Esame di Programmazione Informatica

15 settembre 2023

Esercizio 1 (15/30)

Si consideri, nel piano, un punto P identificato dalle sue coordinate cartesiane (x_P, y_P) ed una retta nella forma implicita $ax + by + c = 0$. La distanza del punto P dalla retta è dato dalla formula:

$$d_i = \frac{|ax_P + by_P + c|}{\sqrt{a^2 + b^2}}$$

[9 punti] Scrivere una funzione MATLAB che prenda come argomenti in ingresso:

- le coordinate cartesiane di N punti nel piano, sotto forma di una matrice di N righe e 2 colonne: la prima colonna conterrà le coordinate x e la seconda colonna conterrà le coordinate y degli N punti
- i tre scalari a, b, c che identificano la retta

e restituisca come argomento in uscita il vettore $\mathbf{d}$, di N elementi, delle distanze degli N punti dalla retta.

[2 punti] E' possibile un'implementazione mediante funzione anonima? Se sì, mostrarla.

[4 punti] Mostrare poi come utilizzare la precedente funzione per calcolare le distanze di 100 punti equidistanziati su una circonferenza di raggio 5 e centrata nel punto di coordinate $(1, 2)$ (utilizzare le funzioni goniometriche `cos` e `sin`), nel caso di una retta identificata da $a = 1, b = 1, c = 0$.

Soluzione: in questo caso procediamo direttamente con l'implementazione mediante funzione anonima grazie alle convenienti operazioni vettoriali che permettono di scrivere la funzione con una singola riga di codice. Data la matrice `M` delle coordinate degli N punti, ne estraiamo le colonne con l'indicizzazione vettoriale: la prima colonna (coordinate x) sarà `M(:,1)` e la second colonna (coordinate y) sarà `M(:,2)`. Usiamo la funzione `abs()` che può anch'essa agire in maniera vettoriale:

```
d = @(M,a,b,c) abs( a*M(:,1) + b*M(:,2) + c ) / sqrt(a^2+b^2) ;
```

Sfruttando il prodotto matrice-vettore, la funzione può essere ancora più compatta:

```
d = @(M,a,b,c) abs( M*[a;b] + c ) / sqrt(a^2+b^2) ;
```

Calcolo delle distanze di 100 punti equidistanziati su circonferenza di raggio 5 centrata in $(1,2)$ dalla retta $a=1, b=1, c=0$.

```
% Circonferenza
R  = 5 ;
x0 = 1 ;
y0 = 2 ;
%    linspace crea un vettore riga, quindi trasponiamo con ' per
%    ottenere un vettore colonna
t  = linspace(0, 2*pi, 100)' ;
x  = x0 + R*cos(t) ;
y  = y0 + R*sin(t) ;

% Retta
a = 1 ;
b = 1 ;
c = 0 ;

% Distanze
distanze = d([x y],a,b,c) ;
```

Esercizio 2 (18/30)

Il periodo di oscillazione T di un pendolo semplice è dato dalla seguente formula:

$$T = T_0 \left(1 + \sum_{n=1}^{+\infty} \left[\frac{(2n-1)!!}{(2n)!!} q^n \right]^2 \right)$$

dove $T_0 = 2\pi\sqrt{L/g}$, L è la lunghezza del pendolo, g è l'accelerazione di gravità, θ è l'angolo massimo di oscillazione e $q = \sin(\theta/2)$. La funzione !! è il doppio fattoriale ed è implementabile in MATLAB mediante la seguente funziona anonima:

```
fatt_doppio = @(n) prod(n:-2:1)
```

che andrà definita dentro alla funzione successivamente richiesta (vedi template di risposta).

[12 punti] Scrivere una funzione MATLAB che prenda come argomenti in ingresso:

- l'angolo massimo di oscillazione θ
- la lunghezza del pendolo L
- l'accelerazione di gravità g
- il numero massimo di termini, N, da considerare per la somma della formula

e restituisca come argomento in uscita il periodo T.

[4 punti] Mostrare poi come utilizzare la precedente funzione nel caso $\theta = \pi/2$, $L = 1$ m, $g = 9.81$ m/s^2 ed $N = 50$ termini. Che periodo T si ottiene?

[2 punti] Cosa succede al periodo T aumentando l'angolo massimo θ? Aumenta o diminuisce?

Soluzione: utilizziamo un semplice ciclo `for` per sommare N termini della serie. Sfruttiamo quindi la funzione anonima `fatt_doppio` già fornita per calcolare i due valori di $(2n-1)!!$ e $(2n)!!$ richiesti nella formula:

periodo_pendolo.m

```
function T = periodo_pendolo(theta, L, g, N)
  fatt_doppio = @(x) prod(x:-2:1) ;
  q = sin(theta/2) ;
  s = 1 ;
  for n = 1 : N
    s = s + ( fatt_doppio(2*n-1) / fatt_doppio(2*n) * q^n )^2 ;
  end
  T0 = 2*pi*sqrt(L/g) ;
  T  = T0 * s ;
end
```

Utilizzo nel caso $\theta = \pi/2$, $L = 1$ m, $g = 9.81$ m/s^2 ed $N = 50$ termini:

```
theta = pi/2 ;
L     = 1 ;
g     = 9.81 ;
N     = 50 ;
T     = periodo_pendolo(theta, L, g, N)
```

ottenendo $T = 2.3678$ s. Aumentando l'angolo massimo θ, per esempio $\theta = 3\pi/4$, il periodo aumenta:

```
theta = 3*pi/4 ;
T     = periodo_pendolo(theta, L, g, N)
```

ottenendo $T = 3.0651$ s.