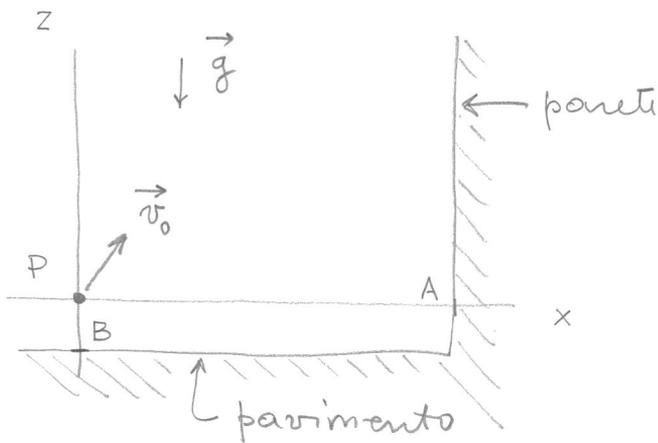


Es :



Un punto  $P$  pesante di massa  $m$  si muove, sotto l'azione della sola forza peso, nell'ambiente rappres. in figura.

Si vuol determinare, numericamente, dove il punto colpirà il pavimento o la parete.

Eq. Newton:  $m\vec{a} = m\vec{g}$  ,  $\vec{a} = \ddot{x}\vec{i} + \ddot{z}\vec{k}$

Pb. di Cauchy: 
$$\begin{cases} \ddot{x} = 0 & ; & x(0) = 0, \dot{x}(0) = \dot{x}_0 \\ \ddot{z} = -g & ; & z(0) = 0, \dot{z}(0) = \dot{z}_0 \end{cases}$$

L'istante finale di integrazione,  $t_f$ , è stabilito dal verificarsi dell'evento:

" il punto ha colpito la parete oppure il pavimento "

L'istante finale  $t_f$  non è noto a priori: è stabilito dal verificarsi di un evento.

Nel caso in esame l'integrazione deve pro-

cedere finché:  $x \leq A$  e  $z \geq B$ .

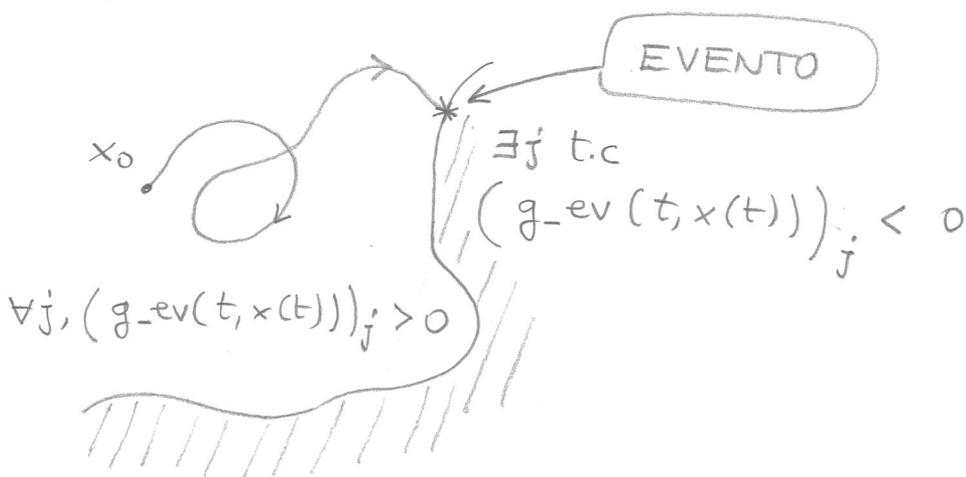
---

la procedura LMV-TS-1-pv-e (vedi pag seguenti)  
integra numericamente il pb di Cauchy in  $\mathbb{R}^n$

$$\begin{cases} \dot{x} = F(t, x) \\ x(t_0) = x_0 \end{cases}$$

con il metodo TS(1) a passo variabile fino  
all'istante in cui si verifica l'evento "una  
almeno delle componenti del vettore  
 $g_{-ev}(t, x)$  ha valore negativo", dove

$g_{-ev} : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^k$  è una funzione,  
definita dall'utilizzatore:



Oss: Quando l'integraz si arresta perché si è  
verif l'evento, la proc comunica quali  
componenti del vettore  $g_{-ev}(t, x(t))$  sono  
diventate  $< 0$ .

```

function [T, X, PASSO, StimaEL]=LMV_TS_1_pv_e(x0, t0, tf, fct, fct2, g_ev, EL_MAX, dialogo)
//
// Integra numericamente il problema di Cauchy in R(n):
// .
// x = F(t,x)
// x(t0) = x0
//
// con il metodo TS(1) - Eulero esplicito - a passo variabile, senza iterazione
// per la scelta del passo, fino all'istante in cui si verifica l'evento "una
// almeno delle componenti del vettore g_ev(t,x(t)) ha valore negativo".
//
// x0: condizione iniziale (colonna di n elementi)
// t0: istante iniziale
// tf: istante finale (raggiunto il quale la procedura si arresta anche se le
// componenti del vettore g_ev(t,x(t)) sono ancora tutte non negative)
// 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.
// g_ev: function che determina quando interrompere l'integrazione - g_ev(t,x) è una colonna
// 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
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,$) == [] & ...
    and(g_ev(T(1,$),X(:,,$)) >= 0),
    // l'iterazione si arresta se si è raggiunto tf o se il passo necessario per ottenere
    // StimaEL = EL_MAX è inferiore a h_min o se una almeno delle componenti del vettore
    // g_ev(T(1,$),X(:,,$)) ha valore negativo
    // passo massimo per questa iterazione:
    h_max_loc = tf - T(1,$);
    // determina 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
    X(:,,$+1) = X(:,,$) + PASSO(1,$)*fct(T(1,$),X(:,,$));
    T(1,$+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
    if PASSO(1,$) <= h_min then
        printf("\n\nIntegrazione interrotta a T = %3.2e", T(1,$));
        printf("\n\nh_min = %3.2e , h = %3.2e\n", h_min, PASSO(1,$));
    else printf("\n\nIntegrazione interrotta a T = %3.2e", T(1,$));
        printf("\n\nperché le componenti:\n\n");
        format("v");disp(find(g_ev(T(1,$),X(:,,$)) < 0));format("e");
        printf("\n\nndi g_ev hanno valore negativo\n");
    end;
end;
if dialogo == "loquace" then printf("\n"); end;
//

```