Progetto 43 ODE: Moto di tre corpi in orbita

La seguente equazione differenziale descrive il moto di un corpo in orbita attorno a due altri corpi molto più pesanti. Un esempio potrebbe essere la capsula Apollo in orbita intorno alla terra e alla luna. I tre corpi determinano un piano nello spazio e fissiamo su questo piano un sistema di coordinate come segue. L'asse x è la retta che congiunge i due corpi pesanti, l'origine viene posta nel loro baricentro e la loro distanza è presa come unità di misura. Quindi se ρ è il rapporto fra la massa della luna e quella della terra, allora la luna e la terra si trovano nei punti di coordinate (1-ρ; 0) e (-ρ; 0), rispettivamente e il sistema di coordinate si muove in accordo con la luna che ruota intorno alla terra. Si suppone che il terzo corpo, l'Apollo, abbia massa trascurabile rispetto ai primi due e che la sua posizione sia una funzione del tempo (x(t); y(t)). Le equazioni che governano il moto dell'Apollo possono essere dedotte dalla legge del moto di Newton e dalla legge di gravitazione. I termini con le derivate prime nell'equazione provengono dal moto del sistema di coordinate ruotante.

$$x'' = 2y' + x - \frac{\rho^*(x+\rho)}{r_1^3} - \frac{\rho \quad (x-\rho^*)}{r_2^3}$$

$$y'' = -2x' + y - \frac{\rho^* y}{r_1^3} - \frac{\rho \quad y}{r_2^3}$$

$$r_1 \quad = ((x+\rho)^2 + y^2)^{1/2}, \quad r_2 \quad = ((x-\rho^*)^2 + y^2)^{1/2}$$

ρ=1/82.45,   ρ*=1-ρ

Si studiano le soluzioni periodiche di questo problema.
E' noto che le seguenti condizioni iniziali

x(0) = 1.2   x'(0) = 0
y(0) = 0   y'(0) = -1.04935751

danno luogo ad una soluzione periodica di periodo T = 6.19216933.

1. Ricondurre il sistema di 2 equazioni differenziali del secondo ordine ad un sistema di 4 equazioni differenziali del primo ordine.
2. Usare le function di Matlab **ode45** oppure **ode23s** per calcolare la soluzione.
3. Riportare l'orbita dell'Apollo in un grafico.
4. Osservato che con i valori di default l'orbita non è perfettamente periodica (modificare il tempo finale di integrazione), cambiare la tolleranza relativa per l'errore ponendo nelle opzioni RelTol=1.e-5.

UTILIZZO di EVENT:

5. Determinare il momento in cui l'Apollo ritorna il più possibile vicino al punto iniziale y0 e inizia ad allontanarsi, quindi termina l'integrazione.
6. Determinare il momento in cui l'Apollo è il più lontano possibile dal punto iniziale ed inizia ad avvicinarsi nuovamente. Aggiungere al grafico.

Sia d(t) il vettore differenza tra posizione corrente e iniziale d(t)=(dx(t),dy(t)), la distanza è data da $||d(t)||^2$ =<d(t),d(t)>. Determinare gli zeri di g(t,y)= Grad ($||d(t)||^2$). Se allo zero si arriva dall'alto allora il punto critico è un minimo, se si arriva dal basso allora il punto è un massimo.

```
function [value,isterminal,direction] = events(t,y,y0)
% Event function -- y0 shared with the outer function.
% Locate the time when the object returns closest to the initial point y0
% and starts to move away, and stop integration.  Also locate the time when
% the object is farthest from the initial point y0 and starts to move closer.
%
% The current distance of the body is
%
%   DSQ = (y(1)-y0(1))^2 + (y(2)-y0(2))^2 = <y(1:2)-y0(1:2),y(1:2)-y0(1:2)>
%
% A local minimum of DSQ occurs when d/dt DSQ crosses zero heading in
% the positive direction.  We can compute d/dt DSQ as
%
%   d/dt DSQ = 2*(y(1:2)-y0)'*dy(1:2)/dt = 2*(y(1:2)-y0)'*y(3:4)
%
% y0 is shared with the outer function.

  dDSQdt = 2 * ((y(1:2)-y0(1:2))' * y(3:4));
  value = [dDSQdt; dDSQdt];
  isterminal = [1;  0];         % stop at local minimum
  direction  = [1; -1];         % [local minimum, local maximum]
end
```

Suggerimenti Usare le seguenti options per produrre il grafico desiderato:

options=odeset('OutputFcn','odephas2')