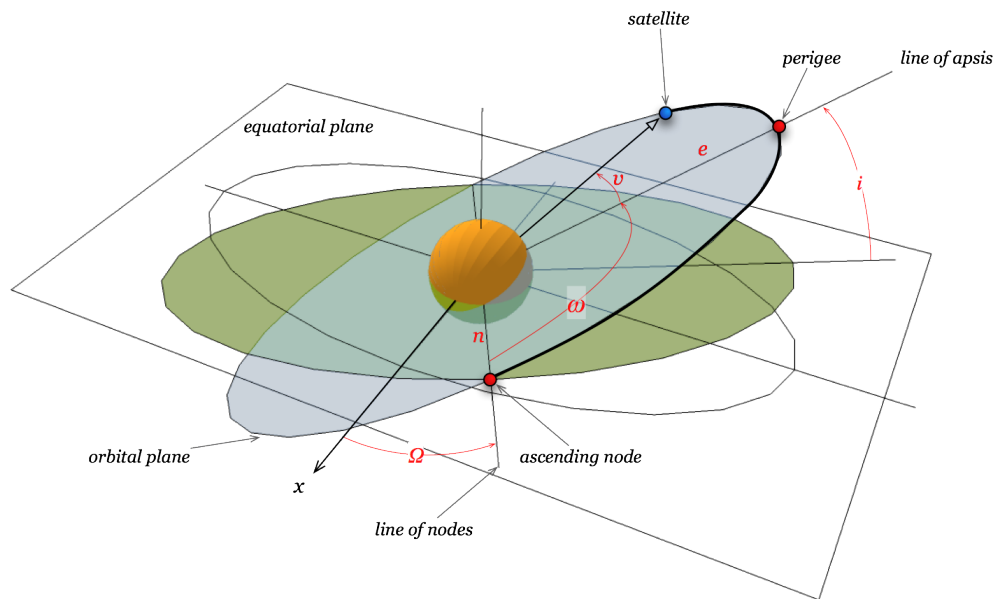


ASTRO lib and Matlab Wrapper

Enrico Bertolazzi and Francesco Biral

1 August 2021

1 ASTRO library



An astronomical unit (abbreviated as **AU**, **au**) is a unit of length equal to about 149,597,871 kilometres (92,955,807 miles) or approximately the mean EarthSun distance.

Il wrapper Matlab della libreria **ASTRO** prevede un'unica chiamata in cui il primo argomento e' una stringa che specifica il metodo della libreria C++ da usare:

I metodi resi disponibili dalla libreria **ASTRO** sono i seguenti:

ASTRO('load', 'filename.txt')

Carica il file dati degli asteroidi definiti con i parametri orbitali dal file 'filename.txt'. Il file da caricare non è più la versione "new", ma quella originale.

astName = **ASTRO**('getname', astNum)

Restituisce il nome astName dell'asteroide astNum.

year = **ASTRO**('period', astNum)

Restituisce la durata dell'anno in giorni year per l'asteroide astNum.

L = **ASTRO**('angle', astNum,t)

Restituisce l'angolo L in coordinate equinoziali dell'asteroide astNum al tempo t espresso in giorni. In questo caso, t è un singolo istante e non un vettore di tempi.

[e, a, i, Omega, omega] = **ASTRO**('orbitalclassic', astNum)

Estrae i parametri orbitali classici (Kepleriani) dato il numero astNum dell'asteroide. Si suppone di aver caricato nella memoria della libreria **ASTRO** i dati degli asteroidi.

[p, f, g, h, k] = **ASTRO**('orbitalequinoctial', astNum)

Fornisce i parametri orbitali in coordinate equinoziali dell'asteroide numero astNum. Si suppone di aver caricato nella memoria della libreria **ASTRO** i dati degli asteroidi.

```
l =  
    ASTRO('invariantl', p, f, g, h, k)
```

Fornisce il vettore di parametri invarianti l per l'identificazione dei cluster.

```
a =  
    ASTRO('invarianta', p, f, g, h, k)
```

Fornisce il vettore di parametri invarianti a per l'identificazione dei cluster.

```
en =  
    ASTRO('orbitenergy', p, f, g, h, k)
```

Fornisce l'energia associata ad un'orbita, dati i suoi parametri orbitali equinoziali.

```
[p, f, g, h, k, L] =  
    ASTRO('classictoequinoctial', e, a, i, Omega, omega, nu)
```

Converte nelle coordinate equinoziali le coordinate classiche.

```
[e, a, i, Omega, omega, nu] =  
    ASTRO('equinoctialtoclassic', p, f, g, h, k, L)
```

Converte nelle coordinate classiche le coordinate equinoziali.

```
[p, f, g, h, k, L] =  
    ASTRO('cartesiantoequinoctial', x, y, z, vx, vy, vz)
```

Converte nelle coordinate equinoziali le coordinate cartesiane.

```
[x, y, z, vx, vy, vz] =  
    ASTRO('equinoctialtocartesian', p, f, g, h, k, L)
```

Converte nelle coordinate cartesiane le coordinate equinoziali.

```
[x, y, z] = ASTRO('position', astNum, timeSpan)
```

Fornisce la posizione in coordinate cartesiane dell'asteroide numero `astNum` per l'intervallo di tempo `timeSpan` definito in numero di giorni. Vengono restituiti tre vettori.

```
[vx, vy, vz] = ASTRO('velocity', astNum, timeSpan)
```

Fornisce la velocità in coordinate cartesiane dell'asteroide numero `astNum` per l'intervallo di tempo `timeSpan` definito in numero di giorni. Vengono restituiti tre vettori.

```
[x, y, z, vx, vy, vz] = ASTRO('pos+vel', astNum, timeSpan)
```

Fornisce in un'unica chiamata la posizione e la velocità in coordinate cartesiane dell'asteroide numero `astNum` per l'intervallo di tempo `timeSpan` definito in numero di giorni. Vengono restituiti sei vettori.

```
[ax, ay, az] = ASTRO('acceleration', astNum, timeSpan)
```

Fornisce l'accelerazione in coordinate cartesiane dell'asteroide numero `astNum` per l'intervallo di tempo `timeSpan` definito in numero di giorni. Vengono restituiti tre vettori.

```
[jx, jy, jz] = ASTRO('jerk', astNum, timeSpan)
```

Fornisce il jerk in coordinate cartesiane dell'asteroide numero `astNum` per l'intervallo di tempo `timeSpan` definito in numero di giorni. Vengono restituiti tre vettori.

```
[e, a, i, Omega, omega, nu] =  
  ASTRO('evalclassic', x, y, z, vx, vy, vz)
```

Converte in coordinate classiche le coordinate cartesiane.

```
[p, f, g, h, k, L] = ASTRO('evalequinocial', x, y, z, vx, vy,  
vz)
```

Converte in coordinate equinoziali le coordinate cartesiane.

```
[Drx, Dry, Drz, Dtx, Dty, Dtz, Dnx, Dny, Dnz ] =  
  ASTRO('evalfrenet',x, y, z, vx, vy, vz)
```

Converte in coordinate cartesiane i versori r, t, n della terna di Frenet solidale alla sonda.

```
[x, y, z, vx, vy, vz, mass] =  
  ASTRO('integratecartesian',x0, y0, z0, vx0, vy0, vz0, mass0,  
  timeSpan, Tr, Tt, Tn)
```

Calcola in coordinate cartesiane posizione, velocità e massa della sonda per l'intervallo di tempo `timeSpan` a partire dalle condizioni iniziali di posizione e velocità e data la storia dei controlli.

```
[p, f, g, h, k, L, mass] =  
  ASTRO('integrateequinoctial', p0, f0, g0, h0, k0, L0, mass0,  
  timeSpan, Tr, Tt, Tn)
```

Calcola in coordinate equinoziali posizione, velocità e massa della sonda per l'intervallo di tempo `timeSpan` a partire dalle condizioni iniziali di posizione e velocità in coordinate equinoziali e data la storia dei controlli.

```
RES = ASTRO('trajectory', N, tStep, P0, P1,  $\alpha$ ,  $\lambda$ )
```

Calcola in coordinate equinoziali la traiettoria della sonda, il tempo t ed i controlli necessari a percorrerla. Gli ingressi sono il numero di punti N in cui la si vuole discretizzare, il passo temporale `tStep`, i vettori posizione iniziale e finale $P0$ e $P1$ di dimensione 6 in coordinate equinoziali, e due vettori di fattori di forma da 5 elementi, $\alpha[0 : 1]$ e $\lambda[-10 : 10]$. Questa funzione dovrebbe essere utilizzata soltanto dalle funzioni interne della libreria, senza necessità per l'utente di chiamarla manualmente.

```
RES = ASTRO('trajectory1', N, tStep, P0, P1,  $\alpha$ ,  $\lambda$ )
```

Simile alla precedente, calcola la traiettoria minimizzando il thrust, ma il tempo finale potrebbe non coincidere con quello dell'asteroide.

```
structRendezvous =
```

```
ASTRO('rendezvous', targetAst, ti, tf, n,  
      p0, f0, g0, h0, k0, L0, mass0)
```

Restituisce una struttura di risultati structRendezvous relativi alla manovra completa di rendezvous. La manovra viene calcolata sulla base delle coordinate equinoziali di partenza, espresse nel vettore [p0, f0, g0, h0, k0, L0, mass0], l'asteroide obiettivo per il rendezvous targetAst, l'istante iniziale, l'istante finale stimato, la massa iniziale ed il numero n di punti su cui calcolare la manovra ottima.

```
structFlyby =  
    ASTRO('flyby', targetAst, ti, tf, n,  
          p0, f0, g0, h0, k0, L0, mass0)
```

Restituisce una struttura di risultati structFlyby relativi alla manovra completa di flyby. La manovra viene calcolata sulla base delle coordinate equinoziali di partenza, espresse nel vettore [p0, f0, g0, h0, k0, L0, mass0], l'asteroide obiettivo per il flyby targetAst, l'istante iniziale, l'istante finale stimato, la massa iniziale ed il numero n di punti su cui calcolare la manovra ottima. La differenza con il precedente è solo nelle condizioni finali richieste, che caratterizzano le due manovre.

```
structGuess =  
    ASTRO('guess', P0, targetAst)
```

Restituisce una struttura di risultati structGuess relativi ad una manovra di primo tentativo che può essere usata dal solutore come soluzione iniziale, sulla base del vettore posizione iniziale in coordinate equinoziali e dell'asteroide a raggiungere.

2 Altre funzioni Matlab

Nella cartella di lavoro sono presenti anche alcune altre funzioni Matlab, utili per operazioni accessorie, post-processing e visualizzazione dei risul-

tati. In particolare si trovano:

```
[Tr, Tt, Tn, x, y, z, vx, vy, vz, mass] =  
  ASTROOpenLoopManoeuvre(x0, y0, z0, vx0, vy0, vz0, mass0, Tri,  
  Tti, Tni, Trf, Ttf, Tnf, t0, ti, tm, tf, steps)
```

Calcola la traiettoria della sonda generata da una data storia di controlli, costituita da tre fasi: spinta costante, rilascio e spinta costante. Gli ingressi sono: posizione e velocità cartesiane iniziali, massa iniziale, componenti di spinta nella terna di Frenet nella prima e nell'ultima fase, istante iniziale, durata temporale delle tre fasi e numero di passi in cui si vuole suddividere la traiettoria. La funzione restituisce quindi i vettori con la storia di controlli, posizione, velocità e massa della sonda.

```
y = asteroidYear(a)
```

Calcola la durata dell'anno di un dato asteroide, data la lunghezza del suo semiasse maggiore a.

```
d = ASTRODistance(ast1,ast2,timeSpan)
```

restituisce la storia delle distanze d tra gli asteroidi ast1 e ast2 nell'arco di tempo timeSpan.

```
ASTROPlotAsteroidXYZ(astNum, viewPoint, ti, tf)
```

Disegna l'orbita annuale dell'asteroide astNum, dal punto di vista specificato in viewPoint (2 per una vista 2D dall'altro, 3 per la vista standard 3D, oppure un vettore [azimuth,elevation] per una vista personalizzata). Se vengono forniti come ingressi opzionali anche due estremi temporali ti e tf, all'orbita annuale sovrappone la traiettoria percorsa nell'intervallo temporale specificato.

```
ASTROPlotRocketXYZ(x, y, z, vx, vy, vz, Tr, Tt, Tn, timeStep, scale,
```

viewPoint)

Disegna la traiettoria percorsa dalla sonda fornita in ingresso. Disegna inoltre una freccia ogni timeStep punti, che rappresenta modulo e direzione della spinta in quell'istante. La dimensione delle frecce è scalabile modificando l'ingresso scale, e la figura viene visualizzata secondo quanto impostato in viewPoint, come nel caso precedente.

[Iin, Dist, IDmin, Dmin] = [closeAst](#)(ast0, day, dist0, period, plt)

Trova i satelliti vicini all'asteroide dato ast0 per un specifico giorno day. I satelliti sono cercati all'interno di una distanza dist0. Quando il parametro plt è uguale viene generato il grafico 3D della traiettoria per il periodo period. La funzione restituisce Iin il vettore degli indici degli asteroidi trovati, Dist il vettore delle distanze da ast0 e l'indice dell'asteroide IDmin più vicino e la Dmin e la sua distanza dall'asteroide ast0.

[Iin, Dist, IDmin, Dmin] = [ASTROPlotManoeuvre](#)(ast0, ast1, Manoeuvre, Tbegin, type)

Se type uguale a 'trajectory' grafica la traiettoria dei due satelliti e quella della sonda. Se type uguale a 'other' grafica i parametri della manovra: controlli, massa e velocità.

[T, V0, V1, DIST, PP] = [ASTROTimeEstimation](#)(ast0, ast1, Tbegin, A)

Restituisce il tempo stimato T per il volo da ast0 ad ast1, che inizia a Tbegin e con accelerazione media A. Restituisce inoltre i moduli delle velocità iniziali e finali, la distanza percorsa e la traiettoria PP in coordinate cartesiane.