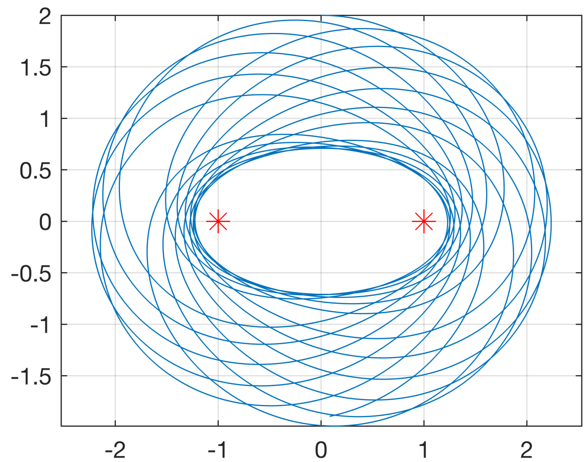
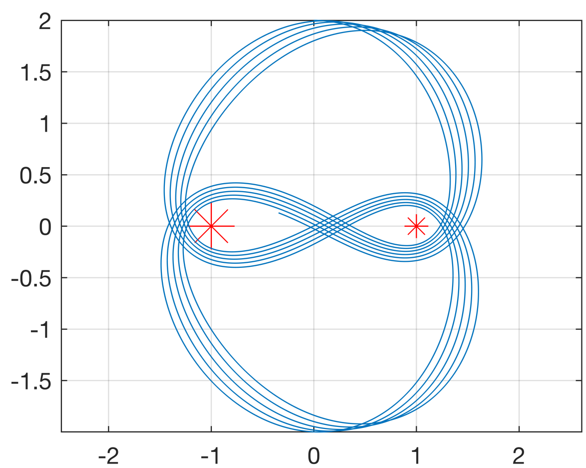
MECH 221 Computer Lab 5

Numerical Solutions of Euler's 3-Body Problem




```
>> three_body_euler(1,[1,0],1,[-1,0],[0,5,2,0],[0,20])
```




```
>> three_body_euler(1,[1,0],2,[-1,0],[0,5,2,0],[0,10])
```

Introduction

Euler's 3-body problem describes the motion of a planet orbiting two stars which are fixed in space such that the planet and stars lie in the same 2D plane. Admittedly, this is an unrealistic situation because the gravitational fields of the stars would pull them together. Nonetheless, the differential equations to be solved give rise to many different kinds of trajectories. For more about Euler's 3-body problem, see

`https://en.wikipedia.org/wiki/Euler%27s_three-body_problem`

For more information about the general three-body problem, see

`https://en.wikipedia.org/wiki/Three-body_problem`.

Instructions

Write a function called `three_body_euler` which plots numerical solutions to Euler's 3-body problem. In particular, the function should plot the trajectory of a planet orbiting two stars, Star 1 and Star 2, which have fixed positions in the coordinate system. Distance is measured in AU (astronomical units, 1 AU is roughly the distance from the Earth to the Sun), time is measured in years, mass is measured in solar mass (multiples of the Sun's mass) and the gravitational constant is $G = 4\pi^2$. The function has 6 input parameters:

```
function three_body_euler(m1,S1,m2,S2,P,tspan)
% Plot the trajectory of a planet orbiting two stars fixed in space
```

- [ ] `m1` is the mass of Star 1
- [ ] `S1` is a vector of length 2:
  - [ ] `S1(1)` is the x-position of Star 1
  - [ ] `S1(2)` is the y-position of Star 1
- [ ] `m2` is the mass of Star 2
- [ ] `S2` is a vector of length 2
  - [ ] `S2(1)` is the x-position of Star 2
  - [ ] `S2(2)` is the y-position of Star 2
- [ ] `P` is a vector of length 4:
  - [ ] `P(1)` is the initial x-position of the planet at time `tspan(1)`
  - [ ] `P(2)` is the initial x-velocity of the planet at time `tspan(1)`
  - [ ] `P(3)` is the initial y-position of the planet at time `tspan(1)`
  - [ ] `P(4)` is the initial y-velocity of the planet at time `tspan(1)`
- [ ] `tspan` is a vector $[t_0, t_f]$ of length 2, the interval of integration

When you are satisfied with your function, submit your M-file (called `three_body_euler.m`) to Connect. **Optional Bonus Challenge!** Create an animation of the planet as it travels through its trajectory.

Hints

The force of gravity between two masses m_1 and m_2 is given by

$$F = \frac{Gm_1m_2}{d^2}$$

where G is the gravitational constant and d is the distance between them. Suppose Star 1 (with mass m_{S_1} measured in solar mass) is fixed at (x_{S_1}, y_{S_1}) and Star 2 (with mass m_{S_2} measured in solar mass) is fixed at (x_{S_2}, y_{S_2}) (where the distances are measured in AU). The gravitational constant is $G = 4\pi^2$ when using these units. The trajectory $(x_P(t), y_P(t))$ of a planet (over time measured in years) orbiting both stars satisfies the system of differential equations

$$\begin{aligned}
\ddot{x}_P &= 4\pi^2 m_{S_1} \frac{(x_{S_1} - x_P)}{((x_{S_1} - x_P)^2 + (y_{S_1} - y_P)^2)^{3/2}} + 4\pi^2 m_{S_2} \frac{(x_{S_2} - x_P)}{((x_{S_2} - x_P)^2 + (y_{S_2} - y_P)^2)^{3/2}} \\
\ddot{y}_P &= 4\pi^2 m_{S_1} \frac{(y_{S_1} - y_P)}{((x_{S_1} - x_P)^2 + (y_{S_1} - y_P)^2)^{3/2}} + 4\pi^2 m_{S_2} \frac{(y_{S_2} - y_P)}{((x_{S_2} - x_P)^2 + (y_{S_2} - y_P)^2)^{3/2}}
\end{aligned}$$

We can rewrite this as a system of first order equations with new variables $u_1 = x_P$, $u_2 = \dot{x}_P$, $u_3 = y_P$ and $u_4 = \dot{y}_P$

$$\begin{aligned}
\dot{u}_1 &= u_2 \\
\dot{u}_2 &= 4\pi^2 m_{S_1} \frac{(x_{S_1} - u_1)}{((x_{S_1} - u_1)^2 + (y_{S_1} - u_3)^2)^{3/2}} + 4\pi^2 m_{S_2} \frac{(x_{S_2} - u_1)}{((x_{S_2} - u_1)^2 + (y_{S_2} - u_3)^2)^{3/2}} \\
\dot{u}_3 &= u_4 \\
\dot{u}_4 &= 4\pi^2 m_{S_1} \frac{(y_{S_1} - u_3)}{((x_{S_1} - u_1)^2 + (y_{S_1} - u_3)^2)^{3/2}} + 4\pi^2 m_{S_2} \frac{(y_{S_2} - u_3)}{((x_{S_2} - u_1)^2 + (y_{S_2} - u_3)^2)^{3/2}}
\end{aligned}$$

The default settings for `ode45` are not precise enough to compute accurate approximations for this system of equations. Include options for the relative error tolerance and absolute error tolerance:

```
options = odeset('RelTol',1e-9,'AbsTol',1e-10);
[T,U] = ode45(@odefun,tspan,P,options);
```

These settings were used with `ode45` to generate the figures in the introduction.