Esercizio: Modificare opportunam il file

LMV_TS_1_pv.sci

in modo che il nuovo file costituisca una realizzazione del metodo TS2.

Oss: Si devono esplicitare due punti:

(A) come si sceglie il passo h_k

(B) come si calcola, scelto h_k, il valore x_{k+1}

Il metodo TS2 si ottiene considerando lo sv di Taylor di una soluz $z(t)$ dell'eq diff in esame:

$$\dot{x}(t) = F(t, x(t))$$

nel generico istante τ...

$\forall \tau, h \; \exists \xi \in [\tau, \tau+h]$ t.c

$$z(\tau+h) = z(\tau) + h\, z'(\tau) + \frac{1}{2} h^2 z''(\tau) + \frac{1}{6} h^3 z'''(\xi)$$

La relazione che consente di calcolare l'appross x_{k+1} all'ist t_{k+1} da x_k, t_k e h_k si ottiene

dallo sviluppo precedente con $\tau = t_k$, $h = h_k$
ed eliminando il resto $\left(\frac{1}{6} h^3 z'''(\xi)\right)$:

$$x_{k+1} = z(t_k) + h_k z'(t_k) + \frac{1}{2} h_k^2 z''(t_k)$$

e considerando che in questo caso:

$$z(t) = x(t; t_k, x_k)$$

Si ottiene:

$$\boxed{x_{k+1} = x_k + h_k F(t_k, x_k) + \frac{1}{2} h_k^2 G(t_k, x_k)}$$

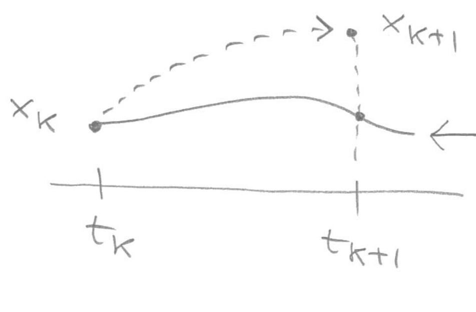
dove $G(t,x)$ è la funzione che restituisce il valore della <u>derivata seconda</u> all'ist t della soluz dell'eq. diff $x(t; t, x)$,
ovvero:

$$G(t,x) = \partial_t F(t,x) + \partial_x F(t,x)\, F(t,x)$$

nel caso di $x \in \mathbb{R}$,

$$G(t,x) = \partial_t F(t,x) + J_F(t,x)\, F(t,x)$$

nel caso di $x \in \mathbb{R}^n$.



Per l'errore locale:

$\leftarrow x(t; t_k, x_k) = z(t)$

$$el_{k+1} = x_{k+1} - z(t_{k+1})$$

$$= x_{k+1} - z(t_k + h_k) =$$

$$= \left[x_k + h_k F(t_k, x_k) + \frac{1}{2} h_k^2 G(t_k, x_k) \right]$$

$$- \left[z(t_k) + h_k z'(t_k) + \frac{1}{2} h_k^2 z''(t_k) + \frac{1}{6} h_k^3 z'''(\xi) \right]$$

$$= \boxed{\frac{1}{6} h_k^3 z'''(\xi)}$$

Dunque:

- TS2 è un metodo di <u>ORDINE 2</u>
- una stima calcolabile dell' err locale all' ist t_k è:

$$\boxed{EL_k \simeq \frac{1}{6} h_{k-1}^3 \, | z'''(t_{k-1}) |}$$

<u>Allora</u>:

(A) Il passo h_k si sceglie imponendo stima $EL_k = E$. Si ottiene:

$$\boxed{h_k = \sqrt[3]{\frac{6E}{\| z'''(t_{k-1}) \|}}}$$

con $z'''(t_{k-1}) = H(t_{k-1}, x_{k-1})$

dove :

$$H(t,x) = \partial_t G(t,x) + \partial_x G(t,x)\, F(t,x)$$

nel caso $x \in \mathbb{R}$ e

$$H(t,x) = \partial_t G(t,x) + J_G(t,x)\, F(t,x)$$

nel caso $x \in \mathbb{R}^n$.

(B) Il valore x_{k+1} si calcola come gia' detto.

Oss: l'intestazione della function sara' del tipo:

[T, X, PASSO, stimaEL] =

LMV_TS_2_pv(xØ, tØ, tf, fct, fct2, fct3, ...

EL_MAX, dialogo)

dove:

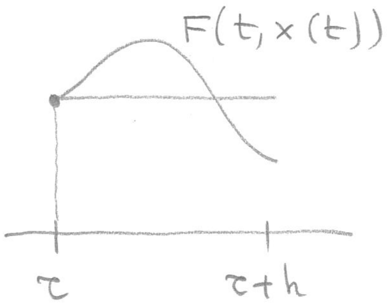
- fct realizza $F(t,x)$
- fct2 realizza $G(t,x)$
- fct3 realizza $H(t,x)$

* Metodo di EULERO IMPLICITO *

- Dall' eq diff $\dot{x}(t) = F(t, x(t))$ si deduce, per ogni τ, h:

$$\boxed{x(\tau+h) - x(\tau)} = \int_{\tau}^{\tau+h} \dot{x}(t)\,dt$$

$$= \boxed{\int_{\tau}^{\tau+h} F(t, x(t))\,dt}$$



<u>Ma</u>:

$$\int_{\tau}^{\tau+h} F(t, x(t))\,dt =$$

$$= h\,F(\tau, x(\tau)) + O(h^2)$$

da cui, eliminando il termine $O(h^2)$ e ponendo $\tau = t_k$, $h = h_k$:

$$x_{k+1} = x_k + h_k\,F(t_k, x_k)$$

(in questo caso $x(t) = x(t; x_k, t_k)$). Si ottiene ancora il metodo di EULERO ESPLICITO.

Ponendo invece:

$$\int_{\tau}^{\tau+h} F(t, x(t))\, dt = h F(\tau+h, x(\tau+h)) + O(h^2)$$

si ottiene (operando come prima):

$$\boxed{x_{k+1} = x_k + h_k F(t_{k+1}, x_{k+1})}$$

Questa relazione definisce il metodo di <u>EULERO IMPLICITO</u>. Il termine si spiega osservando che la relazione definisce x_{k+1} IMPLICITAMENTE: x_{k+1} è <u>uno zero</u> (se esiste) della funzione

$$\boxed{g(z) = z - x_k - h_k F(t_{k+1}, z)}$$

Si può (tentare di) approssimare uno zero della funzione $g: \mathbb{R}^n \to \mathbb{R}^n$ utilizzando, ad esempio, il metodo di Newton. In ogni caso, la ricerca di uno zero di g con un metodo iterativo ad un punto richiede un punto iniziale. Una scelta ragionevole è l'ultima approx fornita dal metodo:

$k = 0$

finché $t_k < t_f$ ripeti

- scegli h_k
- $x_{k+1} = \text{newtonNd}(g, x_k, \dots)$

La scelta di h_k può essere fatta basandosi sulla stima dell'errore locale:

$$EL_k \simeq \frac{1}{2} \| z''(t_{k-1}) \| \, h_{k-1}^2$$

e quindi:

$$\boxed{h_k = \sqrt{\frac{2E}{\| z''(t_{k-1}) \|}}}$$

Una realizzazione del metodo è nel file LMV_eulero_imp_pv.sci riportato alla pag seguente.

```
function [T, X, PASSO, StimaEL]=LMV_eulero_imp_pv(x0, t0, tf, fct, fct2, EL_MAX, dialogo)
//
//  Integra numericamente, sull'intervallo [t0,tf], il problema
//  di Cauchy in R(n):
//   .
//  x = F(t,x)
//  x(t0) = x0
//
//  con il metodo di Eulero all'indietro a passo variabile.
//
//  x0: condizione iniziale (colonna di n elementi)
//  t0: istante iniziale
//  tf: istante finale
//  fct: function per F - fct(t,x) deve essere una colonna
//  fct2: function il cui valore fct2(t,x) è la derivata seconda in t della
//        soluzione dell'equazione differenziale che all'istante t assume valore x.
//  EL_MAX: errore locale massimo consentito
//  dialogo: se "loquace" mostra gli istanti di integrazione
//
//
//  T = [T(1),...,T(N)], nodi
//  X: matrice n x N - la colonna X(:,i) è la soluzione numerica
//     all'istante T(i)
//  PASSO: riga con PASSO(k) = h tale che T(k+1) = T(k) + h
//  StimaEL: riga delle stime dell'errore locale
//
n = length(x0); // determina il numero di equazioni del sistema
h_min = (tf - t0)/1d6; // passo minimo consentito
h_max = (tf - t0)/10;  // passo massimo consentito
T = [];
X = [];
PASSO = [];
StimaEL = [];
//
T(1,1) = t0;
X(:,1) = x0;
StimaEL(1,1) = 0;
//
// ciclo principale
//
while (T(1,$) < tf) & (PASSO(1,$) > h_min | PASSO(1,$) == []),
                        // l'iterazione si arresta se si è
                        // raggiunto tf o se non si è riuscito
                        // a rendere la stima dell'errore
                        // locale inferiore a EL_MAX
    h_max_loc = min(tf - T(1,$), h_max);
    // determina il passo
    Nd2x = norm(fct2(T(1,$),X(:,$)));
    if Nd2x == 0 then
       if PASSO == [] then PASSO(1,$+1) = min(h_min * 100, h_max_loc);
       else PASSO(1,$+1) = min(PASSO(1,$), h_max_loc); end;
    else PASSO(1,$+1) = min(sqrt(2*EL_MAX/Nd2x), h_max_loc);
                 // passo per avere StimaEL = EL_MAX (o non superare tf)
    end;
    // calcola nuovo X e T
    h = PASSO(1,$);
    deff("Y=G(Z)","Y=Z-h*fct(T(1,$)+h,Z)-X(:,$)");
    X(:,$+1) = fsolve(X(:,$),G);
    T(1,$+1) = T(1,$) + PASSO(1,$);
    StimaEL(1,$+1) = (1/2) * Nd2x * PASSO(1,$)^2;
    if dialogo == "loquace" then printf("\nT = %3.2e",T($)); end;
end;
if T(1,$) < tf then
   printf("\n\nIntegrazione interrotta a T = %3.2e", T(1,$));
   printf("\n\nh_min = %3.2e , h = %3.2e\n", h_min, PASSO(1,$));
end;
if dialogo == "loquace" then printf("\n"); end;
//
```