Esame di Programmazione Informatica

19 gennaio 2022

Informazioni generali

- 2 esercizi di script MATLAB e 2 domande a risposta multipla.
- La durata dell'esame è di 2 ore.
- È consigliato l'uso di MATLAB nello svolgimento di tutti i punti.
- Le slide e gli script del corso sono consultabili liberamente.
- Il punteggio assegnato ad ogni esercizio ed alle domande a risposta multipla è indicato tra parentesi.

Esercizio 1 (14/30)

Si consideri il generico quadrilatero di Figura 1 dove i 4 vertici, ordinati in senso antiorario, hanno coordinate cartesiane $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$:

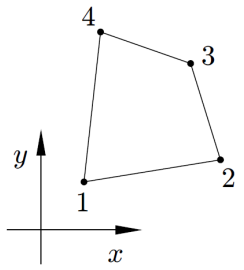


Figura 1: Generico quadrilatero.

L'area del quadrilatero si può calcolare con la seguente formula in funzione delle coordinate dei vertici:

$$A = \frac{1}{2} \sum_{i=1}^{4} (x_i y_{i+1} - y_i x_{i+1}) \tag{1}$$

dove le coordinate con pedice 5 si riferiscono al primo vertice: $(x_5, y_5) = (x_1, y_1)$.

Scrivere una funzione MATLAB che prenda in ingresso:

- un vettore $\mathbf{x} = [x_1, x_2, x_3, x_4]$ delle coordinate x dei 4 vertici,
- un vettore $\mathbf{y} = [y_1, y_2, y_3, y_4]$ delle coordinate y dei 4 vertici,

e restituisca in uscita l'area del quadrilatero.

Si mostri poi come utilizzare la precedente funzione per calcolare l'area del quadrilatero di vertici $(1,0),(4,1),(3,4),(0,3)$.

Soluzione: conviene innanzitutto aggiungere in coda ai due vettori (riga) `x` e `y` le coordinate del primo vertice, come suggerito: $(x_5, y_5) = (x_1, y_1)$.

Possiamo poi operare in maniera vettoriale creando il vettore `i=1:4` degli indici da 1 a 4, richiesti dalla formula (1), e lavorare con esso ricordando di usare le estrazioni vettoriali ed il prodotto elemento per elemento (`.*`). Usiamo infine la funzione `sum()` per sommare tutti gli elementi del vettore risultante:

area_quadrilatero.m

```
function A = area_quadrilatero(x, y)
  x = [ x x(1) ] ;
  y = [ y y(1) ] ;
  i = 1:4 ;
  A = sum( x(i) .* y(i+1) - y(i) .* x(i+1) ) / 2 ;
end
```

Si poteva equivalentemente scrivere un ciclo `for` che scorre sugli indici `i=1:4` e sommando i termini della somma (1) uno a uno.

Utilizzo della precedente funzione nel caso dei vertici $(1,0),(4,1),(3,4),(0,3)$:

```
x = [1 4 3 0] ;
y = [0 1 4 3] ;
A = area_quadrilatero(x, y)
```

ottenendo correttamente $A = 10$.

Esercizio 2 (14/30)

Come da Figura 2, una pallina può andare da A a B seguendo due possibili percorsi in un piano verticale (ossia sotto l'azione della forza di gravità):

- percorso 1: linea retta
- percorso 2: tratto verticale di altezza h + tratto orizzontale di lunghezza w

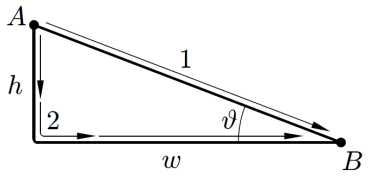


Figura 2: Due percorsi da A a B.

I tempi impiegati dalla pallina per coprire i due percorsi 1 e 2 sono rispettivamente:

$$T_1 = \sqrt{\frac{2\sqrt{h^2 + w^2}}{g \sin \vartheta}} \tag{2}$$

$$T_2 = \sqrt{\frac{2h}{g}} + \frac{w}{\sqrt{2gh}} \tag{3}$$

dove $g = 9.81$ m/s^2 è l'accelerazione di gravità e $h = w \tan \vartheta$.

Scrivere una funzione MATLAB che prenda in ingresso:

- la lunghezza w (in metri),
- l'angolo ϑ (in radianti),

e restituisca in uscita entrambi i tempi T_1 e T_2. Utilizzare possibilmente le operazioni vettoriali (elemento per elemento).

Si mostri poi come utilizzare la precedente funzione per calcolare i due tempi T_1 e T_2 per $w = 1$ m e per 100 valori equidistanziati di ϑ nell'intervallo $\left[\frac{\pi}{20}, \frac{9\pi}{20}\right]$. Fare quindi il plot dei due tempi in funzione dell'angolo ϑ (ϑ in ascissa, T_1 e T_2 in ordinata).

Per angoli ϑ "piccoli" arriva prima la pallina che percorre 1 o quella che percorre 2?

Soluzione: utilizziamo innanzitutto la sintassi per una funzione che restituisce output multipli (i due tempi T_1 e T_2). Dopodichè scriviamo le formule (2) e (3) utilizzando le opportune funzioni native MATLAB `sqrt()`, `sin()`, `tan()`. Poichè sappiamo che ϑ sarà un vettore (e possiamo considerare lo stesso anche per w), utilizzeremo le operazioni vettoriali elemento per elemento per quanto riguarda i prodotti (`.*`), le divisioni (`./`) e gli elevamenti a potenza (`.^`).

`tempi.m`

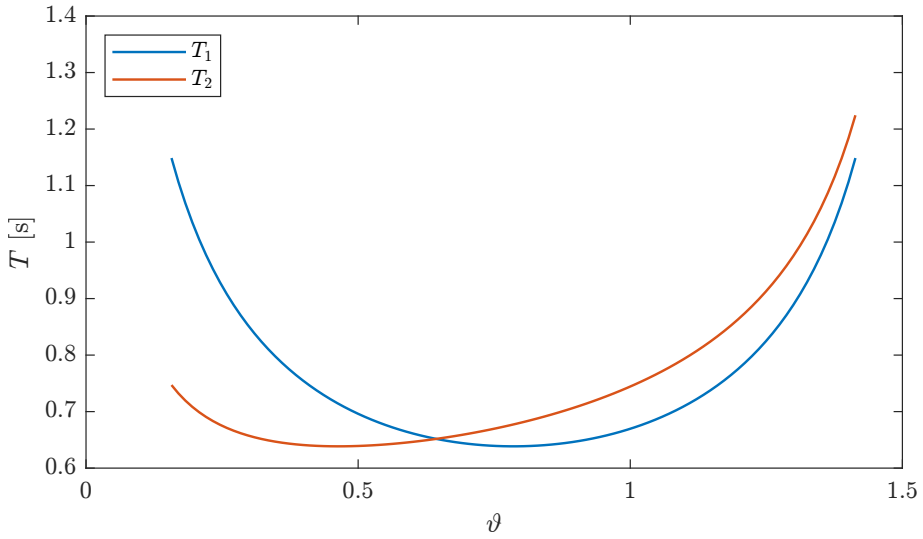
```
function [ T1 , T2 ] = tempi(w, th)
  g  = 9.81 ;
  h  = w .* tan(th) ;
  L  = sqrt( h.^2 + w.^2 ) ;
  T1 = sqrt( 2*L ./ ( g*sin(th) ) ) ;
  T2 = sqrt( 2*h/g ) + w ./ sqrt(2*g*h) ;
end
```

Utilizzo della funzione nel caso di $w = 1$ m e 100 valori equidistanziati di ϑ in $\left[\frac{\pi}{20}, \frac{9\pi}{20}\right]$:

```
w  = 1 ;
th = linspace(pi/20, 9*pi/20, 100) ;
[T1, T2] = tempi(w, th) ;
```

Plot dei due tempi in funzione dell'angolo ϑ:

```
plot(th, T1) ;
hold on ;
plot(th, T2) ;
% Eventuali abbellimenti...
```



Per angoli "piccoli", ossia $\vartheta < \bar{\vartheta} = 2\arctan(2) - \pi/2 \approx 0.6435 = 36.87°$, la pallina che segue il percorso 2, quello più lungo, arriva prima.

Domande a risposta multipla (5/30)

Domanda 1 (2/30)

In MATLAB, quando si eseguono somme con numeri interi relativi, cioè positivi e negativi, bisogna tenere opportunamente conto del *complemento a 2* quando si sommano numeri di segno opposto?

- No, perchè in ogni caso il complemento a 2 verrebbe comunque sfruttato automaticamente a livello della CPU.
- Sì, bisogna calcolare esplicitamente il complemento a 2 per i numeri negativi.
- Sì, perchè altrimenti si otterrebbe un risultato di segno opposto.
- No, perchè in MATLAB non si possono eseguire operazioni con numeri interi.

Soluzione: no, perchè in ogni caso il complemento a 2 verrebbe comunque sfruttato automaticamente a livello della CPU.

Domanda 2 (3/30)

Data la seguente funzione MATLAB:

```
function v = f(x,y,z)
  v = x + y - z ;
  if v > 0
    z =  1 ;
  else
    z = -1 ;
  end
end
```

che funzione matematica $f(x, y, z)$ essa implementa? (sign() è la funzione segno.)

- $f(x, y, z) = \text{sign}(x + y - z)$
- $f(x, y, z) = -\text{sign}(x + y - z)$
- $f(x, y, z) = x + y - z$
- $f(x, y, z) = z$

Soluzione: $f(x, y, z) = x + y - z$.