**Indice**

1. Il quesito 2

2. Discussione analitica 2

3. Soluzione analitica 6

4. Algoritmo per la soluzione al calcolatore 7

5. Esecuzione del programma 8

6. Soluzione grafica del punto 1.4 10

7. Soluzione grafica del punto 1.5 12

8. Codice del programma principale 13

9. Codice del modulo 16

**Cicloide**

**1. Il quesito.** Una ruota avente raggio e velocità angolare di seguito assegnati rotola senza strisciamento su una retta fissa.

1.1)

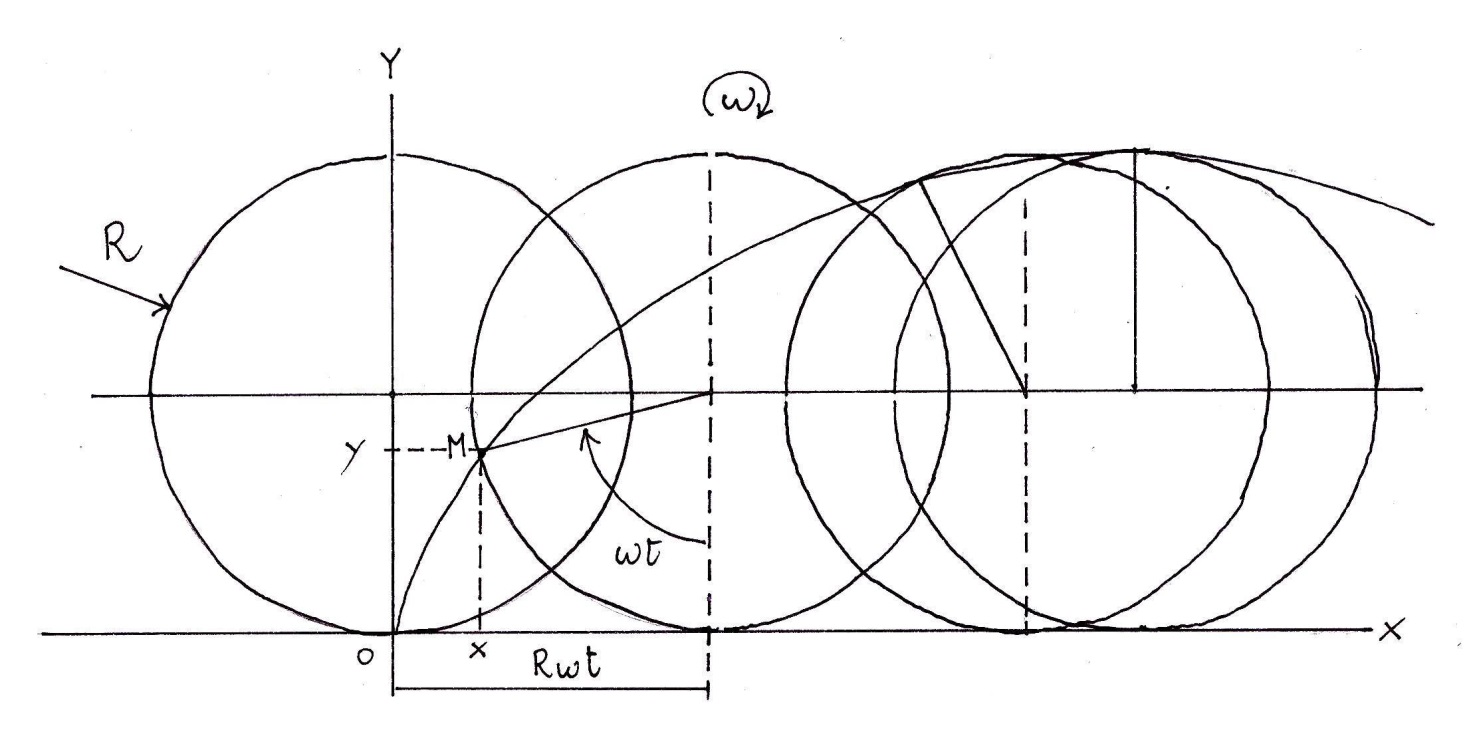
Detto il punto di contatto fra circonferenza e retta all’istante iniziale, si determini allora

1.2) la traiettoria descritta da durante il moto;

1.3) il diagramma dello spazio percorso da in funzione del tempo;

1.4) la velocità e l’accelerazione di e il centro di curvatura in , in corrispondenza a una rotazione ;

1.5) la velocità e l’accelerazione di e il centro di curvatura in , in corrispondenza a uno spazio percorso .



**2. Discussione analitica.** Dato il sistema di riferimento in figura, sia il punto della ruota il quale sia coincidente con per . Volendo ricavare **le coordinate di in funzione del tempo** si ricava dalla figura quanto segue:

2.1)

Per **l’analisi cinematica del primo ordine** del punto derivando le **2.1** si ha

2.2)

Derivando ancora le **2.3** si ottiene **l’analisi cinematica del secondo ordine**

2.3)

Assumendo il punto come origine della ascissa curvilinea abbiamo che lo spazio percorso nell’intervallo di tempo è dato da

Ricordando la formula di dimezzamento

abbiamo[[1]](#footnote-1) che

Ho trovato dunque **lo spazio percorso da in funzione del tempo**

2.4)

Il **versore tangente alla traiettoria** si scrive

2.5)

Il **versore normale** invece si scrive

2.6)

ed essendo poi

la **2.6** si riscrive

Dunque ho trovato

2.7)

Si dimostra poi che il **raggio di curvatura**, ovvero il raggio della circonferenza che oscula la traiettoria, è dato da

2.8)

Ma allora il centro di curvatura della traiettoria di in medesimo è individuato istante per istante dal vettore

Dunque abbiamo che il centro di curvatura ha coordinate

2.9)

Sostituendo nelle **2.9** le **2.1**, **2.2**, **2.3** si ha

2.10)

Volendo la **curvatura** -al fine di calcolare accelerazione angolare e tangenziale del punto che descrive la cicloide- si osservi che

2.11)

Poiché inoltre la **2.4** porge

2.12)

abbiamo per **l’accelerazione normale** che

e per **l’accelerazione tangenziale** che

Quindi si conclude[[2]](#footnote-2) che

2.12)

**3. Soluzione analitica.** Per rispondere al quesito **1.4** si consideri che

3.1)

3.2)

3.3)

3.4)

3.5)

Per il quesito **1.5** si consideri invece che la **2.4** porge

Dunque si ha

3.5)

3.6)

3.7)

3.8)

3.9)

**4. Algoritmo per la soluzione al calcolatore.** Il seguente diagramma di flusso risolve i quesiti **1.2**, **1.3**, **1.4**, **1.5**.

Inizio

invoca la subroutine *cicloide* per disegnare la posizione del cerchio istante per istante oltre alla traiettoria di e di

fine

invoca la subroutine *spazio* per disegnare l’andamento dello spazio percorso nel tempo

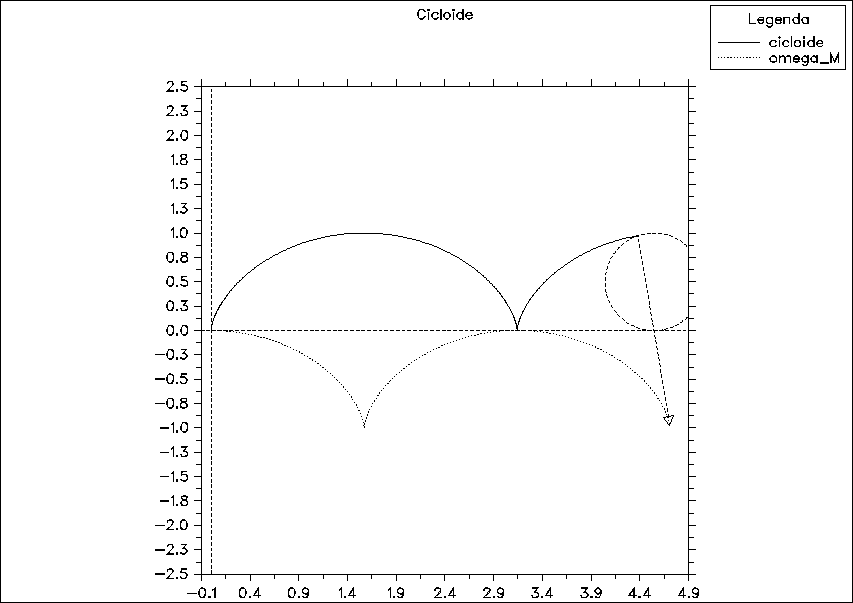
calcola la posizione di per e per , nonché la velocità e l’accelerazione di in questi due istanti, utilizzando le **2.1**, **2.2**, **2.3**, **2.9**

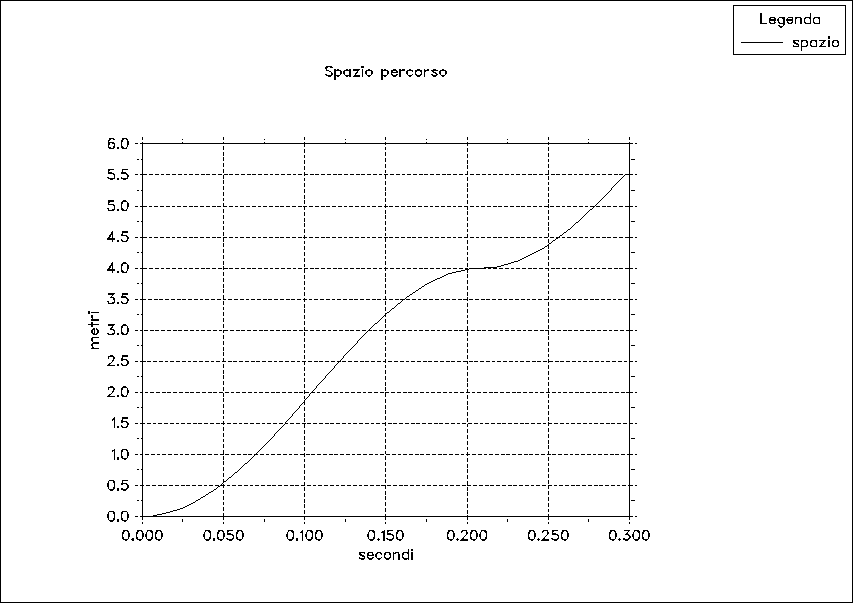
Il diagramma di flusso è sviluppato dai seguenti codici:

* l’unità chiamante *main\_ese\_9\_a*,la quale si occupa di calcolare gli array associati alle **2.1**, **2.4**, da passare poi alle subroutine grafiche, in modo da ottenere una successioni di fotogrammi in cui la circonferenza descrive la sua rotazione mentre il punto traccia la cicloide; e di ottenere il diagramma dello spazio percorso nel tempo;
* il modulo *mod\_ese\_9\_a*, il quale contiene i parametri **1.1** e le seguenti subroutine:
* *cicloide*, la quale traccia per ogni istante la posizione della circonferenza, la traiettoria percorsa a quell’istante da e dal centro di curvatura della cicloide;
* *spazio,* la quale si occupa di tracciare l’andamento dello spazio percorso da in funzione del tempo.

**5. Esecuzione del programma.** Il programma fornisce un fotogramma ogni del moto della circonferenza che descrive una rotazione di , in un tempo complessivo di circa 0.314s, per un totale di più di 600 fotogrammi; in ciascun fotogramma compare la traiettoria descritta -fino a quell’istante- da .

In figura è riportato uno degli ultimi fotogrammi, in cui si può anche vedere un vettore che congiunge al centro di curvatura della cicloide in medesimo.





Il programma fornisce -per la rotazione di - l’andamento dello spazio percorso da in funzione del tempo. In figura l’output grafico.

Il programma fornisce in fine la risposta ai quesiti **1.4**, **1.5** fornendoli sullo schermo, come segue:

Per una rotazione di 3\*pi/4 della ruota si ha che:

le coordinate x,y di M sono rispettivamente 0.82454383 0.8535534

le componenti x,y della velocita' di M sono rispettivamente 25.606602 10.606602

le componenti x,y della accelerazione di M sono rispettivamente 318.1981 -318.1981

le coordinate x,y del centro di curv. della traiettoria di M sono risp. 1.5316507 -0.8535534

Per una rotazione di 2\*pi della ruota si ha che:

le coordinate x,y di M sono rispettivamente 3.1415927 0.

le componenti x,y della velocita' di M sono rispettivamente 0. 0.0000026226833

le componenti x,y della accelerazione di M sono rispettivamente 0.00007868051 450.00006

le coordinate x,y del centro di curv. della traiettoria di M sono risp. 3.1415927 -0.

**6. Soluzione grafica del punto 1.4.** La configurazione del sistema per una rotazione di si trova immediatamente considerando che il centro della circonferenza -stante la condizione di assenza di strisciamento- ha percorso uno spazio ; adottando allora una scala di rappresentazione delle distanze pari a

6.1)

si ha la configurazione riportata in figura, dove è indicata la posizione del punto . Si osservi allora quanto segue:

* il centro delle velocità all’istante considerato coincide con il punto di contatto fra la circonferenza e l’asse , essendo tale punto a velocità nulla per l’ipotesi di puro rotolamento;
* la polare mobile coincide con la circonferenza, essendo essa percorsa dal centro delle velocità, durante il moto;
* la polare fissa coincide con l’asse ;
* il centro di curvatura della polare mobile coincide con il centro della circonferenza;
* il centro di curvatura della polare fissa è il punto improprio dell’asse y.

Per individuare il centro di curvatura della traiettoria di in medesimo si utilizza ora una costruzione geometrica che si basa a sua volta sulla formula di Euler-Savary, che nel presente caso si scrive

6.2)

La costruzione passa attraverso i seguenti passaggi:

* si traccia la retta tra ;
* si traccia la retta per il punto ;
* si traccia la retta per ;
* si indica l’intersezione ;
* si traccia la retta per e per , ovvero -in questo caso- una retta verticale per .

Ebbene, l’intersezione identifica il punto . Leggendo il disegno e considerando la scala **6.1** si ha allora

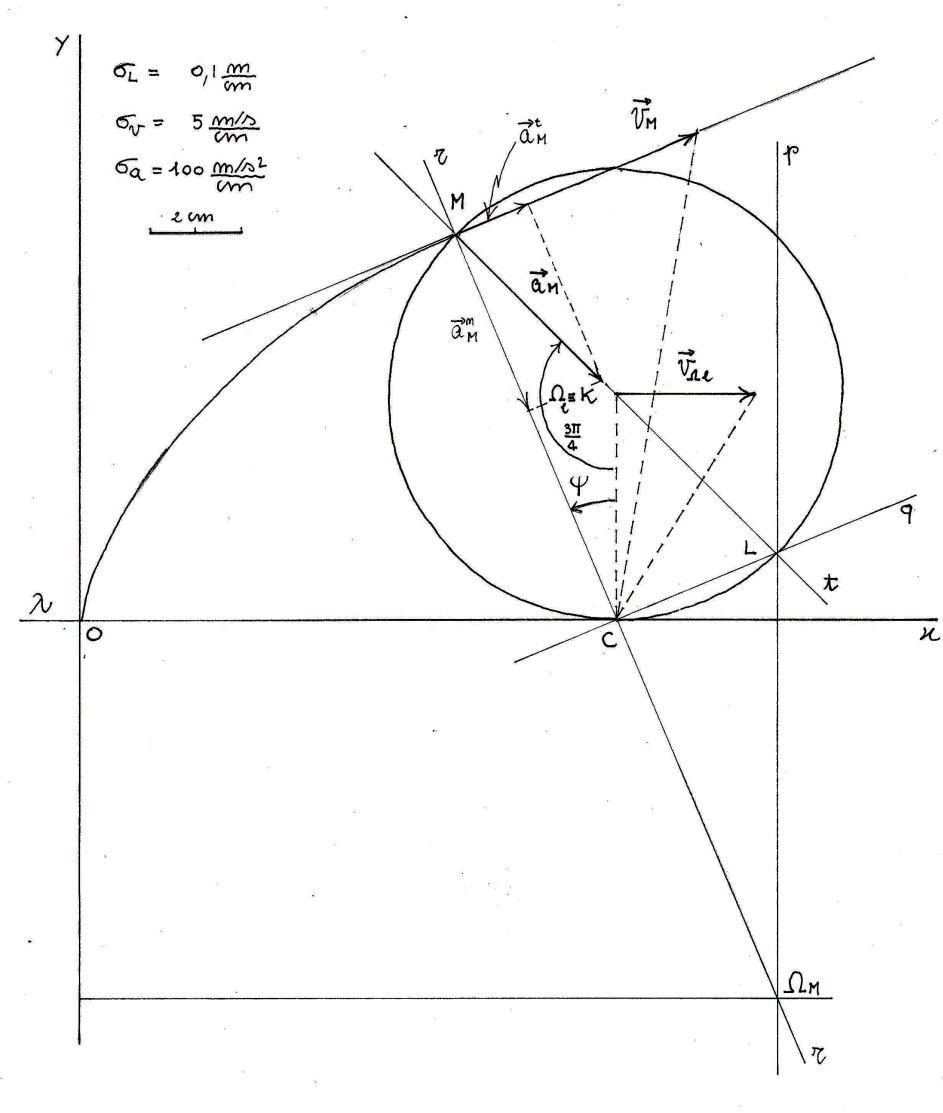
6.3)

che è in accordo con la **3.4**. Per la velocità del punto si consideri che anziché procedere con il metodo dei diagrammi polari si può invece fissare una scala di rappresentazione delle velocità, ad esempio

6.4)

quindi riportare sul disegno stesso -in scala- la velocità del centro della circonferenza, considerando che

6.5)



Dunque il vettore rappresentativo della **6.5** ha lunghezza ; tracciato tale vettore è immediato ricavare la velocità di qualunque altro punto solidale alla circonferenza, tenendo presente la struttura del campo delle velocità e la conoscenza del centro delle velocità . Allora la lettura del disegno porge

6.6)

in discreto accordo con la **3.2**. In conclusione poi - per l’accelerazione del punto tracciante- si consideri che

* il centro della circonferenza si muove di moto rettilineo uniforme, dunque ha accelerazione nulla e dunque coincide con il centro delle accelerazioni ;
* essendo la rotazione del piano mobile a velocità angolare costante, segue che l’accelerazione angolare è nulla e dunque l’accelerazione del generico punto del piano mobile punta il centro della circonferenza con modulo proporzionale alla distanza da tale centro, secondo il quadrato della velocità angolare.

Dunque si ha

6.7)

L’accelerazione **6.7** è riportata -secondo la scala indicata- sul disegno. Mentre il punto del piano mobile non ha accelerazione tangenziale, il punto inteso come punto che percorre la cicloide ha accelerazione normale e tangenziale, che possono essere lette direttamente dalla figura. Considerando la scala di rappresentazione si trova

6.8)

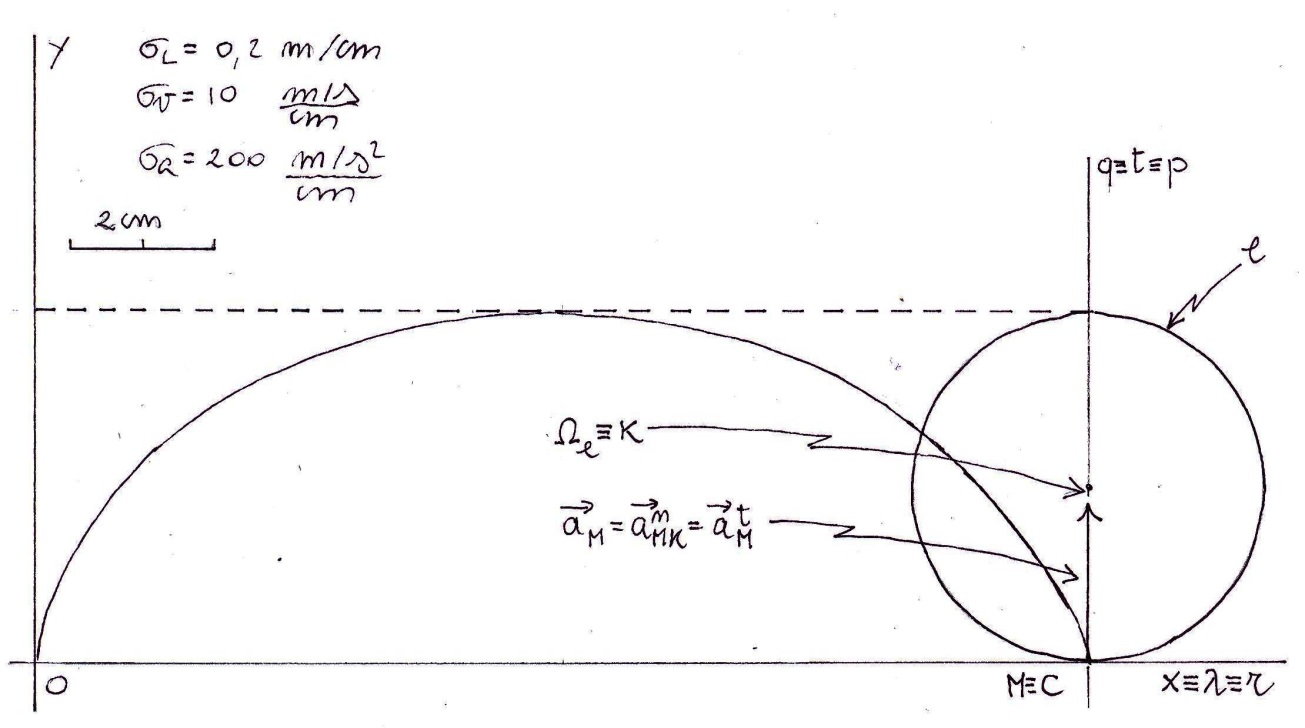
in discreto accordo con la corrispondente soluzione analitica **3.5**.

**7. Soluzione grafica del punto 1.5.** La configurazione del sistema per una rotazione di si trova procedendo come nel paragrafo precedente; adottando allora una scala di rappresentazione delle distanze pari a

7.1)

si ha la configurazione indicata in figura, dove si vede subito come in questo caso il punto tracciante coincida con il centro delle velocità. Dunque si ha immediatamente che

7.2)



Per l’accelerazione si consideri che essa punta il centro delle accelerazioni, essendo costante la velocità angolare; il centro delle accelerazioni poi coincide con il centro della circonferenza, che si muove di moto rettilineo uniforme. Il modulo della accelerazione ha in fine il valore già calcolato in **6.7**.

7.3)

Si osservi che in questo caso l’accelerazione di è tutta normale se inteso come punto del piano mobile, è invece tutta tangenziale se intesa come accelerazione del punto che traccia la cicloide.

In figura è riportato, secondo la scala indicata, un vettore rappresentativo dell’accelerazione del punto tracciante.

Il centro di curvatura non può essere individuato secondo la procedura grafica usata nel precedente paragrafo, poiché le rette in questo caso coincidono, e dunque non permettono di individuare una intersezione; tuttavia la **6.2** suggerisce che sia , quindi il centro di curvatura ha proprio le coordinate calcolate in **3.8**.

**8. Codice del programma principale.**

PROGRAM main\_ese\_9\_a

!usa DISLIN

USE DISLIN

!usa il modulo mod\_ese\_9\_a

USE mod\_ese\_9\_a

!sezione dichiarativa

IMPLICIT NONE

!dichiaro l'array che contiene le coordinate di M

REAL, DIMENSION (1:1000):: xM, yM

!dichiaro l'array che contiene le coordinate del centro di curvatura

REAL, DIMENSION (1:1000):: xc, yc

!dichiaro l'array che contiene i valori dell'ascissa curvilinea

REAL, DIMENSION (1:1000):: st

!dichiaro l'array che contiene gli istanti di tempo

REAL, DIMENSION (1:1000):: tempo = 0.

!dichiaro la variabile che contiene l'istante di tempo considerato

REAL:: t = 0.

!dichiaro l'incremento di tempo

REAL:: delta\_t = 0.0005

!dichiaro l'indice del ciclo

INTEGER:: i

!dichiaro posizione, velocità e accekerazione di M

REAL:: xemme, yemme, xemme\_p, yemme\_p, xemme\_p\_p, yemme\_p\_p

!dichiaro la posizione del centro di curvatura

REAL:: xcentro, ycentro

!dichiaro la stringa per chiudere il programma

CHARACTER(len=10):: chiusura

!sezione esecutiva

!Calcolo i valori delle coordinate di M e del centro di curvatura

i = 1

coordinate\_M: DO

xM(i) = raggio\*( omega\*tempo(i) - SIN(omega\*tempo(i)) )

yM(i) = raggio\*( 1. - COS(omega\*tempo(i)) )

xc(i) = raggio\*( omega\*tempo(i) + SIN(omega\*tempo(i)) )

yc(i) = (-1)\*raggio\*( 1. - COS(omega\*tempo(i)) )

CALL cicloide (xM, yM, xc, yc, tempo, i)

IF (omega\*tempo(i)>3.\*pi) EXIT coordinate\_M

tempo(i+1) = tempo(i) + delta\_t

i = i + 1

END DO coordinate\_M

!calcolo i valori dello spazio percorso

i = 1

spazio\_percorso: DO

st(i) = 4\*raggio\*( 1. - COS(omega\*tempo(i)/2.) )

condizione: IF (omega\*tempo(i)>2.\*pi) THEN

st(i) = 8\*raggio + 8\*raggio - 4\*raggio\*( 1. - COS(omega\*tempo(i)/2.) )

END IF condizione

IF (omega\*tempo(i)>3.\*pi) EXIT spazio\_percorso

tempo(i+1) = tempo(i) + delta\_t\*REAL(i)

i = i + 1

END DO spazio\_percorso

!traccia il diagramma dello spazio in funzione del tempo

CALL spazio (st, tempo, i)

!calcola posizione, velocita' e accelerazione di M per omega\*t=3\*pi/4

xemme = raggio\*( 3.\*pi/4. - SIN(3.\*pi/4.) )

yemme = raggio\*( 1. - COS(3.\*pi/4.) )

xemme\_p = raggio\*omega\*( 1. - COS(3.\*pi/4.) )

yemme\_p = raggio\*omega\*SIN(3.\*pi/4.)

xemme\_p\_p = raggio\*(omega\*\*2.)\*SIN(3.\*pi/4.)

yemme\_p\_p = raggio\*(omega\*\*2.)\*COS(3.\*pi/4.)

xcentro = raggio\*( 3.\*pi/4. + SIN(3.\*pi/4.) )

ycentro = (-1)\*raggio\*( 1. - COS(3.\*pi/4.) )

WRITE(\*,\*) " "

WRITE(\*,\*) "Per una rotazione di 3\*pi/4 della ruota si ha che:"

WRITE(\*,\*) " "

WRITE(\*,\*) "le coordinate x,y di M sono rispettivamente", xemme, yemme

WRITE(\*,\*) " "

WRITE(\*,\*) "le componenti x,y della velocita' di M sono rispettivamente", xemme\_p, yemme\_p

WRITE(\*,\*) " "

WRITE(\*,\*) "le componenti x,y della accelerazione di M sono rispettivamente", xemme\_p\_p, yemme\_p\_p

WRITE(\*,\*) " "

WRITE(\*,\*) "le coordinate x,y del centro di curv. della traiettoria di M sono risp.", xcentro, ycentro

WRITE(\*,\*) " "

!calcola posizione, velocita' e accelerazione di M per omega\*t=2.\*pi

xemme = raggio\*( 2.\*pi - SIN(2.\*pi) )

yemme = raggio\*( 1. - COS(2\*pi) )

xemme\_p = raggio\*omega\*( 1. - COS(2.\*pi) )

yemme\_p = raggio\*omega\*SIN(2.\*pi)

xemme\_p\_p = raggio\*(omega\*\*2.)\*SIN(2\*pi)

yemme\_p\_p = raggio\*(omega\*\*2.)\*COS(2\*pi)

xcentro = raggio\*( 2.\*pi + SIN(2.\*pi) )

ycentro = (-1)\*raggio\*( 1. - COS(2.\*pi) )

WRITE(\*,\*) " "

WRITE(\*,\*) "Per una rotazione di 2\*pi della ruota si ha che:"

WRITE(\*,\*) " "

WRITE(\*,\*) "le coordinate x,y di M sono rispettivamente", xemme, yemme

WRITE(\*,\*) " "

WRITE(\*,\*) "le componenti x,y della velocita' di M sono rispettivamente", xemme\_p, yemme\_p

WRITE(\*,\*) " "

WRITE(\*,\*) "le componenti x,y della accelerazione di M sono rispettivamente", xemme\_p\_p, yemme\_p\_p

WRITE(\*,\*) " "

WRITE(\*,\*) "le coordinate x,y del centro di curv. della traiettoria di M sono risp.", xcentro, ycentro

WRITE(\*,\*) " "

!le seguenti righe permettono di agevolare il lancio del programma dalla icona

WRITE (\*,\*)" "

WRITE (\*,\*)"To close the program press any key. Then press RETURN."

READ (\*,\*) chiusura

STOP

END PROGRAM main\_ese\_9\_a

**9. Codice del modulo.**

MODULE mod\_ese\_9\_a

!sezione dichiarativa

IMPLICIT NONE

!dichiaro le costanti geometriche del quadrilatero

REAL, PARAMETER:: omega = 30.

REAL, PARAMETER:: raggio = 0.5

REAL, PARAMETER:: pi = 3.14159265359

!scrivo le subroutine

CONTAINS

!------------------------------------------------------------------

SUBROUTINE cicloide (xM, yM, xc, yc, tempo, i)

!sezione dichiarativa

!dichiaro gli argomenti fittizi

!dichiaro gli array delle coordinate di M e del centro di curvatura

REAL,INTENT(IN),DIMENSION(1:1000):: xM, yM, xc, yc, tempo

!dichiaro il numero di istanti considerati

INTEGER,INTENT(IN):: i

!dichiaro le variabili locali

!coordinate centro cerchio

REAL:: xcerchio, ycerchio

!stringa usata per la legenda

CHARACTER(len=30)::stringa

!sezione esecutiva

!imposto il tipo di file

CALL METAFL ('bmp') !indico il formato dell'output

CALL BMPMOD (25000,'meter','resolution') !risoluzioneformato .bmp

!imposto la pagina

CALL SCRMOD ('revers') !scritta nera su fondo bianco

CALL DISINI !richaima alcune impostazioni di default

CALL PAGERA !traccio un bordo per il piano xy

CALL DUPLX !font a doppio spessore

!imposto gli assi

CALL AXSPOS (700,2000) !coordinate angolo basso sinistra

CALL AXSLEN (1700,1700) !lunghezza dei due assi in pixel

CALL NAME (' ','x') !nome delle ascisse

CALL NAME (' ','y') !nome delle ordinate

CALL GRAF (-0.1,4.9,-0.1,0.5,-2.5,2.5,-2.5,0.25) !inizio, fine, incremento assi x,y

CALL GRID (1,1) !impone una griglia sul piano coordinato

CALL XAXGIT !traccio la retta x=0

CALL YAxGIT !traccio la retta y=0

CALL DASH !tratteggio per gli assi coordinati

CALL NAME ('metri','x') !nome delle ascisse

CALL NAME ('metri','y') !nome delle ordinate

!imposto il titolo

CALL TITLIN ("Cicloide",1) !prima riga del titolo

CALL TITLE !stampa il titolo di cui sopra

!traccio le curve

CALL CHNCRV ('line') !usa tratti diversi per le tre curve

CALL CURVE (xM, yM, i) !traccio la cicloide

CALL CURVE (xc, yc, i) !traccio la traiettoria del centro di curvatura

!ricavo il centro del cerchio nell'ultima posizione

xcerchio = omega\*tempo(i)\*raggio

ycerchio = raggio

!traccio la circonferenza nell'ultima posizione

CALL RLCIRC (xcerchio, ycerchio, raggio)

!traccio un vettore che congiunge M al centro di curvatura

CALL RLVEC (xM(i), yM(i), xc(i), yc(i), 2211)

!imposto la legenda

CALL LEGINI (stringa,2,10) !variabile di carattere, righe, lunghezza

CALL LEGTIT ('Legenda') !titolo legenda

CALL LEGLIN (stringa,'cicloide',1) !prima riga della legenda

CALL LEGLIN (stringa,'omega\_M',2) !seconda riga della legenda

CALL LEGEND (stringa,3) !posizine in alto a destra

CALL DISFIN

END SUBROUTINE cicloide

!--------------------------------------------------------------------

SUBROUTINE spazio (st, tempo, i)

!sezione dichiarativa

!dichiaro gli argomenti fittizi

!dichiaro l'array dello spazio e quello del tempo

REAL,INTENT(IN),DIMENSION(1:1000):: st, tempo

!dichiaro il numero di istanti considerati

INTEGER,INTENT(IN):: i

!dichiaro le variabili locali

!stringa usata per la legenda

CHARACTER(len=30)::stringa

!sezione esecutiva

!imposto il formato del file

CALL METAFL ('bmp') !indico il formato dell'output

CALL BMPMOD (25000,'meter','resolution') !risoluzione del formato .bmp

!imposto la pagina

CALL SCRMOD ('revers') !scritta nera su fondo bianco

CALL DISINI !richiama alcune impostazioni di default

CALL PAGERA !traccio un bordo per il piano xy

CALL DUPLX !font a doppio spessore

!imposto gli assi x,y

CALL AXSPOS (700,2700) !coordinate angolo basso sinistra

CALL AXSLEN (1700,1300) !lunghezza dei due assi in pixel

CALL NAME ('secondi','x') !nome delle ascisse

CALL NAME ('metri','y') !nome delle ordinate

CALL LABDIG (3,'x') !chiedo 3 cifre decimali per l'asse x

CALL DASH !tratteggio per gli assi coordinati

CALL GRAF (0.0,0.3,0.0,0.05,0.0,6.,0.0,0.5) !inizio, fine, intervallo assi x, y

CALL GRID (1,1) !impone una griglia sul piano coordinato

CALL XAXGIT !traccio la retta x=0

CALL YAxGIT !traccio la retta y=0

!traccio la curva

CALL MYLINE (1,1) !chiede una linea continua per la curva

CALL CURVE (tempo, st, i) !traccio lo spazio in funzione del tempo

!imposto il titolo

CALL TITLIN ("Spazio percorso",1) !prima riga del titolo

CALL TITLE !stampa il titolo di cui sopra

!imposto la legenda

CALL LEGINI (stringa,1,15) !variabile di carattere, righe, lunghezza

CALL LEGLIN (stringa,'spazio',1) !contenuto della legenda

CALL LEGTIT ('Legenda') !titolo legenda

CALL LEGEND (stringa,3) !posizine in alto a destra

CALL DISFIN

END SUBROUTINE spazio

!---------------------------------------------------------------------------

END MODULE mod\_ese\_9\_a

1. Si osservi che i passaggi successivi ammettono implicitamente che il seno integrando sia positivo, ovvero che la rotazione sia entro l’angolo giro. Per rotazioni ulteriori questa espressione non va più bene. [↑](#footnote-ref-1)
2. Si osservi che vale qui quanto detto nella nota uno: poiché l’espressione dello spazio percorso nel tempo vale solo per rotazioni inferiori all’angolo giro, anche queste espressioni, ottenute per derivazione dalla **2.4** sono soggette allo stesso ambito di validità. [↑](#footnote-ref-2)