

Esame di Programmazione Informatica

2 febbraio 2022

Esercizio 1 (14/30)

Si consideri il generico pentagono della Figura 1, dove i 5 vertici, ordinati in senso antiorario, hanno coordinate cartesiane (x_1, y_1) , (x_2, y_2) , (x_3, y_3) , (x_4, y_4) , (x_5, y_5) :

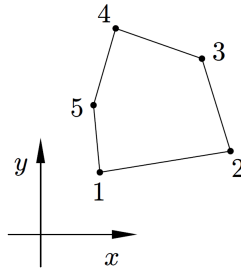


Figura 1: Generico pentagono.

Scrivere una funzione MATLAB che prenda in ingresso:

- un vettore $\mathbf{x} = [x_1, x_2, x_3, x_4, x_5]$ delle coordinate x dei 5 vertici,
- un vettore $\mathbf{y} = [y_1, y_2, y_3, y_4, y_5]$ delle coordinate y dei 5 vertici

e restituisca in uscita il perimetro P del pentagono. Si utilizzi il solito teorema di Pitagora per calcolare la lunghezza L di ciascun lato:

$$L = \sqrt{\Delta x^2 + \Delta y^2}$$

Si mostri poi come utilizzare la precedente funzione per calcolare il perimetro P del pentagono di vertici $(1, 0)$, $(4, 1)$, $(3, 4)$, $(0, 3)$, $(0, 0)$.

Soluzione: conviene innanzitutto aggiungere in coda ai due vettori $\mathbf{x}$ e $\mathbf{y}$ le coordinate del primo vertice: $(x_6, y_6) = (x_1, y_1)$.

Possiamo poi operare in maniera vettoriale creando il vettore $\mathbf{i}=1:5$ degli indici da 1 a 5, e lavorare con esso ricordando di usare le estrazioni vettoriali, l'elevamento a potenza elemento per elemento ($\wedge$) e la funzione `sqrt()` che opera automaticamente elemento per elemento. Usiamo infine la funzione `sum()` per sommare le lunghezze di ciascun lato:

perimetro_pentagono.m

```
function P = perimetro_pentagono(x, y)
    x(6) = x(1) ;
    y(6) = y(1) ;
    i = 1 : 5 ;
    dx = x(i+1) - x(i) ;
    dy = y(i+1) - y(i) ;
    P = sum( sqrt( dx.^2 + dy.^2 ) ) ;
end
```

Si poteva equivalentemente scrivere un ciclo `for` che scorre sugli indici `i=1:5` e sommando le lunghezze di ciascun lato uno a uno.

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

```
x = [ 1 4 3 0 0 ] ;
y = [ 0 1 4 3 0 ] ;
P = perimetro_pentagono(x, y)
```

ottenendo correttamente $P = 13.4868\dots$

Esercizio 2 (14/30)

Vi sono N masse $m_1, m_2, \dots, m_N$ disposte lungo l'asse x alle rispettive coordinate $x_1, x_2, \dots, x_N$, come da Figura 2:

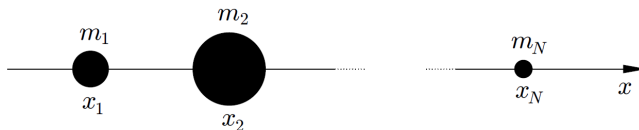


Figura 2: N masse lungo x .

Il baricentro x_B della distribuzione delle masse è dato da:

$$x_B = \frac{\sum_{i=1}^N m_i x_i}{\sum_{i=1}^N m_i} \quad (1)$$

mentre il momento d'inerzia I rispetto ad un generico punto di coordinata $\bar{x}$ è dato da:

$$I = \sum_{i=1}^N m_i (x_i - \bar{x})^2 \quad (2)$$

Scrivere una funzione MATLAB che prenda in ingresso:

- il vettore $\mathbf{m} = [m_1, m_2, \dots, m_N]$ delle N masse,
- il vettore $\mathbf{x} = [x_1, x_2, \dots, x_N]$ delle coordinate delle N masse,
- la coordinata scalare $\bar{x}$

e restituisca in uscita:

- il baricentro x_B della distribuzione delle masse,
- il momento d'inerzia I rispetto al punto $\bar{x}$.

Si mostri poi come utilizzare la precedente funzione per calcolare il baricentro x_B di $N = 100$ masse $m_1 = m_2 = \dots = m_N = 1$ disposte alle coordinate $x_1 = 1, x_2 = 2, \dots, x_i = i, \dots, x_N = N$ ed il momento d'inerzia I per $\bar{x} = x_B$. Per valori di $\bar{x} \neq x_B$ il momento d'inerzia è più grande o più piccolo?

Soluzione: utilizziamo innanzitutto la sintassi per una funzione che restituisce output multipli (il baricentro x_B ed il momento d'inerzia I). Dopodichè scriviamo le formule (1) e (2) utilizzando le operazioni vettoriali elemento per elemento per quanto riguarda i prodotti ($.*$) e gli elevamenti a potenza ($.^$). Usiamo come al solito la funzione `sum()` per sommare tutti gli elementi di un vettore:

prop_masse.m

```
function [xB, I] = prop_masse(m, x, x_bar)
    xB = sum(m .* x) / sum(m) ;
    I   = sum(m .* (x-x_bar) .^ 2 ) ;
end
```

Utilizzo della funzione nel caso di $N = 100$ masse tutte di valore unitario, le cui coordinate $x_i = i, i = 1, \dots, N$ possono essere facilmente generate con il solito comando `x=1:N`. Se la precedente funzione utilizza operazioni vettoriali, i vettori **m** e **x** dovranno avere la stessa dimensione:

```
N = 100 ;
m = 1 * ones(1, N) ;
x = 1 : N ;
x_bar = 0 ;
[xB, I] = prop_masse(m, x, x_bar) ;
[xB, I] = prop_masse(m, x, xB)
```

ottenendo correttamente $x_B = 50.5$ ed $I = 83325$ calcolato rispetto $\bar{x} = x_B$ (dimentichiamo le unità di misura). Per qualsiasi altro valore di $\bar{x} \neq x_B$ il momento d'inerzia risulta sempre maggiore, come da teoria.

Domande a risposta multipla (5/30)

Domanda 1 (2/30)

Si considerino due immagini Bitmap a colori con quantizzazione a 8 bit per ognuno dei 3 canali RGB (ossia $2^8 = 256$ livelli per ciascun canale), la prima ha risoluzione 400×200 pixel mentre la seconda ha, rispetto alla prima, una risoluzione doppia lungo entrambe le dimensioni (righe e colonne).

La seconda immagine, rispetto alla prima, occupa:

- 2 volte tanto.
- 4 volte tanto.
- 8 volte tanto.
- Non si può dire con precisione perchè dipende dalla compressione.

Soluzione: Raddoppiando entrambe le dimensioni, lo spazio richiesto aumenta di 4 volte.

Domanda 2 (3/30)

Data la seguente funzione MATLAB:

```
function y = f(x)
    if x > 0
        y = -f(x)^3 ;
    else
        y = f(x)^3 ;
    end
end
```

che funzione matematica essa implementa?

- Nessuna perchè è una funzione ricorsiva che non si conclude mai.
- $-|y^3|$
- $|y^3|$
- $|x|$

Soluzione: Nessuna perchè è una funzione ricorsiva che non si conclude mai.