

Quindi il vettore di Lenz $\vec{e}$ è da la direzione del pericentro dell'orbita, il suo modulo dà l'eccentricità, il vettore di Lenz è nullo quando l'orbita è circolare e perciò il pericentro non è definito.

Usando la (1.27) si trova la (1.25):

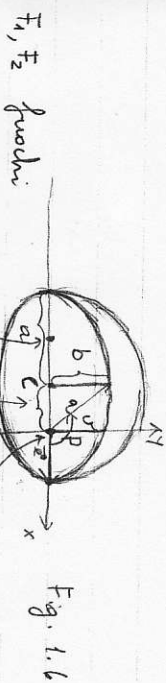
$$r = \frac{a(1-e^2)}{1+e\cos\theta} \quad (1.28)$$

(vedi Fig. 1.5) e circolari.

- 1) $e=0$ $r=a = \frac{j^2}{GM}$ circonferenza
- 2) $0 < e < 1$ $r_{min} = a(1-e)$
 $r_{max} = a(1+e)$

Ellisse con semiasse maggiore $a = \frac{j^2}{GM(1-e^2)}$

e semiasse minore $b = a\sqrt{1-e^2}$ (vedi Fig. 1.6) (1.29)



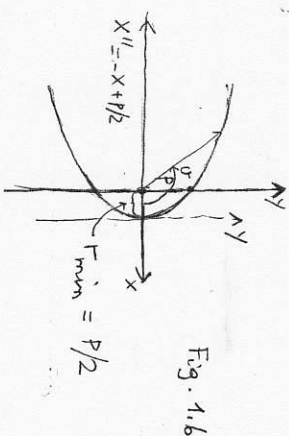
$b = a\sqrt{1-e^2}$ distanza del fuoco dal centro di simmetria

- Notare che l'ellisse si ottiene dalla circonferenza di centro C e raggio a abbassando tutti i punti della circonferenza (lungo la perpendicolare all'asse x) di una frazione $\sqrt{1-e^2}$.
- L'ellisse è il luogo dei punti la cui distanza dai fuochi (F_1 ed F_2) è costante ed uguale all'asse maggiore $2a$.

- 3) $e=1$ (a infinito)

parabola, con parametro P, di equazione:

$$y^2 = p^2 - 2px \quad (1.30)$$



Area X asse di simmetria, origine degli assi nel fuoco; vertice in $(r_{min} = p/2, 0)$. Se spostato l'origine nel vertice: $x \rightarrow x' = x - p/2$, la (1.30) diventa:

$$y^2 = -2px' \quad \text{e poi } x' \rightarrow x'' = -x' : \quad x'' = \frac{1}{2p} y^2$$

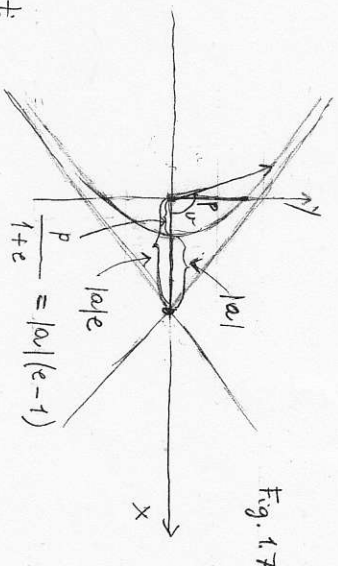
- 4) $e > 1$, a negativo

$$p = a(1-e^2) \quad (p = j^2/GM > 0)$$

$$1+e\cos\theta = \frac{p}{r}$$

$$r = \text{infinito} \quad 1+e\cos\theta = 0$$

$$\cos\theta_0 = -1/e \quad \text{da qui asintoti dell'iperbole}$$



Si noti che $\vec{e}$ giace nel piano individuato da $\vec{r}, \dot{\vec{r}}$ (che inglobano essere fuso - PIANO ORBITALE); quindi, dato che $\vec{e}$ rappresenta una direzione fissa nel piano orbitale conviene cominciare a contare gli angoli da questa direzione.

Chiamo ν "ANOMALIA VERA" l'angolo ($\nu = \nu(t)$ ovviamente) del vettore posizione rispetto alla direzione $\vec{e}$:

$$\cos \nu = \frac{\vec{e} \cdot \vec{r}}{e} \quad (1.23)$$

$$e \cos \nu = \vec{e} \cdot \vec{r} = \frac{1}{GM} (\dot{\vec{r}} \times \vec{j}) \cdot \vec{e} - 1 =$$

$$= \frac{1}{GM} r^3 \dot{\theta}^2 - 1 = \frac{j^2}{GM r} - 1 \quad (1.24)$$

$\nearrow$ da (1.18') $\nearrow$ da (1.15b)

Quindi:

$$r = \frac{j^2 / GM}{1 + e \cos \nu} = \frac{P}{1 + e \cos \nu} \quad (1.25)$$

La (1.25) fornisce r in funzione di ν (ad ogni t, j, M, j, e sono tutte costanti) e l'equazione dell'orbita

La (1.25) è l'eq. di una conica in coordinate polari (polo nel fuoco della conica e origine dell'angolo ν a partire dal vettore $\vec{e}$). Per vederlo posso passare in coordinate cartesiane:

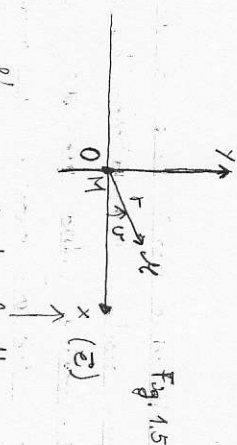
$$x = r \cos \nu$$

$$r + e r \cos \nu = P$$

$$(x^2 + y^2)^{1/2} = P - ex$$

$$x^2 + y^2 = (P - ex)^2 \quad (1.26)$$

La (1.26) è la (1.25) in coordinate cartesiane (definite in Fig. 1.5) ed è chiaramente una conica. P è il semiasse retto della conica, cioè il valore di r per $\nu = \pi/2$ ($\nu = \pi/2, \cos \nu = 0, y = 0, r = P$).



L'asse x coincide col vettore di Lenz $\vec{e}$ (dove si trova M) e il fuoco della conica.

$$2a = r(\nu=0) + r(\nu=\pi) = \frac{P}{1+e} + \frac{P}{1-e} = \frac{2P}{1-e^2}$$

$$a = P / (1 - e^2) \quad (1.27)$$

$a \equiv$ semiasse maggiore della conica

$0 < e < 1 \Rightarrow a$ positivo ($e=0, r=P=a$ circonferenza)

$e=1 \Rightarrow a$ infinito

$e > 1 \Rightarrow a$ negativo

Il punto di minima distanza da O - pericentro - ha $\nu=0$:

$$r_{\min} = \frac{P}{1+e} = \frac{a(1-e^2)}{1+e} = a(1-e) \quad (1.28)$$

(Se $e=0$ $r=P=a$ ed il pericentro non è definito)

Concludo che:

Le soluzioni del problema di 2 corpi (punti fermi) in orbite gravitazionali sono dati da equazioni

$$r = \frac{p}{1 + e \cos v} \quad (1.31)$$

dove i parametri p, e sono determinati in funzione delle condizioni iniziali $\vec{r}(0), \dot{\vec{r}}(0)$:

$$p = \frac{h^2}{G(m_1 + m_2)} \quad \vec{h} = \vec{r}(0) \times \dot{\vec{r}}(0) \quad (1.32)$$

$$e = \left| \frac{1}{G(m_1 + m_2)} \dot{\vec{r}}(0) \times \vec{h} - \vec{e}_r(0) \right| \quad (1.33)$$

La distanza $\vec{e}$ fornisce l'asse di simmetria (asse maggiore) della conica ($e=0$ circonferenza); l'anomalia vera v misura a partire da $\vec{e}$ (in senso antiorario); la velocità angolare $\dot{\theta}$ in ogni punto della conica si ottiene da (1.15b) e (1.31):

$$\dot{\theta} = \frac{d}{dt} = \frac{h}{r^2} (1 + e \cos v)^2 \quad (1.34)$$

NOTA Dall'orbita della massa ridotta M si può ricostruire le orbite delle masse m_1 ed m_2 rispetto al loro comune centro di massa $\vec{R}$ definito dalla (1.1). Basta ricordare che se l'origine è in $\vec{R}$, m_1 ed m_2 sono sempre allineate con l'origine, da parti opposte rispetto ad essa e con velocità antiparallele (le quantità di moto dei due corpi sono uguali ed opposte)

§1.2 Velocità lineare ed integrali dell'energia

Poiché il sistema è isolato l'energia totale si conserva:

$$E = \frac{1}{2} M v^2 - \frac{G M M}{r} \quad (2.1)$$

$$M = \frac{m_1 m_2}{m_1 + m_2} \quad (m_1 + m_2) = m_1 m_2$$

$$v^2 = \dot{\vec{r}} \cdot \dot{\vec{r}} = \dot{r}^2 + r^2 \dot{\theta}^2$$

$$\vec{r} = \vec{r}_2 - \vec{r}_1 \quad (\text{Fig. 1.1})$$

$$E = \frac{1}{2} v^2 - \frac{G(m_1 + m_2)}{r}$$

Energia totale per unità di massa ridotta

$$(\dot{\theta} = \dot{v}; \dot{\theta} \text{ e } v \text{ di fluirono solo in una costante}) \quad (2.2)$$

Ritorno la relazione tra energia ed elementi dell'orbita, dalla (1.16) usando la (1.18):

$$\begin{aligned} e^2 &= \left[\frac{1}{GM} (r^3 \ddot{\theta} \vec{e}_r - r^2 \dot{\theta}^2 \vec{e}_\theta) - \vec{e}_r \right] \cdot \left[\frac{1}{GM} (r^3 \ddot{\theta} \vec{e}_r - r^2 \dot{\theta}^2 \vec{e}_\theta) - \vec{e}_r \right] = \\ &= \frac{1}{G^2 M^2} (r^6 \dot{\theta}^4 + r^4 \dot{\theta}^2 \ddot{\theta}^2) - \frac{2}{GM} r^3 \dot{\theta}^2 + 1 = \\ &= 1 + \frac{r^4}{G^2 M^2} \frac{d^2}{dt^2} (r^3 \dot{\theta}^2 + \dot{r}^2) - \frac{2}{GM} r^3 \frac{d^2}{dt^2} = \\ &= 1 + \frac{d^2}{dt^2} \frac{r^2}{G^2 M^2} - \frac{2 \dot{r}^2}{GM r} = 1 + \frac{2 \dot{r}^2}{G^2 M^2} \left(\frac{1}{2} v^2 - \frac{GM}{r} \right) \end{aligned} \quad (\dot{\theta} = \frac{d}{dt})$$

$$\begin{aligned} \text{cioè:} \\ e^2 &= 1 + \frac{2 E^2}{G^2 M^2} \end{aligned} \quad (2.4)$$

La (2.4) lega energia e momento angolare (per unità di

massa ridotta) con l'eventualità. Dallo stesso, la (1.25) e la (1.27) danno:

$$j^2 = GMa(1-e^2) \quad (2.5)$$

Sostituendo la (2.5) nella (2.4) ottengo:

$$E = - \frac{GM}{2a} \quad (2.6)$$

NOTA: la (2.6) dice che l'energia totale fra una degli elementi orbitali, il semiasse maggiore la (2.6) e la (2.5) insieme danno che, fissati E e j sono fissati due elementi orbitali, cioè il semiasse maggiore e l'eventualità.

Sostituendo la (2.6) nella (2.3) ottengo v^2 in funzione di r :

$$v^2 = GM \left(\frac{2}{r} - \frac{1}{a} \right) \quad (2.7)$$

→ fissata E , e quindi a , la (2.7) fornisce il modulo della velocità lineare per ogni valore di r .

la classificazione delle orbite segue dalla (2.6) e (2.4):

$E > 0$ ($a < 0$), $e > 1$	ORBITA IPERBOLICHE
$E = 0$ ($a \rightarrow \infty$), $e = 1$	ORBITA PARABOLICHE
$E < 0$ ($a > 0$), $0 \leq e < 1$	ORBITA ELLITTICHE (o CIRCOLARI)

Un altro modo ancora per vedere la classificazione delle orbite è quello di usare la (1.15b) ($\dot{\theta} = j/r^2$) per eliminare $\dot{\theta}$ dalla (2.3) e ridurre il problema al caso unidimensionale con variabile $r, \dot{r}$.

$$E = \frac{1}{2} v^2 - \frac{GM}{r} = \frac{1}{2} (\dot{r}^2 + r^2 \dot{\theta}^2) - \frac{GM}{r} = \frac{1}{2} \dot{r}^2 + \left(\frac{j^2}{2r^2} - \frac{GM}{r} \right) = \frac{1}{2} \dot{r}^2 + \mathcal{V}(r) = E(r, \dot{r}) \quad (2.9)$$

$$\mathcal{V}(r) = \frac{j^2}{2r^2} - \frac{GM}{r} \quad (2.10)$$

→ è il nuovo potenziale del problema ridotto al caso unidimensionale (si chiama anche "potenziale effettivo").

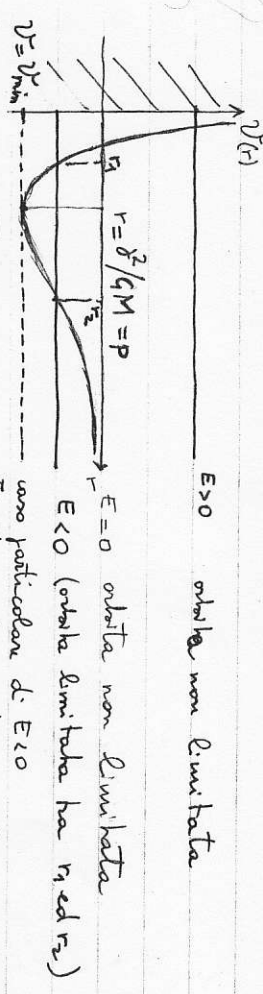


Fig. 2.1

di massa ridotta) si possono avere orbite limitate purché sia

$0 > E \geq \mathcal{V}_{\min}$; in particolare, il minimo valore possibile di E - cioè $E = \mathcal{V}_{\min} = -\frac{1}{2} \frac{GM^2}{j^2}$ - corrisponde all'orbita circolare con quel j .

Il caso $E = 0$ ed $E > 0$ danno entrambi orbite non limitate, paraboliche ed iperboliche rispettivamente; una importante differenza fra le due è il valore della velocità all'infinito. Dalla (2.7):

$$v^2(\infty) = -\frac{GM}{a} = 2E$$

(2.11)

per $v^2(\infty) = 0$ per l'orbita parabolica e $v^2(\infty) > 0$ per l'orbita iperbolica.

§13. Legge area ed equazione di Keplero

da (1.31)

$$r = \frac{p}{1 + e \cos \nu}$$

$$p = \ell^2 / GM = a(1 - e^2)$$

(3.1)

è l'eq. della traiettoria, dalla (3.1), se conosciamo $r = r(t)$, conosciamo anche la legge area, cioè il modo in cui l'orbita viene percorsa.

Dalla (1.15b)

$$\dot{\theta} = \dot{\nu} = \frac{d}{r^2}$$

(3.2)

$$\dot{\nu} = \frac{d(1 + e \cos \nu)^2}{p^2}$$

(3.3)

Invece di integrare la (3.3) prendo nel modo seguente. Definisco $dh(t)$

$$dh(t) = \frac{1}{2} \int_{t_0}^t r^2(t) \dot{\nu}(t) dt$$

(3.4)

$$h(t) = \frac{1}{2} r^2 \dot{\nu}$$

(3.5)

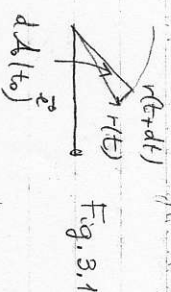
↑
velocità areolare $\equiv$ area spazzata dal raggio vettore nell'unità di tempo

Per la (3.2)

$$\dot{h}(t) = \frac{1}{2} \ell$$

(3.6)

La (3.6) dice che la velocità areolare è costante. Il raggio vettore spazza aree uguali in tempi uguali - 2° legge di Keplero.



ovvero:

$$r = a(1 - e \cos u) \quad \left(\text{da } r = \frac{a(1-e^2)}{1+e \cos u} \right) \quad (3.14)$$

$$\cos u = \frac{1 - e \cos u}{1 - e \cos u} \quad (3.15)$$

$$\sin u = \frac{\sqrt{1-e^2} \sin u}{1 - e \cos u} \quad (3.16)$$

Notare che la (3.14) è l'equazione dell'orbita nella variabile u (anziché r, u come nella (3.1)).

→ $u = v$ solo per multipli di π .

Ora uso la 2^a legge di Keplero (legge delle aree), più alcune semplici considerazioni geometriche, per trovare la relazione tra u (e quindi v) ed t (e quindi t).

$$\text{per la (3.15), (3.16)} \quad \text{per la (3.12)}$$

Per la legge delle aree (e si veda Fig. 3.2):

$$\frac{\text{area SPA}}{\text{area dell'ellisse}} = \frac{t - t_0}{P}$$

$$\text{area SPA} = \pi a b (t - t_0) / P = \frac{\pi a b t}{n P} = \frac{1}{2} a b t = \frac{1}{2} a^2 \sqrt{1-e^2} t \quad (3.17)$$

per la (3.12)

D'altra parte, guardando la Fig. 3.2 posso dire che:

$$\text{area SPA} = \text{area SPR} + \text{area RPA} = \text{area SPR} + \sqrt{1-e^2} \text{area QRA} \quad (3.18)$$

Infatti:

$$\text{area RPA} = \sqrt{1-e^2} \text{area QRA}$$

per la relazione di affinità ha il centro e l'ellisse (dividete l'area in sfere parallele all'asse minore dell'ellisse e calcolatela per integrazione)

Allora:

$$\begin{aligned} \text{area SPA} &= \text{area SPR} + \sqrt{1-e^2} (\text{area QCA} - \text{area QCR}) = \\ &= \frac{1}{2} r^2 \cos u \sin u + \sqrt{1-e^2} \left(\frac{1}{2} a^2 u - \frac{1}{2} a^2 \cos u \sin u \right) = \end{aligned}$$

$$= \frac{1}{2} (r \cos u)(r \sin u) + \sqrt{1-e^2} \left(\frac{1}{2} a^2 u - \frac{1}{2} a^2 \cos u \sin u \right) =$$

$$\begin{aligned} &\Rightarrow \frac{1}{2} a^2 (\cos u - e) \sqrt{1-e^2} \sin u + \frac{1}{2} \sqrt{1-e^2} a^2 u - \frac{1}{2} \sqrt{1-e^2} a^2 \cos u \sin u = \\ &= -\frac{1}{2} a^2 e \sqrt{1-e^2} \sin u + \frac{1}{2} a^2 u \sqrt{1-e^2} = \end{aligned}$$

$$= \frac{1}{2} a^2 \sqrt{1-e^2} (u - e \sin u) \quad (3.20)$$

Uguaglio la (3.20) alla (3.17) ed ho:

$$\frac{n(t-t_0)}{2} = u - e \sin u \quad \text{EQUAZIONE DI KEPLERO} \quad (3.21)$$

L'eq. di Keplero (3.21) dà l'anomalia media in funzione dell'anomalia eccentrica ($t = t(e, u)$) ma non ha soluzione analitica esplicita (per $e > 0$)

$$u = u(t, e) \quad (3.22)$$

Dalle condizioni iniziali: $\vec{r}(0) = \vec{r}_1(0) - \vec{r}_2(0)$, $\dot{\vec{r}}(0) = \dot{\vec{r}}_2(0) - \dot{\vec{r}}_1(0)$, calcolo: $\vec{r} = \vec{r}(0) \times \frac{\vec{r}(0)}{r(0)}$

$$\vec{e} = \frac{1}{GM} \dot{\vec{r}}(0) \times \vec{r}(0) \quad (3.33)$$

Calcolo. L'integrale dell'energia per unità di massa ridotta:

$$E = \frac{1}{2} \dot{\vec{r}}(0) \cdot \dot{\vec{r}}(0) - \frac{GM}{r(0)} \quad \text{IMPORTANTE controllo da cui} \quad (3.34)$$

da cui ottengo il semiasse maggiore dell'orbita

$$a = \frac{GM}{(-2E)} \quad (3.35)$$

Dalla (3.33) ho ovviamente l'eccentricità e controllo che sia $e < 1$ (3.36)

Dalla (3.35) ho il mt, medio e il periodo (3^a legge di Keplero):

$$n = \frac{(-2E)^{3/2}}{GM} \quad P = \frac{2\pi}{n} \quad (3.37)$$

Dalla (3.14) calcolo l'anomalia eccentrica all'istante iniziale:

$$\cos u(0) = \frac{1}{e} \left(1 - \frac{r(0)}{a} \right) \quad (3.38)$$

Ma mi serve $u(0)$, per dedurre in quale quadrante sono ho bisogno anche di $\sin u(0)$. Lo ottengo derivando la (3.14):

$$\dot{r} = a e \sin u \quad (\dot{r} \text{ è la velocità radiale. In particolare: } \dot{r}(0) = \dot{\vec{r}}(0) \cdot \frac{\vec{r}(0)}{r(0)})$$

e la calcolo più avanti (vedi la 3.43). Quindi ho

$$\sin u(0) = \frac{r(0) \dot{r}(0)}{a e n} \quad (3.39)$$

Avendo $u(0)$, uso l'eq. di Keplero (3.21) per calcolare $t(0)$

tenendo conto che nella (3.21) t_0 è il tempo del passaggio al pericentro (che ora diamo t_p per evitare confusione):

$$t(t=0) = n(t=0 - t_p) = u(t=0) - e \sin u(t=0) \quad (3.40)$$

da cui

$$t_p = -\frac{t(0)}{n} \quad (3.41)$$

Adesso calcolo, sempre per $t=0$, le coordinate cartesiane della massa ridotta nel sistema di riferimento d.

Fig. 3.3:

$\hat{e}, \hat{e}_1, \hat{r}$ sono 3 vettori che formano una terna mista.

Fig. 3.3

$\hat{r} = \vec{r}/r$ $\hat{e} = \vec{e}/e$

$\hat{e}_1 = \hat{r} \times \hat{e}$ (3.42)

In realtà so che il moto si svolge nel piano $\hat{e}, \hat{e}_1$.

Fig. 3.4

$x = r \cos u$
 $y = r \sin u$
 $\hat{e} \equiv x$

Adesso le coordinate cartesiane x, y (vedi Fig. 3.4) al tempo $t=0$ sono (uso la (3.13)):

$$\begin{cases} x(0) = a \cos u(0) - ae \\ y(0) = a \sqrt{1-e^2} \sin u(0) \end{cases} \quad (3.43)$$

Ne posso ricavare naturalmente la velocità delle coordinate
costante (e magari anche la velocità) ad un tempo
 $t_1 > t=0$

Allora:

$$x(t_1) = r(t_1 - t_p) \quad (3.44)$$

Dall'eq. di Keplero:

$$x(t_1) = u(t_1) - e \sin u(t_1) \quad (3.45)$$

posso ricavare $u(t_1)$ (vedi p. 3.7 per la riduzione dell'eq. di Keplero). E quindi:

$$\begin{cases} x(t_1) = a \cos u(t_1) - a e \\ y(t_1) = a \sqrt{1-e^2} \sin u(t_1) \end{cases} \quad (3.46)$$

$$r(t_1) = \left(x^2(t_1) + y^2(t_1) \right)^{1/2} \quad (3.47)$$

Se voglio anche la velocità devo calcolarmi $\dot{u}$ in funzione di t . Leggo la (3.1) e la (3.14):

$$(1 - e \cos u)(1 + e \cos u) = 1 - e^2 \quad (3.48)$$

Derivo rispetto al tempo:

$$\dot{u} = \dot{r} \frac{a \sin u (1 - e \cos u)}{a \sin u (1 + e \cos u)}$$

Uso:

$$\dot{r} = \frac{b}{r^2} = \frac{b}{a^2 (1 - e \cos u)^2}$$

$$1 + e \cos u = \frac{a(1 - e^2)}{r} = \frac{a(1 - e^2)}{a(1 - e \cos u)}$$

$$\sin u = \frac{a}{r} \sqrt{1 - e^2} \sin u = \frac{a \sqrt{1 - e^2} \sin u}{a(1 - e \cos u)}$$

ed ottengo:

$$\dot{u} = \frac{b}{a^2 (1 - e \cos u)^2} \frac{1}{\sin u} \frac{\sqrt{1 - e^2} \sin u}{(1 - e \cos u)} \frac{(1 - e \cos u)^2}{(1 - e^2)}$$

$$\begin{aligned} \dot{u} &= \frac{b}{a^2} \frac{1}{\sqrt{1 - e^2} (1 - e \cos u)} = \frac{b}{a r \sqrt{1 - e^2}} = \frac{\sqrt{GM a (1 - e^2)}}{a \sqrt{1 - e^2} r} = \\ &= \frac{(GM)^{1/2}}{a^{3/2}} \frac{1}{r} = \frac{(GM)^{1/2}}{a^{3/2}} \frac{a}{r} = n \frac{a}{r} \end{aligned} \quad (3.49)$$

$$\dot{u}(t_1) = n \frac{a}{r(t_1)} \quad (3.50)$$

E quindi:

$$\begin{cases} \dot{x}(t_1) = -a \dot{u}(t_1) \sin u(t_1) \\ \dot{y}(t_1) = a \sqrt{1 - e^2} \dot{u}(t_1) \cos u(t_1) \end{cases} \quad (3.51)$$

Notare però che con ho ottenuto posizione e velocità per ogni tempo $t_1 > t_0 = 0$ nel riferimento di Fig. 3.3 in cui il piano x, y coincide con il piano orbitale e l'asse x con la direzione del pericentro. Per avere le corrispondenti posizioni e velocità nel riferimento inerziale di partenza (Fig. 3.2) devo eseguire una rotazione (corrispondente al moto uniforme del centro di massa) e poi una rotazione corrispondente all'orientazione dell'orbita nello spazio. Nel § 1.4 discutereò appunto l'orientazione dell'orbita nello spazio e finirà di risolvere il problema dei due corpi.

Ora mostro come si risolve l'eq. di Keplero con il metodo di Newton.

Il metodo di Newton serve per risolvere una eq. del tipo:
 $f(x) = 0$ (3.52)
 cioè per trovare gli zeri della $f(x)$ in un certo intervallo della variabile indipendente x . La $f(x)$ deve essere continua e non cambiare di convessità nell'intervallo considerato.

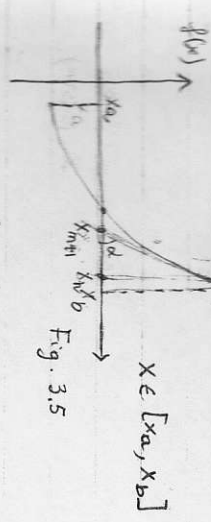


Fig. 3.5

$$tgr = f'(x_n) = \frac{f(x_n)}{x_n - x_{n+1}}$$

$$x_n - x_{n+1} = \frac{f(x_n)}{f'(x_n)}$$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (3.53)$$

Il metodo converge alla soluzione in pochissimi passi; ricordarsi di controllare che $f''(x)$ non cambia segno in $x \in [x_a, x_b]$; ricordarsi di partire con il valore di x in il quale la derivata prima è massima (altrimenti si rischia di finire fuori dall'intervallo considerato).

Ora applico il metodo di Newton per risolvere l'eq. di Keplero (3.21).
 Trovo il valore di u che in un dato valore dell'anomalia media l risolve l'equazione; (e finito dalle condizioni iniziali)

$$f(u) = u - e \sin u - l = 0$$

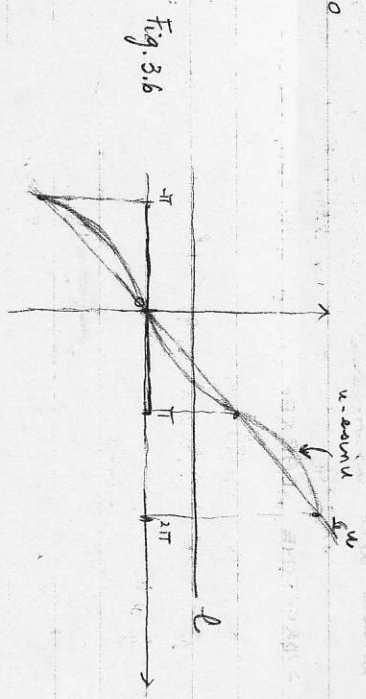


Fig. 3.6

Qualunque sia il valore dato di l , due ricordando ad una l due intervalli $0, \pi$ oppure $\pi, 2\pi$ altrimenti nessuno in problemi di cambiamento di segno della derivata seconda; inoltre deve sapere in quali dei due intervalli ho fatto, in decidere da dove partire (soggettivamente da π e da $-\pi$). Una volta fatto questo uso la formula iterativa

$$u_{n+1} = u_n - \frac{u_n - e \sin u_n - l}{1 - e \cos u_n} \quad (3.54)$$

e in pochi passi arrivo alla soluzione cercata in quel dato intervallo.

Calcolo dell'anomalia eccentrica in CORKEP

SUBROUTINE CORKEP (r, rp, gm, eps, ek, n)

(vedi programma Kepler9)

$$l = u - e \sin u \quad (\text{eq. Kepler}) \quad (1)$$

$$\begin{aligned} l &= u - e \sin u \\ \Rightarrow u &= \frac{n}{1 - e \cos u} = n \frac{a}{r} \quad (2) \end{aligned}$$

$$r = a(1 - e \cos u) \quad \text{eq. orbita} \quad (3)$$

$$\cos u = \left(1 - \frac{r}{a}\right) / e \quad (4)$$

Derivando la (3):

$$\dot{r} = ae u \sin u$$

$$\begin{aligned} \sin u &= \frac{\dot{r}}{ae u} = \frac{\dot{r}}{ae \left(\frac{n}{1 - e \cos u}\right)} = \frac{\dot{r} \cdot r}{na^2 e} = \frac{(\dot{r} \cdot \frac{r}{r})}{na^2 e} = \frac{\dot{r} \cdot \frac{r}{r}}{na^2 e} \end{aligned}$$

componenti radiali

($\dot{r}$ è il modulo della velocità, che è data in INPUT dal vettore $\dot{\mathbf{r}}$ moltiplicato opportunamente per il verso del vettore $\mathbf{r}$)

§1.4 L'orbita nello spazio

Considero un sistema di riferimento iniziale $Oxyz$ nella cui origine immagino la sia concentrata tutta la massa $m_1 + m_2$

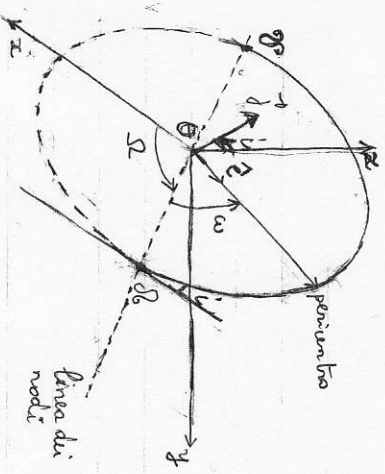


Fig. 4.1

(case $e < 1$)

x, y, z sono gli assi coordinati.

Nella Fig. 4.1 il piano orbitale, individuato dal vettore di Lenz $\mathbf{e}$ e dal vettore $\mathbf{r}$ della massa ridotta M (purché non sia zero al pericentro e all'apocentro, nel qual caso uso $\dot{\mathbf{r}}(t)$), taglia il piano xy lungo una retta detta linea dei nodi, e forma col piano xy un angolo diechmo i (inclinazione)

L'orbita incontra la linea dei nodi in 2 punti: nodo ascendente $\mathcal{D}$ (quando il corpo $\mathcal{C}$ attraversa il piano xy dal basso verso l'alto - l'alto essendo definito dalle z positive), e nodo discendente.

L'angolo della linea $O\mathcal{D}$ con l'asse x si chiama longitudine del nodo ascendente. ($0 \leq \Omega < 2\pi$).

Dati i ed Ω è fissato il piano orbitale, ma se l'orbita non è circolare bisogna ancora fissare l'orientazione, e cioè il pericentro. L'angolo ω (argomento del pericentro) è l'angolo del vettore di Lane è rispetto alla linea dei nodi $O\hat{D}_0$.
($0 \leq \omega < 2\pi$)

Sappiamo già che a, e danno le dimensioni dell'orbita, se viene dato anche t_p (tempo del passaggio al pericentro) i 6 elementi $a, e, i, \omega, \Omega, t_p$ individuano il moto di $\hat{r}$ per qualunque tempo e sono del tutto equivalenti alle 6 condizioni iniziali. (posizioni e velocità) $\vec{r}(0), \dot{\vec{r}}(0)$ in coordinate cartesiane.

Per avere Ω, ω a partire dalle condizioni iniziali cartesiane basta costruire il vettore del nodo ascendente:

$$\hat{\Omega} = \frac{\hat{z} \times \hat{f}}{|\hat{z} \times \hat{f}|} \quad (4.1)$$

(analogamente, per il nodo discendente:

$$\hat{\Omega} = \frac{\hat{z} \times (-\hat{f})}{|\hat{z} \times (-\hat{f})|})$$

E quindi:

$$\cos \Omega = \hat{\Omega}_0 \cdot \hat{z} \quad (4.2)$$

$$\cos \omega = \hat{\Omega}_0 \cdot \hat{e} \quad (4.3)$$

I 6 elementi $a, e, i, \omega, \Omega, t_p$ si chiamano ELEMENTI KEPLERIANI.

Essi sono validi anche per orbite iperboliche e paraboliche (in questo ultimo caso con P al posto di a ed $e=1$). Per orbite circolari ω non ha senso; per orbite con $i=0$, $i=\pi$ Ω non ha senso. Per questo nel caso di piccole inclinazioni si preferisce usare l'angolo

$$\sigma = \Omega + \omega \quad \text{LONGITUDINE DEL PERICENTRO} \quad (4.4)$$

che per $i \rightarrow 0, \pi$ tende all'angolo del vettore $\hat{r}$ con l'asse della x nel piano xy visto da $\hat{f}$.

Risultato come dalle condizioni iniziali: $\vec{r}(0), \dot{\vec{r}}(0)$ (vedi (3.31)) si ricavano gli elementi kepleriani nel caso $e < 1$

i) calcolo gli integrali primi $\vec{f}, \hat{e}, E$ da cui ricavare a, e (ed anche n, P)

ii) calcolo $\omega(0)$ dalla (3.38), (3.39) e quindi ω tramite l'eq. di

Keplero - il tempo t_p di passaggio al pericentro.

iii) $i = \text{angolo}(\hat{f}, \hat{z})$ visto da $\hat{D}_0$ $0 \leq i < \pi$ INCLINAZIONE

calcolo Ω con la (4.1)

$\Omega = \text{angolo}(\hat{\Omega}_0, \hat{z})$ visto da $\hat{z}$ $0 \leq \Omega < 2\pi$ LONGITUDINE del

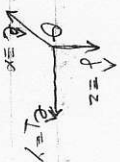
$\omega = \text{angolo}(\hat{e}, \hat{\Omega}_0)$ visto da $\hat{f}$ $0 \leq \omega < 2\pi$ ARG. del

PERICENTRO

Notare che il riferimento $Oxyz$ è un riferimento iniziale parallelo a quello in cui sono state date le condizioni iniziali; è solo traslato rispetto a questo offrendo il centro di massa sia fisso, inoltre in esso si considera il moto relativo di $\hat{r}$ rispetto ad $M = m_1 + m_2$ che è fisso nell'origine O_0 .

Una volta trovate le coordinate al tempo $t_1 > 0$ nel piano dell'orbita

(vedi la (3.46)) dove ruotano da questo riferimento



nel riferimento Θ_{xyz} . Lo faccio in 3 passi:

1. ruoto intorno a $\hat{j}$ di $(-w)$ in modo che portati $\hat{e}$ a coincidere con $\hat{z}$
2. ruoto intorno ad $\hat{j}_0 (= \hat{i})$, perché $\hat{j}$ coincida con $\hat{x}$
3. ruoto intorno a $\hat{z}$ di $(-\Omega)$, perché $\hat{z}$ ($= \hat{e}$) coincida con $\hat{z}$

Nota: ruoto gli assi di riferimento lasciando fissi i vettori.

Ognuna di queste 3 rotazioni è espressa da una matrice ortogonale; la matrice prodotta delle 3 (nell'ordine indicato) è la matrice che porta dal riferimento $\Theta_{\hat{e}, \hat{i}, \hat{j}}$ al riferimento Θ_{xyz} ; se applico R al vettore posizione $\vec{r}(t)$ scritto in $\Theta_{\hat{e}, \hat{i}, \hat{j}}$ ottengo le componenti dello stesso vettore $d\vec{r}/dt$. I vettori posizione $\vec{r}_1$ ed $\vec{r}_2$ di m_1 ed m_2 sono semplicemente (in Θ_{xyz})

$$\begin{cases} \vec{r}_1 = -\frac{m_2}{M} \vec{r} \\ \vec{r}_2 = \frac{m_1}{M} \vec{r} \end{cases} \quad (4.5)$$

e nel riferimento iniziale di figura (3.2):

$$\begin{cases} \vec{r}_1 = \vec{R} + \vec{r} = \vec{R} - \frac{m_2}{M} \vec{r} \\ \vec{r}_2 = \vec{R} + \vec{r} = \vec{R} + \frac{m_1}{M} \vec{r} \end{cases} \quad (4.6)$$

(con $\vec{R}(t)$ dato dalla (3.26))

Ora calcolo la matrice R :

$$R = R_3 R_2 R_1 \quad (4.7)$$

$$R_1 = \begin{pmatrix} \cos(w) & \sin(-w) & 0 \\ -\sin(w) & \cos(w) & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \cos w & -\sin w & 0 \\ \sin w & \cos w & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{step 1} \quad (4.8)$$

$$R_2 = \begin{pmatrix} 1 & 0 & 0 \\ \cos i & -\sin i & 0 \\ \sin i & \cos i & 0 \end{pmatrix} \quad \text{step 2} \quad (4.9)$$

$$R_3 = \begin{pmatrix} \cos \Omega & -\sin \Omega & 0 \\ \sin \Omega & \cos \Omega & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{step 3} \quad (4.10)$$

En seguito la rotazione R basterà applicare le rotazioni successivamente avendo cura di scegliere gli assi di rotazione giusti. VEDREMO CHE IN REALTÀ NEL PROGRAMMA BASTA USARE

MATRICI 2×2

Nota: ho costruito R in modo che ruoti gli assi lasciando fissi i vettori; applicare R significa scrivere lo stesso vettore nel nuovo sistema di coordinate.

Analogamente, applico R al vettore velocità (3.51) (la componente z è nulla perché siamo nel piano dell'orbita) per ottenere le componenti nel riferimento Θ_{xyz} .

Ora possiamo cominciare a scrivere un programma che risolva automaticamente il problema di due corpi secondo il procedimento

mento descritto fin qui (a partire da pag. 13.4); si tratta di un procedimento analitico con la piccola eccezione del metodo di Newton per la soluzione dell'eq. di Keplero.

Fu avanti descritto un metodo numerico per la soluzione di eq. differenziali del tipo (1.1) che applico al problema dei due corpi per fare un confronto con il procedimento analitico e trovare così il metodo numerico.

Ovviamente, visto poi il metodo numerico per risolvere problemi più interessanti che non sono integrabili se non numericamente.

§2. PRINCIPI BASE DEL LINGUAGGIO FORTRAN CON ESEMPI RELATIVI AL PROBLEMA DEI DUE CORPI

(FORTRAN: FORMULA TRANSLATOR)

Un moderno elaboratore elettronico è una macchina che esegue le 4 operazioni e capisce alcuni linguaggi (FORTRAN, BASIC, APL, PASCAL etc...). Per interagire con esso e ottenere che faccia del lavoro per noi bisogna "parlarne" uno di questi linguaggi.

Una volta tradotte le nostre formule matematiche in un programma FORTRAN (cioè in una sequenza di istruzioni comprensibile al calcolatore) un apposito programma COMPILATORE provvederà a tradurre il nostro programma FORTRAN in linguaggio macchina, cioè a spezzettare ogni operazione in una serie di operazioni elementari che sono quelle che la macchina esegue per ottenere, ad esempio, la somma di due numeri (il calcolatore lavora in binario).

• In FORTRAN è permesso di usare delle variabili: il cui nome consiste al massimo di 6 caratteri alfanumerici (lettere e numeri); il primo carattere deve essere una lettera. Poiché la variabile può essere reale o intera è convenzione che ciò sia stabilito dalla prima lettera:

A, H - O, Z reale

I, J, K, L, M, N intera

Le 4 operazioni si indicano con
+, -, *, /
e l'elevamento a potenza con

L'ordine con cui vengono eseguite le operazioni è quello usuale (prima l'elevamento a potenza, poi moltiplicazioni e divisione, infine somma e differenza). In caso di dubbio si possono usare le parentesi - che sono solo lode - ;
attenzione a dividere tutte.

• Si noti che il segno di = non ha il significato usuale:

$$A = B$$

significa quindi il valore della variabile B (a destra) è assegnato alla variabile A (a sinistra). Per questo ha senso scrivere

$$X = X + 1$$

che significa: prendi il valore della variabile X, aumentalo di uno e assegnalo ancora alla variabile X.

• Le variabili possono avere una dimensione > 1 che può essere dichiarata in cima al programma

DIMENSION A(N)

Per le operazioni vanno sempre eseguite componente per componente.

• E' possibile passare dei dati al programma per mezzo di un DATA
DATA PIGR/3.1415926/

• Il vantaggio fondamentale del FORTRAN è quello di essere un linguaggio strutturato, in modo tale che il programma principale può delegare tutta una serie di calcoli a delle FUNCTION e SUBROUTINE. Questo fa sì che il programma principale resti limitato e comprensibile ed abbia conto i sottoprogrammi possono essere scritti e testati separatamente.

Una FUNCTION è un'entità una funzione di una o più variabili: per ogni valore delle variabili indipendenti fornisce un numero ^(numerical) che si indica col nome che viene dato alla FUNCTION

Una SUBROUTINE può avere più di una variabile sia in INPUT che in OUTPUT. Si chiama dal programma principale con l'istruzione CALL seguita dal nome della subroutine e dalla corrispondente sequenza di chiamate (vedi esempi).

Il nome di una FUNCTION o SUBROUTINE è una stringa

allarmistica di un massimo di 6 caratteri il nome di
quale deve essere una lettera; il nome di una FUNCTION
e anche il nome della variabile dipendente fornita in output
dalla FUNCTION stessa e può dare cominciare con una
lettera che rispetti le regole (2.1) (vedi esempi).

NOTA:
1 2 3 4 5 6
| | | | |
istruzione
Fm FORTRAN ogni istruzione di programma
comincia alla colonna numero 7; la

colonna n. 6 si usa per continuare una
stessa istruzione che risulti essere più lunga
del numero di caratteri disponibile in una
riga (in genere
le colonne 2, 3, 4 e 5 si usano, quando
necessario (vedere esempi), per attribuire un
numero a quella data istruzione;
la colonna n. 1 si usa per scrivere una C
(per commento) quando si vuole che quella
riga non sia una riga di istruzione del
programma, ma semplicemente di commento
per ricordare a chi ha scritto il programma
qualche informazione importante sul
programma stesso (ad esempio, che era fe
quel programma o cosa vogliono dire
certi simboli ecc...). E' buona norma
in certe righe di commento.

Esempi di SUBROUTINE e FUNCTION
1, 2, 3, 4, 5, 6, 7

SUBROUTINE PRVEC (A, B, C)
C CALCOLA IL PRODOTTO VETTORE DI DUE VETTORI DI R3
C C = A x B

DIMENSION A(3), B(3), C(3)
C (1) = A(1)*B(3) - A(3)*B(2)
C (2) = A(3)*B(1) - A(1)*B(3)
C (3) = A(1)*B(2) - A(2)*B(1)
RETURN
END

FUNCTION PRSCAL (A, B)
C CALCOLA IL PRODOTTO SCALARE DI DUE VETTORI DI R3
C PRSCAL = A · B

DIMENSION A(3), B(3)
PRSCAL = 0.
DO 1 I = 1, 3
PRSCAL = PRSCAL + A(I)*B(I)
1 CONTINUE
RETURN
END

SUBROUTINE SUM (A, B, C)
C CALCOLA LA SOMMA DI DUE VETTORI DI R3
C C = A + B
DIMENSION A(3), B(3), C(3)
DO 1 I = 1, 3
C(I) = A(I) + B(I)

Mo

```
1 CONTINUE
RETURN
END
```

SUBROUTINE SCMU (ALFA, A, B)

C CALCOLA IL PRODOTTO DI UN VETTORE DI R3 PER UNO SCALARE REALE

```
C B = ALFA * A
  DIMENSION A(3), B(3)
  DO 1 I=1,3
```

```
B(I) = ALFA * A(I)
```

```
1 CONTINUE
```

```
RETURN
```

```
END
```

SUBROUTINE VLINC (ALFA, A, BETA, B, C)

C CALCOLA LA COMBINAZIONE LINEARE DI DUE VETTORI DI R3

```
C C = ALFA * A + BETA * B
```

```
  DIMENSION A(3), B(3), C(3), W1(3), W2(3)
```

```
  CALL SCMU (ALFA, A, W1)
```

```
  CALL SCMU (BETA, B, W2)
```

```
  CALL SUM (W1, W2, C)
```

```
RETURN
```

```
END
```

FUNCTION SIZE(A)

C CALCOLA LA LUNGHEZZA DI UN VETTORE DI R3

```
C SIZE = MODULO DI A
```

```
  DIMENSION A(3)
```

```
  SIZE = SQRT (PRSCAL(A,A))
```

```
RETURN
```

```
END
```

Nota: l'uso dei vettori
auxiliari W1 e W2
(iniziali da "workspace")

FUNCTION PROTRI (A, B, C)

C CALCOLA IL PRODOTTO TRIPLO DI TRE VETTORI DI R3

```
C PROTRI = A · (B × C)
```

```
  DIMENSION A(3), B(3), C(3), W(3)
```

```
  CALL PRVEC (B, C, W)
```

```
  PROTRI = PRSCAL (A, W)
```

```
RETURN
```

```
END
```

SUBROUTINE ROTA (A, TETA, B)

C RUOTA IL VETTORE A DEL PIANO DI UN ANGOLO TETA PORTANDOLO

C NEL NUOVO VETTORE B E LASCIANDO FERNI GLI ASSI. OPPURE, RUOTA

C GLI ASSI DI -TETA LASCIANDO FERNI I VETTORI. (GLI ANGOLI SONO POSITIVI

C SE LA ROTAZIONE AVVIENE IN VERSO ANTIORARIO)

```
  DIMENSION A(2), B(2), W(2)
```

```
  COST = COS (TETA)
```

```
  SINT = SIN (TETA)
```

```
  W(1) = A(1) * COST - A(2) * SINT
```

```
  W(2) = A(1) * SINT + A(2) * COST
```

```
  B(1) = W(1)
```

```
  B(2) = W(2)
```

```
RETURN
```

```
END
```

← sento di dover ricalcolare
il seno e il coseno di
θ se mi servono più di
una volta

Nota: il vettore ausiliario

W in questo caso serve per
poter dare al nuovo vettore lo
stesso nome del vecchio, cosa
che può essere comoda. Se dovessi:

→ A(1) = A(1) * COST - A(2) * SINT
A(2) = A(1) * SINT + A(2) * COST.

questa è la prima componente
del vettore ruotato!

• Tutte le SUBROUTINE e FUNCTION scritte finora servono per eseguire operazioni generali tra vettori di R³. Ad esempio, una qualunque rotazione - di un vettore o degli assi - coordinati - nello spazio si può scrivere componendo 3 rotazioni del tipo di quelle descritte dalla SUBROUTINE ROTA, ciascuna attorno ad un asse opportuno. conviene tenere tutte queste routine in uno stesso file.

• Adesso mi occupo di scrivere l'eq. di Keplero^(3.21) ed il metodo di Newton (3.54). Lo faremo con una FUNCTION che fornisce l'anomalia eccentrica in un valore fisso della anomalia media E . Ricordo che per prima cosa occorre rivedere l'angolo E ed i suoi valori $-\pi$ e π :

```

FUNCTION PRINC (TETA)
C DATO UN ANGOLO TETA IN RADIANI CALCOLA IL SUO VALORE
C NELL'INTERVALLO  $-\pi/4, +\pi/4$  ( $\pi/4 = \pi/4\text{RADO}$ )
DATA PI4 / 3.1415926 /
DATA DPI4 / 6.2831852 /
NGR1 = ABS (TETA) / DPI4
TETA = TETA - (NGR1 * DPI4 * SIGN (TETA))
IF (TETA.LT. -PI4) TETA = TETA + DPI4
IF (TETA.GT. PI4) TETA = TETA - DPI4
IF (TETA.GT. PI4) TETA = TETA - DPI4
RETURN
END
    
```

less than =
minore di =
(in senso stretto)
greater than = maggiore di =>

ABS fornisce il valore assoluto di una variabile. E' una funzione già esistente. SIGN e una function - da definire - che fornisce il segno di una variabile

FUNCTION SIGN (X)
C FORNISCE IL SEGNO DELLA VARIABILE X (1 SE PIU', -1 SE MENO, 0 SE ZERO)

```

IF (X.GT.0) SIGN = 1
IF (X.LT.0) SIGN = -1
IF (X.EQ.0) SIGN = 0
RETURN
END
    
```

anomalia eccentrica
eccentricità
anomale media

C CALCOLA L'ANOMALIA ECCENTRICA U, DATE L'ECCENTRICITA' E e
C L'ANOMALIA MEDIA AM, RISOLVENDO L'EQ. DI KEPLERO $AN = U - E * SIN(U)$
C COL METODO DI NEWTON. EPS E' LA TOLLERANZA NELLA CONVERGENZA
C DEL METODO DI NEWTON, LA DOBBIAMO DARE TENENDO CONTO DEL NUMERO
C DI CIFRE SIGNIFICATIVE CON CUI IL CALCOLATORE LAVORA.
C PIU' < AM < PI4; U(0) = PI4 SE AM > 0; U(0) = -PI4 SE AM < 0
C U(N+1) = U(N) - (U(N) - E * SIN(U(N)) - AM) / (1 - E * COS(U(N)))

```

DATA PI4 / 3.1415926 /
AM = PRINC (AM)
U = PI4 * SIGN (AM)
DO 1 I = 1, 20
D = (U - E * SIN(U) - AM) / (1 - E * COS(U))
U = U - D
IF (ABS(D).LT.EPS) RETURN
1 CONTINUE
WRITE (6,100) D, AM, U, E
100 FORMAT ('NON CONVI IN KEPLER EQ' / 'D = ' E20.7 /  
'AM = ' E20.7 / 'U = ' E20.7 / 'E = ' E20.7 /)
READ (5,*) L
RETURN
END
    
```

READ (5,*) L
S = il device da cui legge e lo abbiamo

> 20 cifre di cui 7 dopo la virgola; il 20 comprende segno, esponente e segno

6 = il device su cui scrive e lo schermo indica con che formato scrive

100 FORMAT ('NON CONVI IN KEPLER EQ' / 'D = ' E20.7 / 'AM = ' E20.7 / 'U = ' E20.7 / 'E = ' E20.7 /)

Ho bisogno di una subroutine per far passare dalle coordinate cartesiane a quelle del problema rotato: (3.24 e 3.25)

SUBROUTINE REDUCE (RH1, RH2, RH1P, RH2P, PM1, PM2, R, RP, BQ, BP, PNTOT, PHRED)

C FA IL PASSAGGIO DALLE COORDINATE CARTESIANE DEI DUE CORPI
C (RH1, RH2 POSIZIONI, RH1P, RH2P VELOCITA') A QUELLE DEL PROBLEMA
C RIDOTTO (R, RP POSIZIONE E VEL RELATIVA, BQ, BP POSIZIONE E
C VEL DEL BARICENTRO). IN INPUT VENGONO DATE LE MASSE DEI DUE CORPI
C PM1, PM2, IN OUTPUT ABBIAO LA MASSA RIDOTTA PHRED E LA
C MASSA TOTALE PNTOT.

DIMENSION RH(3), RH2(3), RH1P(3), RH2P(3), R(3), RP(3), BQ(3), BP(3)

CALL SUM (RH1, -RH2, R)

CALL SUM (RH2P, -RH1P, RP)

PNTOT = PM1 + PM2

PHRED = PM1 * PM2 / PNTOT

CALL VLINC (PM1/PNTOT, RH1, PM2/PNTOT, RH2, BQ)

CALL VLINC (PM1/PNTOT, RH1P, PM2/PNTOT, RH2P, BP)

RETURN

END

Viceversa ho bisogno anche di una subroutine per tornare dal problema rotato a quello non rotato (m_1, m_2, M in input): (4.6)

SUBROUTINE UNREDU (R, RP, BQ, BP, PNTOT, PHRED, PM1, PM2, RH1, RH2, RH1P, RH2P)

C FA IL PASSAGGIO DAL PROBLEMA RIDOTTO A QUELLO NON RIDOTTO. IN
C INPUT POS. E VEL RELATIVA R, RP, POSIVE VEL DEL BARICENTRO
C BQ, BP, MASSA TOTALE E MASSE DEI DUE CORPI PNTOT, PM1, PM2.
C IN OUTPUT POS. E VEL DEI DUE CORPI RH1, RH2, RH1P, RH2P.

DIMENSION R(3), RP(3), BQ(3), BP(3), RH1(3), RH2(3), RH1P(3), RH2P(3)

CALL VLINC (1., BQ, -PM2/PNTOT, R, RH1)

CALL VLINC (1., BP, -PM1/PNTOT, R, RH2)

CALL VLINC (1., BP, -PM2/PNTOT, RP, RH1P)

CALL VLINC (1., BP, PM1/PNTOT, RP, RH2P)

RETURN

END

duecor.f

```

program kepleg
implicit double precision (a-h,o-z)
C Main per il calcolo delle effemeridi dei pianeti nella approssimazione
C del problema dei due corpi mediante soluzione della equazione
C di Keplero. Si parte da coordinate cartesiane relative (problema ridotto).
dimension r(3),rp(3),ek(7)
write(*,*)'t0?'
read(*,*)t0
eps=1.d-15
gm=1.d0
write(*,*)'vettore posizione relativa a t0?'
read(*,*)r
write(*,*)'vettore velocita relativa a t0?'
read(*,*)rp
C Chiamo la routine che fornisce il vettore degli elementi kepleriani
C ek=n,e,i,argomento del pericentro,longitudine del nodo ascendente,
C anomalia media; semiasse maggiore). Notare che ek e' lungo 7 perche'
C all'ultimo posto ho aggiunto il semiasse maggiore. Notare che
C l'anomalia media e' quella del tempo t0 -am0-.
call carkep(r,rp,gm,eps,ek,en)
open(2,file='uscita.kep', status='new')
write(2,202)t0,r,rp,ek,en
C ek(6) e' l'anomalia media al tempo t0
202 format('t0=',d23.16/'r(t0)=' ,3d23.16/'rp(t0)=' ,3d23.16/
* 'ek(n,e,i,omp,omg,l,a)=' ,/3d23.16/3d20.12/d23.16/'en=' ,d23.16)
am0=ek(6)
C Mi chiede il tempo successivo t1
do 1 i=1,10
write(*,*)'t1?'
read(*,*)t1
C Calcolo l'anomalia media al tempo t1
deltat=t1-t0
ek(6)=am0+ek(1)*deltat
C Chiamo la routine che fornisce posizione e velocita' relative r,rp a t1
C e anche l'energia (per unita' di massa ridotta) a t1 per controllare
C che si mantenga costante
call kepcar(ek,gm,eps,r,rp,en)
write(2,302)t1,r,rp,en
C Notare che en e' stata ricalcolata al tempo t1
302 format('t1,r,rp,en',/d23.16/3d20.12/3d20.12/d24.16)
1 continue
end
C*****
C*****
subroutine carkep(r,rp,gm,eps,ek,en)
implicit double precision (a-h,o-z)
C Date le coordinate cartesiane relative r,rp della massa ridotta e il
C prodotto gm calcola il vettore ek degli elementi kepleriani e

```

1

duecor.f

```

C l'energia en (per unita' della massa ridotta). Si noti che ek e' lungo
C 7 e non 6: moto medio, eccentricita', inclinazione, argomento del
C pericentro, longitudine del nodo ascendente, anomalia media e semiasse
C maggiore. eps dipende dalla macchina, che non sa distinguere due
C numeri che differiscono di <= di eps.
dimension r(3),rp(3),ek(7),gei(3),vz(3),g(3),ev(3),an(3),
*anv(3)
data pig,vz/3.1415926535897932d0,0.d0,0.d0,1.d0/
C Calcolo il momento angolare per unita' di massa gei
call prvec(r,rp,gei)
geis=size(gei)
C Se geis e' zero non ho moto orbitale
if(geis.lt.eps)goto 101
cosi=gei(3)/geis
C Calcolo l'inclinazione ek(3)
ek(3)=dacos(cosi)
C Se i e' 0 o pig il nodo non e' definito
if(ek(3).lt.eps)goto 102
if(ek(3).gt.pig-eps)goto 103
C Calcolo il vettore di Lenz e
call prvec(rp,gei,e)
rs=size(r)
call vlinc(1.d0/gm,e,-1.d0/rs,r,e)
C Calcolo l'eccentricita' ek(2)
ek(2)=size(e)
C ev=versore dall'origine al pericentro
call scmu(1.d0/ek(2),e,ev)
C Se l'eccentricita' e' 0 il pericentro non e' definito
if(ek(2).lt.eps)goto 104
C Calcolo il vettore del nodo ascendente con versore anv
call prvec(vz,gei,an)
ans=size(an)
C ek(5)=longitudine del nodo ascendente
ek(5)=datan2(an(2),an(1))
C ek(4)=omega piccolo=argomento del pericentro
call scmu(1.d0/ans,an,anv)
cosop=prscal(anv,ev)
call prvec(anv,ev,g)
C sinop=prscal(g,gei)/geis possibile errore di segno in sinop
sinop=size(g)
ek(4)=datan2(sinop,cosop)
C duen=doppio dell'energia orbitale (per unita' di massa ridotta)
duen=prscal(rp,rp)-2.d0*gm/rs
en=duen/2.d0
C Deve essere negativa perche' studio le orbite ellittiche
if(en.gt.-eps)goto 105
C ek(7)=semiasse maggiore
ek(7)=-gm/duen

```

2

duecor.f

```

C ek(1)=moto medio
  ek(1)=(-dueen)**1.5d0/gm
C Calcolo l'anomalia eccentrica u
  sinu=prscal(r,rp)/(ek(1)*ek(2)*ek(7)*ek(7))
  cosu=(1.d0-rs/ek(7))/ek(2)
  u=atan2(sinu,cosu)
C Calcolo l'anomalia media ek(6) dall'equazione di Keplero
  ek(6)=u-ek(2)*sinu
  return
101 write(*,*)'mom ang zero'
  stop
102 write(*,*)'inclinaz zero'
  stop
103 write(*,*)'inclinaz pig'
  stop
104 write(*,*)'eccentricita zero'
  stop
105 write(*,*)'orbita aperta'
  stop
end
C*****
  subroutine kepcar(ek,gm,eps,r,rp,en)
  implicit double precision (a-h,o-z)
C Dato il vettore degli elementi kepleriani (lungo 7) (n,e,i, ompic, omg
C l,a) calcola posizione e velocita' relativa x e v nel piano
C dell'orbita e li ruota fino ad ottenere posizione e velocita' nel
C riferimento inerziale di partenza r,rp. Ricalcola anche l'energia en
C per controllo. eps come al solito.
  dimension ek(7),x(3),v(3),r(3),rp(3)
  u=ecca(ek(2),ek(6),eps)
  cosu=dcos(u)
  sinu=dsin(u)
  beta=dsqrt(1.d0-ek(2)*ek(2))
  x(1)=ek(7)*(cosu-ek(2))
  x(2)=ek(7)*beta*sinu
  x(3)=0.d0
  call rota(x,ek(4),x)
  call rota(x(2),ek(3),x(2))
  r(3)=x(3)
  call rota(x,ek(5),r)
  rs=size(r)
C Calcolo la velocita'
  rsua=1.d0-ek(2)*cosu
  up=ek(1)/rsua
  v(1)=-ek(7)*up*sinu
  v(2)=ek(7)*beta*up*cosu
  v(3)=0.d0
  call rota(v,ek(4),v)

```

3

duecor.f

```

  call rota(v(2),ek(3),v(2))
  rp(3)=v(3)
  call rota(v,ek(5),rp)
  v2=prscal(rp,rp)
  en=v2/2.d0-gm/rs
  return
end
C*****
  subroutine prvec (a,b,c)
C Calcola il prodotto vettore di due vettori di R3. axb=c
  implicit double precision (a-h,o-z)
  dimension a(3), b(3), c(3)
  c(1) = a(2)*b(3)-a(3)*b(2)
  c(2) = a(3)*b(1)-a(1)*b(3)
  c(3) = a(1)*b(2)-a(2)*b(1)
  return
end
C*****
  double precision function prscal(a,b)
C Calcola il prodotto scalare di due vettori di R3. prscal=a.b
  implicit double precision (a-h,o-z)
  dimension a(3),b(3)
  prscal=0.d0
  do 1 i=1,3
    prscal=prscal+a(i)*b(i)
  1 continue
  return
end
C*****
  subroutine sum(a,b,c)
C Calcola la somma di due vettori di R3. a+b=c
  implicit double precision (a-h,o-z)
  dimension a(3),b(3),c(3)
  do 1 i=1,3
    c(i)=a(i)+b(i)
  1 return
  return
end
C*****
  subroutine scmu(alfa,a,b)
C Calcola il prodotto di uno scalare per un vettore di R3. alfaa=b
  implicit double precision (a-h,o-z)
  dimension a(3),b(3)
  do 1 i=1,3
    b(i)=alfa*a(i)
  1 return
  return
end
C*****

```

4

```

duecor.f

      subroutine vlinc(alfa,a,beta,b,c)
c Calcola la combinazione lineare di due vettori di R3
c alfa+beta*x=c
      implicit double precision (a-h,o-z)
      dimension a(3),b(3),c(3),w1(3),w2(3)
      call scmu(alfa,a,w1)
      call scmu(beta,b,w2)
      call sum(w1,w2,c)
      return
      end
c*****
      double precision function size(a)
c Calcola la lunghezza di un vettore di R3. size=modulo di a
      implicit double precision (a-h,o-z)
      dimension a(3)
      size=dsqrt(prscal(a,a))
      return
      end
c*****
      double precision function protri(a,b,c)
c Calcola il prodotto triplo di 3 vettori di R3
c protri=a(bxc)
      implicit double precision (a-h,o-z)
      dimension a(3),b(3),c(3),w(3)
      call prvec(b,c,w)
      protri=prscal(a,w)
      return
      end
c*****
      subroutine rota(a,teta,b)
c matrice che ruota il vettore a (di un agolo teta, positivo in verso
c antiorario) nel vettore b lasciando invariati gli assi coordinati;
c analogamente, la matrice fornisce le coordinate dello stesso vettore
c tenuto fisso nel nuovo sistema di assi ottenuto ruotato di (-teta)
c rispetto a quello di partenza
      implicit double precision (a-h,o-z)
      dimension a(2),b(2),w(2)
      cost=dcos(teta)
      sint=dsin(teta)
      w(1)=a(1)*cost-a(2)*sint
      w(2)=a(1)*sint+a(2)*cost
      b(1)=w(1)
      b(2)=w(2)
      return
      end
c*****
      double precision function ecca(e,am,eps)
      implicit double precision (a-h,o-z)

```

5

```

duecor.f

c      data pig/3.1415926535897932d0/
      pig=4.d0*datan(1.d0)
      am=princ(am)
      u=pig*segno(am)
      do 1 i=1,20
        d=(u-e*dsin(u)-am)/(1.d0-e*dcos(u))
        u=u-d
        if(dabs(d).lt.eps)then
          ecca=u
          return
        endif
1      continue
      write(*,106)d,am,u,e
106  format('non convergenza eq kepler','/'d= 'd23.16/'am= 'd23.16/
*'ecca= 'd23.16/'e= 'd23.16)
      return
      end
c*****
      double precision function princ(teta)
c Fornisce il valore di un angolo in radianti tra -pig e +pig
      implicit double precision (a-h,o-z)
c      data pig/3.1415926535897932d0/
      pig=4.d0*datan(1.d0)
      dpig=2.d0*pig
      ngiri=dabs(teta)/dpig
c fornisce teta tra 0 e dpig oppure 0 e -dpig. Ridurre tra -pig e pig
      teta=teta-(ngiri*dpig*segno(teta))
      if(teta.lt.-pig)teta=teta+dpig
      if(teta.gt.pig)teta=teta-dpig
      princ=teta
      return
      end
c*****
      double precision function segno(x)
c Fornisce il segno di una variabile x
      implicit double precision (a-h,o-z)
      if(x.eq.0.d0)segno=0.d0
      if(x.gt.0.d0)segno=1.d0
      if(x.lt.0.d0)segno=-1.d0
      return
      end
c*****

```

6