



Corso Luigi Einaudi, 55 - Torino

Appunti universitari

Tesi di laurea

Cartoleria e cancelleria

Stampa file e fotocopie

Print on demand

Rilegature

NUMERO : 139

DATA : 28/09/2011

A P P U N T I

STUDENTE : Franchino

MATERIA : Appunti di Calcolo Numerico
Prof. Pieraccini

Il presente lavoro nasce dall'impegno dell'autore ed è distribuito in accordo con il Centro Appunti.

Tutti i diritti sono riservati. È vietata qualsiasi riproduzione, copia totale o parziale, dei contenuti inseriti nel presente volume, ivi inclusa la memorizzazione, rielaborazione, diffusione o distribuzione dei contenuti stessi mediante qualunque supporto magnetico o cartaceo, piattaforma tecnologica o rete telematica, senza previa autorizzazione scritta dell'autore.

**ATTENZIONE: QUESTI APPUNTI SONO FATTI DA STUDENTIE NON SONO STATI VISIONATI DAL DOCENTE.
IL NOME DEL PROFESSORE, SERVE SOLO PER IDENTIFICARE IL CORSO.**

APPUNTI DI CALCOLO NUMERICO

by Franchino

ANNO: 2010/2011

PROFESSORE: SANDRA PIERACCINI

CORSO: INGEGNERIA GESTIONALE

SISTEMI LINEARI

al fine di ottenere la risoluzione dobbiamo sempre ricondurci ad una matrice triangolare

$$\begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$y_1 = b_1 / l_{11}$$

$$y_2 = (b_2 - l_{21}y_1) / l_{22}$$

IN MATLAB UNA MATRICE TRIANGOLARE

SI RISOLVE:

$$y(1) = b(1) / L(1,1);$$

for $i = 2 : n$

$$y(i) = (b(i) - L(i,1:n-1) * y(1:n-1)) / L(i,i);$$

come ottenere la matrice triangolare?

ELIMINAZIONE GAUSSIANA

$$r_2 \leftarrow r_2 + \left(-\frac{a_{21}}{a_{11}}\right) r_1$$

l'eliminazione gaussiana genera moltiplicatori che annullano la colonna e modificano la riga.

l'eliminazione Gaussiana porta a

Esiste poi la fattorizzazione LU che viene preferita alla prima quando devo risolvere n sistemi equivalenti che variano solo per il vettore dei termini noti.

COSTO COMPUTAZIONALE FATTORIZZAZIONE LU = ELIMINAZIONE GAUSSIANA

$$O\left(\frac{n^3}{3}\right) \text{ moltiplicazioni}$$

questo valore vale quindi sia $x \cdot LU$ che x eliminazione Gaussiana.

COSTO COMPUTAZIONALE: SOLUZIONE DI UN SISTEMA TRIANGOLARE

$$O\left(\frac{n^2}{2}\right) \text{ moltiplicazioni}$$

In generale il metodo di fattorizzazione è conveniente quando si deve risolvere un certo numero di sistemi lineari in cui la matrice dei coef è sempre la stessa e cambia solo vettore termini noti.

il modulo - se $|\lambda| < 1$ allora il sistema converge
 se λ è un valore immaginario calcolo $\sqrt{\text{Re}^2 + \text{Im}^2}$ e guardo se il risultato è < 1 $\rho(B_J) = \lambda < 1$

PASSI METODO JACOBI

scelgo un vettore $x^0 (0, 0, 0)^T$ iniziale, solitamente con questi valori.
 calcolo:

terminato

valori riga in alto x^k

$$x^{k+1} = (b - E x^k - F x^k) / D$$

→ elemento su diag. x^k

↑ valori riga corrisp alla riga x^k in basso

$A = \begin{pmatrix} x & x & F \\ x & D & x \\ E & x & x \end{pmatrix}$

da scrivere x tante x^k quanti sono quelle del vettore x^0 .
 Nel nostro caso sono necessari 3.

applicabilità metodo Gauss-Seidel: come x Jacobi, che sia invertibile.

convergenza metodo Gauss-Seidel: è convergente x matrici a predominanza diagonale

stretta - x matrici simmetriche definite positive (= possiede gli autovallori non nulli positivi).

Nel caso in cui nessuna delle 2 opzioni fosse applicabile applico il teorema di Rosenberg:

$$0 < \rho(B_{GS}) < \rho(B_J) < 1$$

$$1 < \rho(B_J) < \rho(B_{GS})$$

Inoltre: se A è tridiagonale con elementi diag non nulli

$$\rho(B_{GS}) = \rho^2(B_J)$$

Ovvero GS e J convergono, o divergono simultaneamente e il tasso asintotico di conv. è doppio di Jacobi.

ERRORE trovare l'errore conoscendo x^*

1 - studio $|e^0|_\infty = |x^0 - x^*|$

2 - valuto $|e^1|_\infty = |x^1 - x^*|$

come cambia e rispetto ad 1? se diminuisce \Rightarrow l'errore è monotono decresc. Altrimenti è semplicemente decrescente - mai crescente data la condizione di convergenza.

INTERPOLAZIONE E APPROSSIMAZIONE

TEOREMA WEIERSTRASS | sia f una funzione continua tra $[a, b]$

fissato un $\epsilon > 0$, esiste sempre un polinomio $p_n(x)$ tale che

$$\forall x \in [a, b]: |f(x) - p_n(x)| < \epsilon$$

definiamo quindi ϵ come un piccolo errore che discosta il polinomio dalla funzione.

INTERPOLAZIONE POLINOMIALE

BASE DI LAGRANGE

dato un polinomio generico $p_n(x) = \sum a_k l_k(x)$ chiamo la mia base $l_k(x)$. questa base rispetta una variabile binaria δ_k dal momento che $l_k(x_i) = \delta_k = \begin{cases} 1 & i=k \\ 0 & i \neq k \end{cases}$ cio' significa che associa una sola base per ogni punto.

COSTRUZIONE BASE:
$$l_k = \frac{\prod_{i=0, i \neq k}^n (x - x_i)}{\prod_{i=0, i \neq k}^n (x_k - x_i)}$$

quindi a trovare la base l_0 :

$$l_0 = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}$$

questo e' un caso con 3 nodi (x, y) . Quindi dovrai iterare il processo anche x le base l_1, l_2 nello stesso modo.

Trovate le basi impongo condizioni di interpolazione ovvero:

$$p_n(x) = \sum a_k l_k(x) \quad \text{essendo } a_k = y_i \quad \text{il polinomio interp. risultante} \Rightarrow p_n(x) = y_0 l_0(x) + y_1 l_1(x) + \dots + y_n l_n(x)$$

le diff divise possono dare valori = 0 - significa solo che il grado del polinomio sarà $n - 1$ (diff divise = 0) 4/3

RUFFINI - HORNER

è semplic. un modo di scrivere i polinomi - scrivo come + esterno il valore di grado ϕ e raccolgo le x una dentro l'altra -

ERRORE N INTERPOLAZIONE

$$E_n(x) = f(x) - p_n(x) = \frac{f^{(n+1)}(x)}{(n+1)!} \omega_{n+1}(x)$$

↳ resto di Lagrange

$$\|f(x) - p_n(x)\|_{\infty} \leq E_n(f) \cdot (1 + \Lambda_n)$$

con $E_n(f) = \min \|f(x) - q(x)\|_{\infty} : x$ Weierstrass $E_n(f)$ tende a ϕ x
con $(1 + \Lambda_n)$ h che tende a ∞ -

$$\Lambda_n = \left\| \sum_{k=0}^n |l_k(x)| \right\|_{\infty} \left. \vphantom{\sum} \right\} \text{ COSTANTE N LEBESGUE}$$

Λ_n dipende dai nodi e tende a ∞ x $n \rightarrow \infty$ il valore ottimo di questa costante = $\frac{2}{\pi} \log_2 n$ - la distribuzione di nodi che + si avvicina a questo andamento è quella dei nodi di Chebichev. Questi sono simmetrici rispetto all'intervallo e si addensano agli estremi - Se utilizzo nodi equidist $\Lambda_n \gg e^{n/2}$

TEOREMA BERNSTEIN

Sia $f \in C^1[a, b]$ il polinomio interpolante costruito sui nodi di Chebichev converge uniformemente a f se è di classe C^2 allora $\|f(x) - p_n(x)\|_{\infty} = O(1/\sqrt{n})$

TEOREMA DI FABER

x ogni succ di nodi esiste sempre una funzione continua x cui non si ha conv uniforme del polinomio interpolante -

CONVERGENZA DELLA SPLINE CUBICA

$$\|S_3^{(p)}(x) - f^{(p)}(x)\|_\infty = O(h^{2-p}), \quad h \rightarrow 0, \quad p = 0, 1, 2$$

x derivate prime, 2°, ... convergono sempre + lentamente -
 quindi non ho solo la convergenza della spline ma anche delle sue
 derivate questo x $f \in C^2[a, b]$

se $f \in C^k[a, b]$ con $k = 3, 4$; $p = 0, 1, 2, 3$,

$$\|S_3^{(p)} - f^{(p)}(x)\|_\infty = \begin{cases} O(h^{3-p}) \\ O(h^{4-p}) \end{cases}$$

APPROSSIMAZIONE DEI MINIMI QUADRATI

data come modello una funzione e i punti

x	x_1	x_2	x_3
y	y_1	y_2	y_3

$$f(x) = c_1 \varphi_1(x) + c_2 \varphi_2(x) + c_3 \varphi_3(x) \dots$$

prendo in considerazione $\langle 1, x, x^2 \rangle$ x andamenti parabolici

$\langle 1, x, \sin x, \cos x \rangle$ x andamenti sinusoidali -

1 - costruisco la matrice A

che ad ogni colonna corrisponde il valore che il punto assume
 se moltiplicato x φ_i .

$$A = \begin{bmatrix} 1 & x & x^2 \\ 1 & \dots & \dots \\ 1 & \dots & \dots \\ 1 & \dots & \dots \end{bmatrix} \begin{matrix} (x_i) \\ x_2 \\ x_3 \end{matrix}$$

2 - calcolo $A^T \cdot A$

3 - calcolo $A^T y$ (ovvero y = vettore colonne con i valori y del dati)

$$A^T \cdot A \cdot c_i = A^T \cdot y$$

applico elim. Gaussiana e mi riconduco ad una matrice triangolare

$A^T A$ = e' matrice simmetrica def positiva se x_i sono distinti

5 - trovo i valori di c_i li sostituisco nel eq. ne di partenza
 e ho così trovato la soluzione -

QUADRATURA NUMERICA

$$I = \int_a^b f(x) dx \approx I_h$$

$$I_h = \sum_{k=0}^n w_k f(x_k)$$

} generale

quadratura interpolatoria

$$I = \int_a^b f(x) dx = \int_a^b p_h(x) dx$$

polinomio interpolante che approssima l'integrale di $f(x)$ con integrale $p_h(x)$.

$$f(x) = p_h(x) + E_h(x)$$

→ ERRORE anche chiamato R_h o residuo

→ **QUADRATURA ESATTA**: x polinomi interpolanti tali che $R_h = 0$

$$p_h(x) = \sum_{k=0}^n f(x_k) \cdot L_k(x)$$

quindi

$$I = \int_a^b f(x) dx \approx \int_a^b \sum_{k=0}^n f(x_k) \cdot L_k(x) dx + R_h =$$

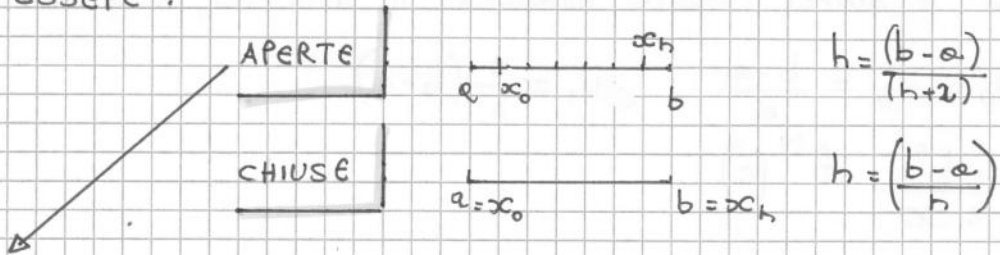
$$= \sum_{k=0}^n f(x_k) \cdot \int_a^b L_k(x) dx + R_h$$

è la base di Lagrange e nell'esercizio di quadrature numerica = pesi w_i

QUADRATURA : FORMULE DI NEWTON - COTES

caratterizzate dai nodi equidistanti - non sono raccomandabili x i pesi assumono segni opposti e se prendo in considerazione molti nodi $\sum w_n f(x)$ si elide.

possono essere :



la distanza tra a e x_0 è cmq uguale alla dist tra 2 nodi qualunque del segmento

CASO 1 : con numero di nodi dispari si ha :

$$R_n = \frac{f^{(n+2)}(c)}{(n+2)!} \cdot (h)^{n+3} \quad d = n+1$$

CASO 2 : con numero di nodi pari si ha :

$$R_n = \frac{f^{(n+1)}(c)}{(n+1)!} \cdot (h)^{n+3} \quad d = n$$

In entrambi i casi $n = \text{numero nodi} - 1$
 $h = \text{distanza tra 2 nodi}$ varia a seconda del caso e se aperte o chiuse -
 h si intende come il numero di sezioni in cui è suddiviso il segmento

ESEMPI DI NEWTON - COTES

sezioni create dai soli nodi, a e b esclusi se non sono nodi.

FORMULA DEL RETANGOLO

possiedo un solo nodo equidistante dai limiti dell'intervallo - $\Rightarrow h = 0 \quad h = \frac{b-a}{n+2}$

$$I_0 = w_0 f(x_0) \rightarrow (b-a) f\left(\frac{a+b}{2}\right)$$

$$R_n(f) = \frac{f''(c)}{24} \cdot (a+b)^3$$

FORMULA DEI TRAPEZI

possiedo 2 nodi equidistanti \Rightarrow cadono in a e in b quindi $h = 1 \quad h = b-a$

$$I_1 = \frac{b-a}{2} f(a) + \frac{b-a}{2} f(b)$$

$$R_n = \left[-\frac{f''(c)}{12} \right] \frac{1}{12}$$

EQUAZIONI NON LINEARI

$$\log_a b = \frac{\log_c b}{\log_c a}$$

le caratteristiche da analizzare che caratterizzano i metodi sono **ROBUSTEZZA**: capacità di conv alla soluzione indipend dalla soluzione iniziale - **VELOCITÀ CONVERGENZA**: TAT + veloce qnt + in fretta, va a zero.

METODO DI BISEZIONE

si basa sul teorema esistenza zeri - prendo sempre il punto medio tra a e b e si capisce in che metà iterare il processo guardo: $f(a_k) \cdot f(b_k) < 0 \Rightarrow$ la radice è nella prima metà altrimenti nella 2^a - questo metodo è molto robusto - in questo metodo l'errore tra la soluzione iterata e l'ottimo: $0 < |e_{k+1}| < \frac{1}{2} \frac{(b-a)}{2^k}$

Quindi si capisce quante iterazioni servono a un'errore pari a 10^{-8} (es) sarà: $k_{k+1} \geq \log_2 \left(\frac{b-a}{10^{-8}} \right)$ calcolato il valore k calcolo bisezione fino a x^k

\rightarrow devo valutare fino alla $k+1$ iteraz. per $k \rightarrow \infty$ $e = 0$ per il Teo del confronto

METODO DI NEWTON O DELLE TANGENTI

In generale scelgo un punto x^0 iniziale - faccio la derivata nel punto e calcolo il punto di intersezione con l'asse delle x .

Dunque $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$ studiare la convergenza Newton

si quanto riguarda la velocità di convergenza: Se $f'(x) = 0 \Rightarrow p = 1$
 $f'(x) \neq 0 \Rightarrow p = 2$

nel primo caso mi trovo di fronte una funzione con radice multiple (parabola) e ovviare questo probl si applica il metodo di quasi Newton

$$x_{k+1} = x_k - \left(m \frac{f(x_k)}{f'(x_{k+1})} \right)$$

METODO SECANTI

In generale si parte da: $f'(x_k) \approx \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$
 in questi casi la velocità di convergenza è $p = \frac{1 + \sqrt{5}}{2} \approx 1,6$

ATTENZIONE: l'intervallo di ampiezza è di 1

METODO DEL PUNTO FISSO | Data una funzione $g(x) \rightarrow$ posso riformulare il problema $f(x)=0$ come problema punto fisso?

devo verificare le 3 condizioni di convergenza - In generale il teorema del punto fisso, valido x funzioni che non intersecano l'asse delle x - quindi consideriamo come punto ottimo E il valore di intersezione tra funzione e bisettrice primo quadrante. Scelto un punto x_0 lo prolungo fino alla bisettrice la L di questo valore incontra la funzione in un altro punto che chiamo x_1 e proseguo fino ad avvicinarsi a E

3 condizioni: 1 - $\exists k < 1 / |f'(x)| < k \forall x \in [a, b]$

Questa condizione la verifico su ciascuna funzione $\phi(x)$ si verifica sempre $k = 1$ con $[a, b]$

2 - $f \in C^1[a, b] \rightarrow$ la funzione deve essere x lo meno di classe 1 - ovvero una volta deriv.

x valutare che sia finita agli estremi controllo $\phi(a), \phi(b)$

3 - $f[a, b] \rightarrow [a, b]$ la funzione è continua e limitata nell'intervallo e finita

* (verificate allora ho un unico punto fisso qualunque x_0 scelga.

Inoltre $f'(E) \neq 0$ allora $\lim_{k \rightarrow \infty} \frac{x_{k+1} - E}{(x_k - E)^{\mu}} = f'(E)$

il risultato mi dice e che cosa converge - quanto vale $\Rightarrow \mu = 1$

se $|f'(x)| > 1$ allora il metodo non può convergere

se $f'(E) = 0$ ma $f''(E) \neq 0$ allora il metodo del punto fisso

converge con ordine $\mu = 2$ $\lim_{k \rightarrow \infty} \frac{x_{k+1} - E}{(x_k - E)^2} = C$

FUNZIONE LIPSCHITZIANA

si dice funzione lip di costante L se

$$\|f(x) - f(y)\| \leq L \|x - y\|$$

con $L < 1$ condizione che sostituisce la 1) e la 3)

Posso quindi riscrivere le condizioni:

1 - $f(D) \subseteq D$

2 - f è lipschitziana di costante L

3 - $L < 1$

allora in tal caso si avrà

EQUAZIONI DIFFERENZIALI

Metodi numerici x risolvere sistemi di eq. differenziali

METODO DI EULERO ESPlicito

è un metodo one step: dato il sistema iniziale

$$\begin{cases} y'(x) = f(x_n, y_h(x)) \\ y(x_0) = y_0 \end{cases}$$

voluto il valore:

$$y_{n+1} = y_n + hf(x_n, y_n)$$

la funzione che lega x con y -
Δx = h

oppure $y_{n+1}^T = y_n^T + h(A \cdot x_n^T + g(x) \cdot y_n^T)$

ESEMPIO:

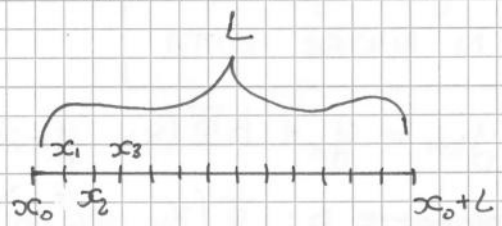
$$\begin{cases} y'(x) = 5y(x) \\ y(0) = 27 \end{cases}$$

dato = 1/10

$$y_1 = y_0 + h \cdot f(x_0, y_0) \rightarrow 27 + \frac{1}{10} (5 \cdot 27) = 31/2$$

$$x_1 = x_0 + \frac{1}{10} = 0 + \frac{1}{10}$$

$h = L/N$ ovvero:



METODO DI EULERO IMPLICITO

$$y_{n+1} = y_n + hf(x_{n+1}, y_{n+1})$$

è valido x funzioni non lineari -

METODO TRAPEZI IMPLICITO

$$y_{n+1} = y_n + \frac{h}{2} (f(x_n, y_n) + f(x_{n+1}, y_{n+1}))$$

METODO DI HEUN

$$y_{n+1} = y_n + \frac{h}{2} (f(x_n, y_n) + f(x_n + hf(x_n, y_n), y_n + hf(x_n, y_n)))$$

METODO EULERO MODIFICATO

$$y_{n+1} = y_n + hf(x_n + \frac{1}{2}h, y_n + \frac{h}{2} f(x_n, y_n))$$

θ - METODO

$$y_{n+1} = y_n + h(\theta f(x_n, y_n) + (1-\theta) f(x_{n+1}, y_{n+1}))$$

RICAVARE LA REGIONE DI ASSOLUTA STABILITÀ X UN METODO

In generale intendiamo regione di assoluta stabilità quella x cui:

$$R_a = \{h\lambda \in \mathbb{C} : |p(h\lambda)| < 1\}$$

x riscontrare tale ipotesi analizziamo un qualsiasi metodo applicandolo su un problema test: $y' = \lambda y$

ESEMPIO SU E.E: $y_{n+1} = y_n + h f(x_n, y_n)$

verifico il metodo sul problema test:

$$y_{n+1} = y_n + h\lambda y_n \Rightarrow y_{n+1} = y_n (1 + h\lambda)$$

$$\Rightarrow R_a = \{h\lambda \in \mathbb{C} : |(1 + h\lambda)| < 1\}$$

ASSEGNATO CAUCHY SCRIVERLO COME SISTEMA AL PRIMO ORDINE

$$\begin{cases} y'''(x) - 4y''(x) + 2y'(x) + 3y(x) = x^2 \\ y(0) = 1 \\ y'(0) = 2 \\ y''(0) = -2 \end{cases}$$

SOSTITUZIONE
STANDARD

i passaggi iniziali sono:

$$\begin{aligned} z_1(x) &= y(x) \\ z_2(x) &= y'(x) \\ z_3(x) &= y''(x) \end{aligned}$$

con questi passaggi riscrivo i valori fino al grado $y^{n-1}(x)$ - dopo di che riscrivo l'intero sistema in z - o meglio in $z_i(x)$ e z'_i

$$\begin{cases} z'_3(x) = 4z_3(x) - 2z_2(x) - 3z_1(x) + x^2 \\ z'_1(x) = z_2(x) \\ z'_2(x) = z_3(x) \\ z_1(0) = 1 \\ z_2(0) = 2 \\ z_3(0) = -2 \end{cases}$$

si vede come ogni z_i è nella
→ forma base e nella forma z'_i

Questo sistema viene poi usato x applicare i metodi di calcolo vengono prima però riscritti nella formula:

$$z(x) = \begin{pmatrix} z_1(x) \\ z_2(x) \\ z_3(x) \end{pmatrix} \quad z(0) = \begin{pmatrix} 1 \\ 2 \\ -2 \end{pmatrix} \quad A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -3 & -2 & 4 \end{pmatrix} \quad g(x) = \begin{pmatrix} 0 \\ 0 \\ x^2 \end{pmatrix}$$

o questo punto verifico la stabilità $\det(\lambda I - A)$ devo avere $\text{Re}(\lambda) < 0$

ASSOLUTA STABILITÀ

1. - prima è necessario analizzare se un problema è asintoticamente stabile $\text{Re} \lambda < 0$ (ricorriamo eq.ni e $y' = \lambda y$ $\text{Re} \lambda < 0$?)

• NEL CASO di una singola equazione:

$$y' = -10y$$

valuto: $y' = \lambda y$ $\lambda = -10$ $\text{Re} \lambda = -10$
quindi è asintoticamente stabile.

• NEL CASO di una eq.ne superiore alla derivata prima:

$$y'' + 5y' + 6y = 0$$

1. eseguo il problema di Cauchy

$$\begin{cases} z_1(0) = 1 \\ z_2(0) = 1 \\ z_1'(x) = -6z_1(x) - 5z_2(x) \\ z_2'(x) = z_2(x) \end{cases}$$

2. da cui: $z'(x) = Az(x)$

$$z(0) = (1, 1)^T \quad A = \begin{pmatrix} 0 & 1 \\ -6 & -5 \end{pmatrix}$$

$$z(x) = (z_1, z_2)^T$$

3. costruisco un polinomio caratteristico da eq.ne di partenza:

$$p = \lambda^2 + 5\lambda + 6$$

$\lambda_1 = -2$; $\lambda_2 = -3$ } sicuramente è asintoticamente stabile xke $\text{Re} \lambda < 0$ x entrambi
In generale valuto il valore + negativo.

2. - verifico assoluta stabilità:

x EULERO-ESPlicito: $|1 + \lambda h| < 1$

Questa formulazione se verificata rappresenta anche la regione di ammissibilità.

METODO TRAPEZI COMPOSITI IN MATLAB

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

$$h = (b - a) / N;$$

$$x = \text{linspace}(a, b, N+1); \rightarrow \text{linspace mi dà i vari valori dei nodi } x_i$$

genera i valori di y corrispondenti in matrice

$$y = \text{eval}(f); \rightarrow \text{preleva i valori della matrice del 2° al N°$$

$$I = 2 * \text{sum}(y(2:N));$$

$$I = h * (y(1) + I + y(N+1)) / 2;$$

METODO SIMPSON COMPOSITO CON MATLAB

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

$$h = (b - a) / N;$$

$$x = \text{linspace}(a, b, 2 * N + 1);$$

$$y = \text{eval}(f);$$

$$I = 4 * \text{sum}(y(2:2:2 * N));$$

$$I = I + 2 * \text{sum}(y(3:2:2 * N - 1));$$

$$I = h * (y(1) + I + y(2 * N + 1)) / 6;$$

function [y] = effe [x]
 $y = \exp((\sin(x) \cdot \wedge 2)) \cdot \log(x \cdot \wedge 3 + 1);$
 $\gg I = \text{trap_compositi}(@\text{effe}, a, b, 100)$

→ ESEGUA UN PASSO DEL METODO EE ←

function [y_np1] = ee (t_n, y_n, h, f)
 tmp = eval(f, t_n, y_n);
 y_np1 = y_n + h * tmp;

function [w] = f(t, y)
 w = -sin(y) - t^3 * exp(t);
 $\gg ee(-3, 5, 0, 1, @f)$