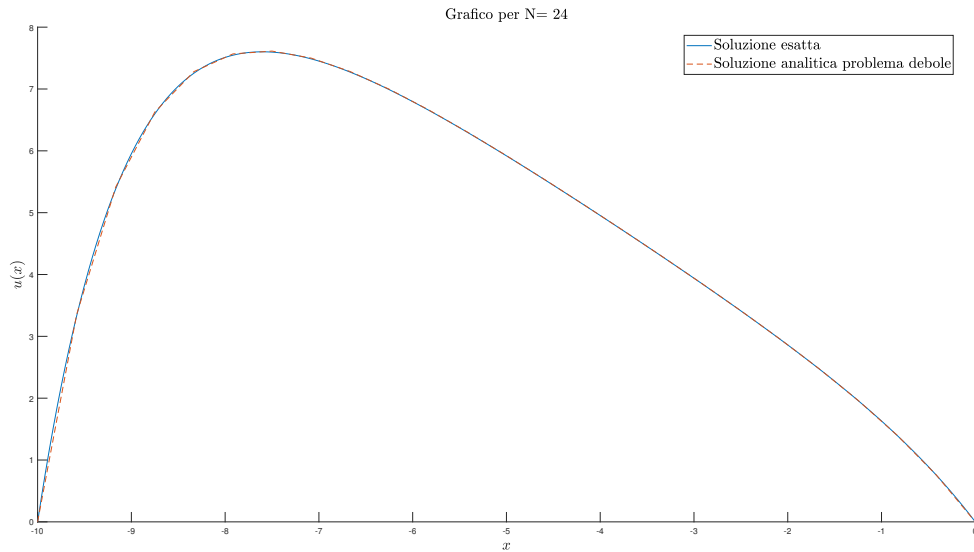
Procediamo con il calcolo di

$$\begin{cases} -u''(x) + u(x) = 1 - x & x \in [-10, 0] \\ u(-10) = 0, u(0) = 0 \end{cases} \quad (8)$$

La cui soluzione esatta è

$$u(x) = 1 + 11e^{10-x} - e^{-x} - 11e^{10+x} + e^{20+x} + e^{20} \frac{x-1}{1-e^{20}}$$

La soluzione approssimata tramite metodo di Galérkin



Vediamo dunque che la soluzione u_n fornisce una buona approssimazione della soluzione esatta u già per N non elevati.

Possiamo apprezzare maggiormente la convergenza del metodo nelle animazioni presenti in Appendice.

Studiamo ora la *convergenza* e l'*ordine* del metodo di Galérkin. Notiamo che si ottiene lo stesso risultato se si aumenta il numero di sottointervalli in cui dividiamo $[a, b]$

$$e_N = |u - u_n| \quad \text{per } N \rightarrow \infty$$

o se studiamo la convergenza al diminuire di h , in quanto

$$\begin{cases} h = \frac{b-a}{N} \Rightarrow h \rightarrow 0 & \text{per } N \rightarrow \infty \\ \lim_{h \rightarrow 0} e_N(h) = 0 \end{cases}$$

Analizziamo l'*ordine* di convergenza dimezzando h . Così facendo, poichè l'errore è un $O(h^p)$ otteniamo che il rapporto

$$\frac{e_i^p}{e_{i+1}^p} = \frac{e_i^p}{(\frac{e_i}{2})^p} = 2^p$$

Per l'esempio (8), la tabella relativa a *convergenza* ed *ordine* è la seguente (N.B per mettere in evidenza p , il rapporto tra due errori successivi è stato normalizzato dividendo per $\log 2$)

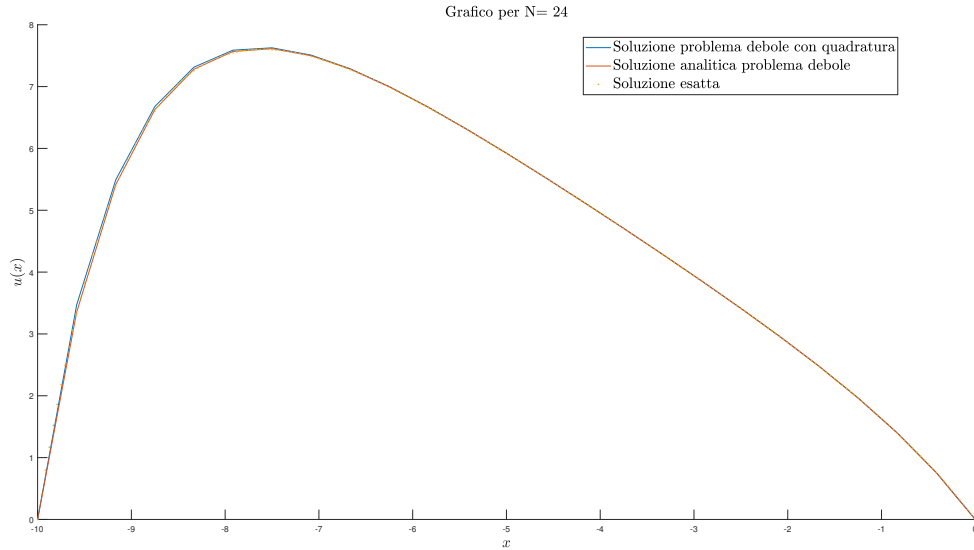
h	Errore	Ordine
5.000000000000000e-01	2.51278816394760e-01	0.00000000000000e+00
2.500000000000000e-01	7.31306983841484e-02	1.78074000788872e+00
1.250000000000000e-01	1.97935738723555e-02	1.88544501466500e+00
6.250000000000000e-02	5.15368150532786e-03	1.94135683757978e+00
3.125000000000000e-02	1.31520303313537e-03	1.97031785090035e+00
1.562500000000000e-02	3.3221952006218e-04	1.98506622174359e+00
7.812500000000000e-03	8.34878290362634e-05	1.99250960436028e+00

Risulta così evidente che il metodo di Galérkin è convergente e ha ordine 2.

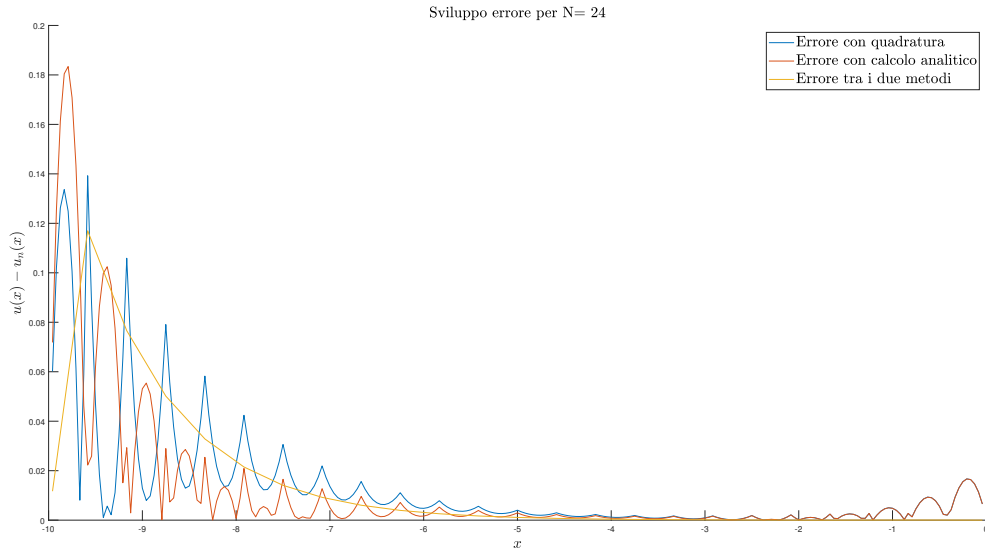
È opportuno fare ora un'osservazione sui passaggi svolti per trovare la soluzione approssimata u_n : per calcolare la *matrice di rigidità* abbiamo seguito un approccio analitico sfruttando

l'*equispazialità* dei nodi della mesh su cui abbiamo costruito le *B-splines lineari*. Ma cosa succede se invece di fare considerazioni di tipo analitico calcoliamo il valore di (6) con una formula di quadratura (ad es. la function *trapz.m* di MatLab)? Il codice MatLab per questa soluzione viene fornito in Appendice.

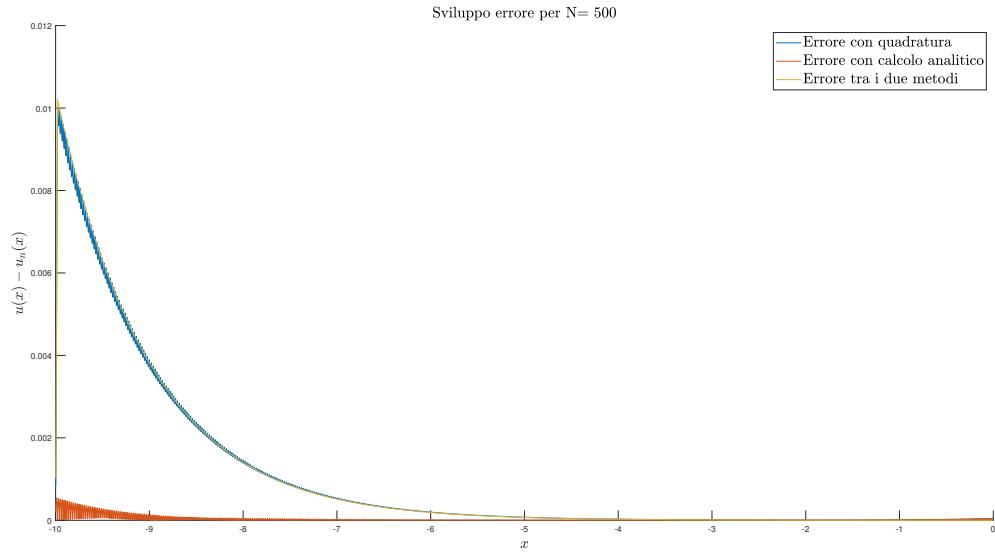
In figura le due soluzioni a confronto



Graficamente non notiamo grosse differenze tra le due soluzioni. È però curioso sottolineare come nei primi nodi la soluzione trovata con la *function* che sfrutta la quadratura piuttosto che i risultati analitici, approssimi in modo migliore u . Questa differenza si assottiglia nei nodi centrali fino a che il metodo con quadratura restituisce una peggiore approssimazione. Ne possiamo osservare il comportamento nella seguente



Bisogna però chiarire che questo comportamento cambia drasticamente al crescere di N . Infatti come mostra la successiva immagine notiamo che la tendenza osservata precedentemente si inverte per N sufficientemente grandi.



Confrontiamo anche la convergenza e l'ordine tra le due scelte e mostriamo come per h sufficientemente piccoli, il metodo con calcoli analitici ha ordine doppio rispetto a quello con formula di quadratura

N	h	Errore Analitico	Errore Quadratura	Ordine Analitico	Ordine Quadratura
2.0000e+01	5.0000e-01	2.2842e-02	2.2874e-02	1.7807e+00	1.2565e+00
4.0000e+01	2.5000e-01	6.6481e-03	9.5740e-03	1.8854e+00	7.9663e-01
8.0000e+01	1.2500e-01	1.7994e-03	5.5117e-03	1.9413e+00	8.7867e-01
1.6000e+02	6.2500e-02	4.6851e-04	2.9976e-03	1.9703e+00	9.3458e-01
3.2000e+02	3.1250e-02	1.1956e-04	1.5683e-03	1.9851e+00	9.6613e-01
6.4000e+02	1.5625e-02	3.0202e-05	8.0279e-04	1.9925e+00	9.8278e-01
1.2800e+03	7.8125e-03	7.5898e-06	4.0622e-04	1.9962e+00	9.9132e-01
2.5600e+03	3.9063e-03	1.9024e-06	2.0433e-04	1.9981e+00	9.9564e-01
5.1200e+03	1.9531e-03	4.7622e-07	1.0248e-04	0.0000e+00	0.0000e+00

È evidente quindi che lo sforzo analitico compiuto viene premiato e in modo significativo. Questo risultato, ragionevole, è dovuto al fatto che il metodo di quadratura porta con sé un errore dovuto al calcolo dell'integrale tramite metodi numerici che si somma all'errore di discretizzazione.

3 Ulteriori esempi

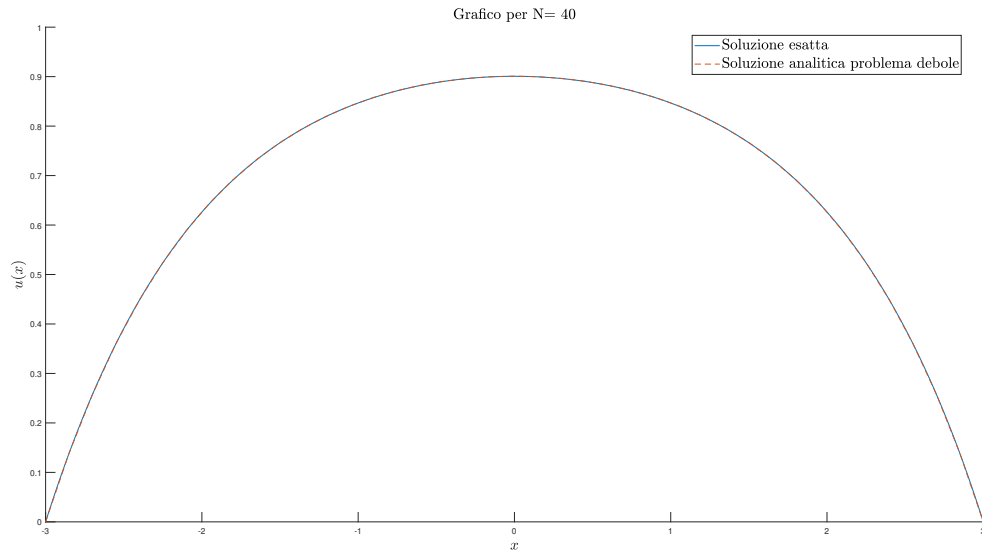
Consideriamo il seguente

$$\begin{cases} -u''(x) + u(x) = 1 \in [-3, 3] \\ u(-3) = 0, u(3) = 0 \end{cases} \quad (9)$$

La cui soluzione esatta è

$$u(x) = \frac{-e^{3-x} - e^{x+3} + 1 + e^6}{1 + e^6}$$

La soluzione approssimata è mostra in figura



e possiamo ancora una volta confrontare gli *errori* e gli *ordini*

h	Errore Analitico	Errore Quadratura	Ordine Analitico	Ordine Quadratura
5.0000e-01	2.2817e-02	2.2825e-02	1.7798e+00	1.2586e+00
2.5000e-01	6.6450e-03	9.5397e-03	1.8851e+00	7.9832e-01
1.2500e-01	1.7990e-03	5.4855e-03	1.9412e+00	8.7891e-01
6.2500e-02	4.6847e-04	2.9829e-03	1.9702e+00	9.3458e-01
3.1250e-02	1.1956e-04	1.5606e-03	1.9850e+00	9.6610e-01
1.5625e-02	3.0201e-05	7.9887e-04	1.9925e+00	9.8276e-01
7.8125e-03	7.5897e-06	4.0424e-04	1.9962e+00	9.9131e-01
3.9063e-03	1.9024e-06	2.0334e-04	1.9981e+00	9.9564e-01
1.9531e-03	4.7622e-07	1.0198e-04	0.0000e+00	0.0000e+00

Notiamo che in (9) l'intervallo è simmetrico e che $f(x)$ è di classe $C^1([-3, 3])$.

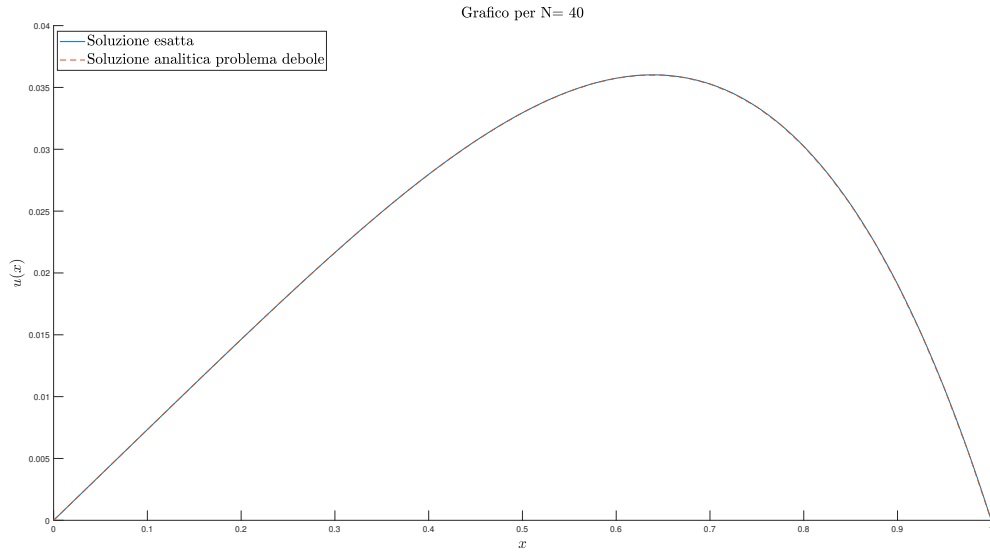
Abbiamo dunque analizzato un caso con l'intervallo interamente negativo (l'esempio (8)) e uno con intervallo simmetrico. Prendiamo in esame ora

$$\begin{cases} -u''(x) + u(x) = x^2 \in [0, 1] \\ u(0) = 0, u(1) = 0 \end{cases} \quad (10)$$

Troviamo che ha soluzione

$$u(x) = e^{-x} \frac{3e - 2e^2 + 2e^{2x} - 3e^{1+2x} - e^x(2 + x^2) + e^{2+x}(2 + x^2)}{e^2 - 1}$$

Di seguito il grafico di u_n



Interessante in questo caso è la tabella dell'errore

N	h	Errore Analitico	Errore Quadratura	Ordine Analitico	Ordine Quadratura
2.0000e+00	5.0000e-01	1.6800e-02	1.6488e-02	1.5456e+00	1.5441e+00
4.0000e+00	2.5000e-01	5.7551e-03	5.6539e-03	1.7764e+00	1.7728e+00
8.0000e+00	1.2500e-01	1.6799e-03	1.6546e-03	1.8906e+00	1.8891e+00
1.6000e+01	6.2500e-02	4.5306e-04	4.4670e-04	1.9458e+00	1.9451e+00
3.2000e+01	3.1250e-02	1.1760e-04	1.1600e-04	1.9730e+00	9.7483e-01
6.4000e+01	1.5625e-02	2.9954e-05	5.9023e-05	1.9866e+00	9.8475e-01
1.2800e+02	7.8125e-03	7.5587e-06	2.9825e-05	1.9933e+00	9.9229e-01
2.5600e+02	3.9063e-03	1.8985e-06	1.4992e-05	1.9966e+00	9.9613e-01
5.1200e+02	1.9531e-03	4.7573e-07	7.5163e-06	1.9983e+00	9.9806e-01
1.0240e+03	9.7656e-04	1.1907e-07	3.7632e-06	1.9992e+00	9.9903e-01
2.0480e+03	4.8828e-04	2.9785e-08	1.8829e-06	1.9996e+00	9.9951e-01
4.0960e+03	2.4414e-04	7.4484e-09	9.4176e-07	0.0000e+00	0.0000e+00

Poichè l'intervallo ha ampiezza 1, già per $N = 2$ otteniamo un soluzione approssimata molto vicina alla soluzione u . Questo è dovuto al fatto che per $N = 2$ il valore del passo è dell'ordine di 10^{-1} .

Come ultimo caso studiamo il caso in cui f sia una funzione C^∞

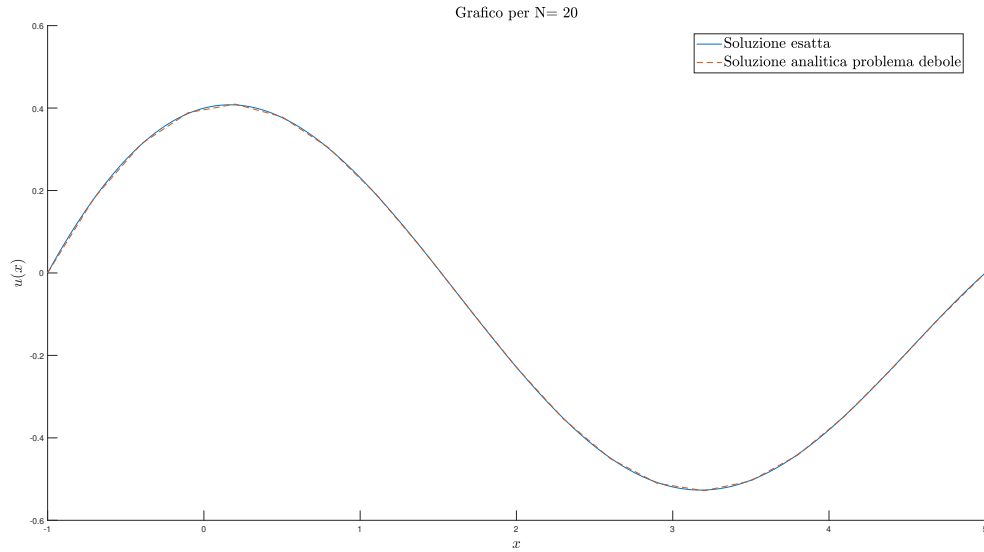
Consideriamo

$$\begin{cases} -u''(x) + u(x) = \cos(x) \in [-1, 5] \\ u(-1) = 0, u(3) = 0 \end{cases} \quad (11)$$

La cui soluzione è

$$u(x) = e^{-x} \frac{(e^{12} - 1)e^x \cos(x) - e^{2x+7} \cos(5) + e^{2x+1} \cos(1) + e^5 \cos(5) - e^1 \cos(1)}{2(e^{12} - 1)}$$

In questo caso la soluzione assume anche valori negativi nel nostro intervallo. Vediamo ora il grafico della soluzione esatta e di quella approssimata



Notiamo ancora una volta che l'esito della tabella dell'*errore* e dell'*ordine* è in linea con le precedenti

h	Errore Analitico	Errore Quadratura	Ordine Analitico	Ordine Quadratura
5.0000e-01	1.6080e-02	1.2780e-02	1.9752e+00	8.8518e-01
2.5000e-01	4.0899e-03	6.9193e-03	1.9802e+00	8.3778e-01
1.2500e-01	1.0366e-03	3.8714e-03	1.9879e+00	8.9283e-01
6.2500e-02	2.6132e-04	2.0850e-03	1.9934e+00	9.4042e-01
3.1250e-02	6.5633e-05	1.0864e-03	1.9965e+00	9.6880e-01
1.5625e-02	1.6448e-05	5.5510e-04	1.9982e+00	9.8406e-01
7.8125e-03	4.1170e-06	2.8063e-04	1.9991e+00	9.9194e-01
3.9063e-03	1.0299e-06	1.4110e-04	1.9995e+00	9.9595e-01
1.9531e-03	2.5756e-07	7.0749e-05	0.0000e+00	0.0000e+00

Parte III

Conclusioni

Dallo studio degli esempi considerati possiamo dire che il metodo implementato è consistente al variare di $f(x)$ (sufficientemente regolare) e degli estremi a e b . Si è anche intrapreso lo studio per $f(x) = \frac{1}{x}$, nell'intervallo $[-1, 1]$. Il metodo implementato in questo caso fallisce la rappresentazione della soluzione in quanto vengono restituiti valori logici *NaN* dovuti alla discontinuità nell'origine. Risolvendo lo stesso problema in $[1, 2]$ riusciamo però ad ottenere una buona approssimazione, poichè in tale intervallo la funzione risulta di classe C^∞ .

Parte IV

Appendice

3.1 Calcolo integrale

Calcolo di (6) per $i = 1$ e $j = i, j = i + 1$

$$\begin{aligned} & c_1 \int_a^b \varphi_j'(x) \cdot \varphi_i'(x) + \varphi_j(x) \cdot \varphi_i(x) dx + c_2 \int_a^b \varphi_j'(x) \cdot \varphi_i'(x) + \varphi_j(x) \cdot \varphi_i(x) dx = \\ & c_1 \left(\int_{x_1}^{x_1+2h} \frac{1}{h^2} dx + \int_{x_1}^{x_1+h} \left(\frac{x-x_1}{h} \right)^2 dx + \int_{x_1+h}^{x_1+2h} \left(-\frac{x-x_1+h}{h} + 2 \right)^2 dx \right) + \\ & + c_2 \left(\int_{x_1+h}^{x_1+2h} -\frac{1}{h} \frac{1}{h} + \left(-\frac{x-x_1+h}{h} + 2 \right) \cdot \left(\frac{x-x_1+h}{h} \right) dx \right) = \\ & c_1 \left(\frac{2}{h} + \frac{2}{3}h \right) + c_2 \left(\frac{h}{6} - \frac{1}{h} \right) \end{aligned} \tag{12}$$

3.2 Animazioni convergenza

Vengono mostrati di seguito le animazioni degli esempi trattati in questo elaborato, per mostrare come al crescere di N la soluzione u_n si avvicina alla soluzione esatta

Animazione per soluzione u_n relativa a (8)

Animazione per soluzione u_n relativa a (9)

Animazione per soluzione u_n relativa a (10)

Animazione per soluzione u_n relativa a (11)