

东北大学

三体问题的迭代法求解及系统仿真

计算物理结课论文

应物 0701 李雅雪 20070754

20010-5-17

三体问题的迭代法求解及系统仿真

摘要

本文利用计算机在数值运算方面的优势,根据经典力学方程迭代求解天体物理中的三体问题,并判定其是否为束缚态。

三体问题是天体力学中的基本模型,研究三个可视为质点的天体在万有引力作用下的运动规律问题。本文对系统中三个对象进行受力分析,利用牛顿第二定律和万有引力定律列出方程,并根据方程编写 `matlab` 程序进行迭代求解,计算出不同情况下三个天体在某时刻具体的位置和速度,并通过比较系统动能与势能的大小来进行束缚态的判定。

另外本文将 `matlab` 的 `gui` 设计及动画表示作为探究问题的良好辅助工具,设计出简洁友好的界面,并通过擦除式动画将天体的位置和速度在坐标轴上实时地表现出来,作出了在计算机上进行天体物理实验的尝试。

关键字: 三体问题; 束缚态; 迭代法; 系统仿真

目录

1. 问题重述与问题分析	4
2. 模型假设.....	5
3. 符号说明.....	5
4. 模型建立.....	6
4.1 具体时刻的状态	6
4.1.1 受力分析.....	6
4.1.2 运动规律	7
4.2 关于束缚态的判断.....	8
4.2.1 问题分析与简化.....	8
4.2.2 束缚态的判定.....	9
5. 模型求解与系统仿真.....	10
5.1. 程序及界面设计	10
5.2 模型检验.....	12
6. 模型评价与展望.....	12
参考文献.....	13
附录	14
附录 I	14
附录 II	17
附录III.....	18

1. 问题重述与问题分析

三体问题在古典力学、量子力学及天文学中都是一著名而至今尚未彻底解决的问题。天体物理中的三体问题研究三个可视为质点的天体在相互之间万有引力作用下的运动规律，即给定三个天体的质量、初始位置和初始速度，讨论它们位置和速度随时间的变化以及运动的宏观规律和全局性质。

因为系统中三个物体的位置与速度每时每刻都在变化，所以无法用与处理两体问题相似的方法将其化简。由于三体问题不能严格求解，在研究天体运动时，只能根据实际情况采用各种近似的解法。研究三体问题的方法大致可分为 3 类：第一类是分析方法，其基本原理是把天体的坐标和速度展开为时间或其它小参数的级数形式的近似分析表达式，从而讨论天体的坐标或轨道要素随时间的变化；第二类是定性方法，采用微分方程的定性理论来研究长时间内三体运动的宏观规律和全局性质；第三类是数值方法，这是直接根据微分方程的计算方法得出天体在某些时刻的具体位置和速度。这三类方法各有利弊。

由于三体问题求解难度在于其不确定性和不可预测性，可以考虑以受力分析和经典力学方程为基础，着眼于第三类方法，利用计算机在迭代计算中的优势求其数值解

2. 模型假设

- (1) 忽略相对论效应的影响；
- (2) 所处宇宙空间是各向同性的，不考虑其它天体对系统的作用；
- (3) 忽略潮汐力、宇宙空间粒子的阻力，假设运动过程中没有能量损失；
- (4) 将天体看做质点，不考虑它们之间的相互碰撞；
- (5) 三个天体的质量相同。

3. 符号说明

G	万有引力常量；
m	天体质量；
t	运行时间；
d	天体位置；
p	天体速度；
a	天体加速度；
x, y, z	天体在直角坐标系中的位置；
u, v, w	天体在直角坐标系中的速度；
a_x, a_y, a_z	天体在直角坐标系中的加速度；
F_x, F_y, F_z	天体之间的作用力；
E_1, E_2, E_3	天体的动能与势能之差；

4. 模型建立

4.1 具体时刻的状态

因为三个天体的位置与速度是时刻变化的，现取间隔很短的时间 dt ，假设这段时间内系统参数没有变化，则可以考虑从系统每一时刻的状态计算 dt 后下一时刻的状态。在每一时刻，每个天体各有一位置与速度，三个天体即共有六个未知量，每两个天体间应用牛顿第二定律和万有引力定律各可以列出两个方程，共六个方程组成的方程组足以求解这六个未知量。如果 dt 无限小，则可以求出精确解，但在实际中这是无法实现的，因此我们将一段时间分割成很多个小份，用计算机迭代求解来得出各参量随时间的变化和系统的末状态。

4.1.1 受力分析

如图 1 所示，每个天体都受到其他两个天体万有引力的作用。为方便计算，将天体 1 对天体 2 的作用 F_{12} ，天体 2 对天体 3 的作用 F_{23} ，天体 3 对天体 1 的作用 F_{31} 分别按直角坐标系分解为 F_{12x} ， F_{12y} ， F_{12z} 、 F_{23x} ， F_{23y} ， F_{23z} 和 F_{31x} ， F_{31y} ， F_{31z} ，由于 $F_{12} = F_{21}$ ， $F_{23} = F_{32}$ ， $F_{31} = F_{13}$ ，可以用式子右边等效表示左边。

在万有引力 F_{12} 、 F_{23} 、 F_{31} 作用下，天体 1、天体 2、天体 3 的加速度分别为

$$\frac{F_{12}+F_{31}}{m}, \frac{F_{12}+F_{23}}{m}, \frac{F_{23}+F_{31}}{m}。$$

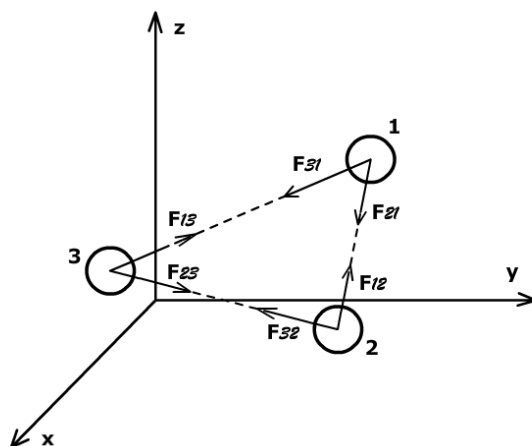


图 1 三体问题受力分析

4.1.2 运动规律

三个天体遵循以下运动规律及迭代方程：

一、万有引力定律

$$F_{12} = G \frac{m^2}{(d_1 - d_2)^2} \quad (1)$$

$$F_{23} = G \frac{m^2}{(d_2 - d_3)^2} \quad (2)$$

$$F_{31} = G \frac{m^2}{(d_3 - d_1)^2} \quad (3)$$

二、牛顿第二定律

$$a_1 = \frac{F_{12} + F_{31}}{m} \quad (4)$$

$$a_2 = \frac{F_{12} + F_{23}}{m} \quad (5)$$

$$a_3 = \frac{F_{23} + F_{31}}{m} \quad (6)$$

三、速度迭代

$$p_1' = p_1 + a_1 \times t \quad (7)$$

$$p_2' = p_2 + a_2 \times t \quad (8)$$

$$p_3' = p_3 + a_3 \times t \quad (9)$$

四、坐标值迭代

$$d_1' = d_1 + p_1 \times t + \frac{1}{2} \times a_1 \times t^2 \quad (10)$$

$$d_2' = d_2 + p_2 \times t + \frac{1}{2} \times a_2 \times t^2 \quad (11)$$

$$d_3' = d_3 + p_3 \times t + \frac{1}{2} \times a_3 \times t^2 \quad (12)$$

求解时还须将力 F 、加速度 a 、速度迭代方程、坐标值迭代方程按直角坐标系分解如附录 1。

用计算机编程求解，即可得任一时刻三个天体的位置和速度。

4.2 关于束缚态的判断

三个天体时而接近，时而远离，始终存在相互作用，没有任一天体逃逸到宇宙空间中去不再回来，这种状态称为三体运动的束缚态。为了判断给定三体系统是否为束缚态，我们通过计算每个天体在系统中的动能与势能并加以比较来得出结果。

4.2.1 问题分析与简化

如图 2 所示，在两体系统中，以其中一个天体为坐标原点建立坐标系，如果另外一个天体对它的重力势能小于这个天体在此坐标系中的动能（式（13）），则这个天体会逃逸出去，不再返回，反之若势能大于动能（式（14）），这个天体就会一直在位于坐标原点的天体周围运动。

$$\frac{1}{2}m \times v^2 - G \frac{M \times m}{r} > 0 \quad (13)$$

$$\frac{1}{2}m \times v^2 - G \frac{M \times m}{r} < 0 \quad (14)$$

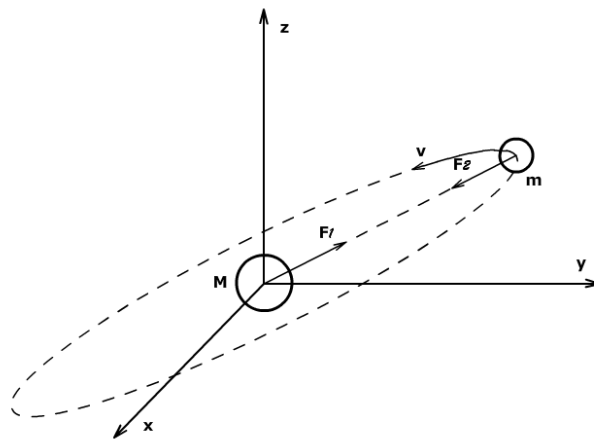


图 2 两体运动

下面考虑三体情形。如图 3，在讨论天体 1 的动能时，为简化模型，以天体 2 和天体 3 质心连线的中点为坐标原点，这样计算出的天体 1 的动能作为天体 1 在系统中的动能。由数学关系可以知道，若天体 2、天体 3 速度分别为 v_2 和 v_3 ，则它们连线中点的速度为

$$v = \frac{v_2 + v_3}{2} \quad (15)$$

天体 1 与此点的相对速度

$$\Delta v = \frac{v_2 + v_3}{2} - v_1 \quad (16)$$

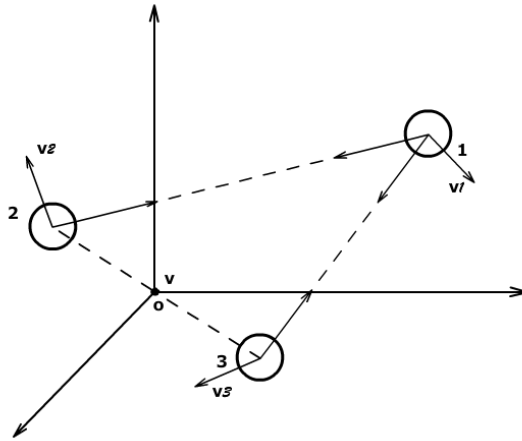


图 3 三体运动

4.2.2 束缚态的判定

天体 1 的动能

$$V_1 = \frac{1}{2} m \times \left(\frac{v_2 + v_3}{2} - v_1 \right)^2 \quad (17)$$

天体 1 的势能

$$H_1 = G \times \frac{m^2}{(d_2 - d_1)^2} + G \times \frac{m^2}{(d_1 - d_3)^2} \quad (18)$$

则

$$E_1 = V_1 - H_1 = \frac{1}{2} m \times \left(\frac{v_2 + v_3}{2} - v_1 \right)^2 - G \times \frac{m^2}{(d_2 - d_1)^2} - G \times \frac{m^2}{(d_1 - d_3)^2} \quad (19)$$

为束缚态的判定公式

若 $E_1 > 0$ ，则对于天体 1 不是束缚态，若 $E_1 < 0$ ，则对于天体 1 是束缚态。

同理有天体 2、天体 3 的判定公式

$$E_2 = V_2 - H_2 = \frac{1}{2} m \times \left(\frac{v_1 + v_3}{2} - v_2 \right)^2 - G \times \frac{m^2}{(d_2 - d_1)^2} - G \times \frac{m^2}{(d_3 - d_2)^2} \quad (20)$$

$$E_3 = V_3 - H_3 = \frac{1}{2} m \times \left(\frac{v_1 + v_2}{2} - v_3 \right)^2 - G \times \frac{m^2}{(d_1 - d_3)^2} - G \times \frac{m^2}{(d_3 - d_2)^2} \quad (21)$$

这三个公式同样也要按直角坐标系分解方可编程求解，分解结果详见附录 2。

5. 模型求解与系统仿真

5.1. 程序及界面设计

用 matlab 语言进行编程，代码与注释见附录 3，程序的图形界面如下，点击“start”，即可在弹出的对话框中输入各参数值。默认的参数值： G 为实际的万有引力常量， m 为太阳质量，行星的位置和速度都是随机设定的。

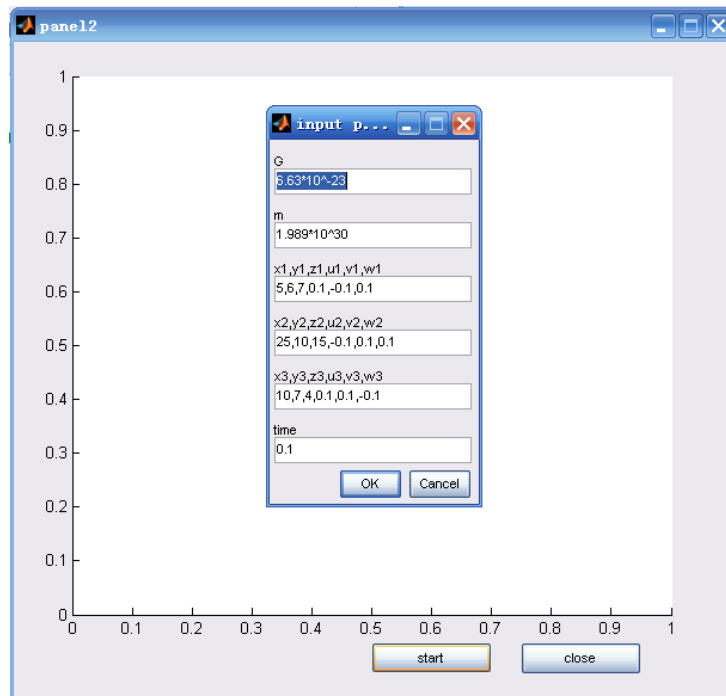


图 4 输入参数

点击“OK”，程序读取对话框中输入的参数（若不作更改则使用默认的参数值），按照公式进行迭代计算，确定每经过一个 dt 步长时三个天体的位置变化，并在坐标系中即时显示，这样我们就可以看到三个天体运动的动画（在某些情况下天体会运动到坐标轴外面去）。

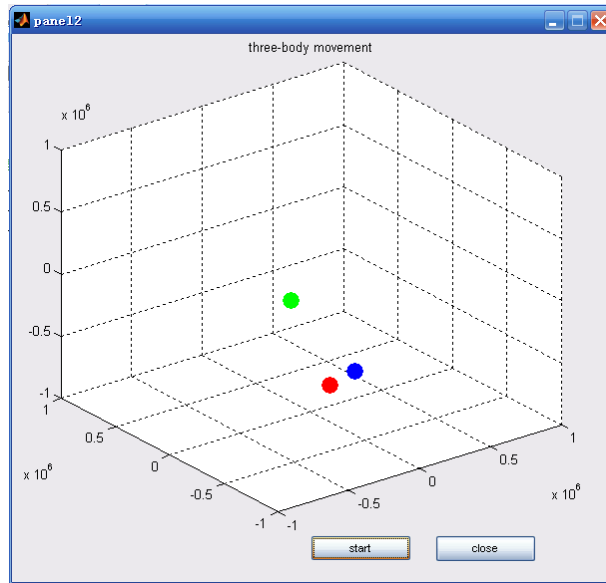


图 5 显示动画

稍等片刻，迭代结束后，在当前窗口中显示出末状态三个天体的坐标 x_1 、 y_1 、 z_1 、 x_2 、 y_2 、 z_2 、 x_3 、 y_3 、 z_3 ，并且自动计算每个天体动能与势能之差 E_1 、 E_2 、 E_3 ，若三个天体的动能都小于势能，在界面上自动弹出对话框显示 “It is an engage state, which means that the body will come back if time is long enough.”，不然则显示 “It is not an engage state, which means that the body will not come back no matter how much time is spent.”

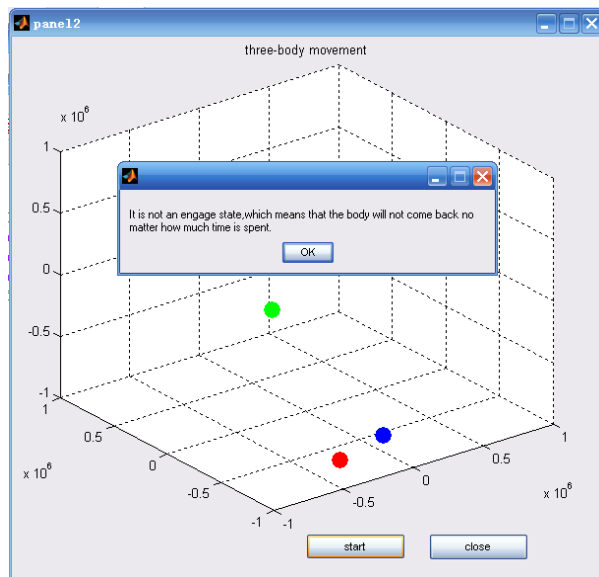


图 6 得出结果

在 Command Window 中显示求解结果：

```
x1 = -3.0776e+005
y1 = -7.0934e+005
z1 = -9.6242e+005
x2 = -3.3810e+005
y2 = -1.1578e+005
z2 = -6.0977e+003
x3 = 9.2697e+004
y3 = -5.9355e+005
z3 = -9.5631e+005
E1 = 1.5285e+040
E2 = 4.8756e+040
E3 = 1.6720e+040
```

改变位置与速度的初始值，即可得到各种不同的解，观察是否为束缚态。但注意不要使任一轴上两球的坐标值相同，因为这样会在计算中产生为 0 的分母，使求解出错。

5.2 模型检验

在对话框中改变各个参数，运行程序，发现当 G 、 m 比较大时天体运动速度加快，并且会随着距离的加大进行加速运动，当 G 、 m 比较小时天体运动速度减慢。将其中一个天体的初速度设为 0，它将以较小速度开始运动。这些都与实际情况相符，说明此模型是有实际应用价值的。

如果将 m 都设为比较小的数量级，比如 1.989×10^{19} ，三个天体初速度都设为 0，运行程序，发现小球几乎不动，当前窗口返回值：

```
x1 =5.0006, y1 =5.9817, z1 =7.0016
x2 =25.0000, y2 =9.9970, z2 =14.9996
x3 =10.0008, y3 =6.9847, z3 =4.0021
E1 =-5.5053e+015, E2 =-2.2982e+015, E3 =-5.7897e+015
```

对话框显示 “It is an engage state, which means that the body will come back if time is long enough.” 这是一个束缚态。将天体初速度设为很小的数量级，也可以出现束缚态。

6. 模型评价与展望

本文结合定量的数值计算和定性的动画表示来求解三体问题。友好的用户图形界面给使用者带来便利，通过在对话框中设定参数和在源程序中修改迭代步长、坐标轴范围和动画的播放频率，理论上可以解答任意情况下的三体问题，简便且直观。

当然，本文在建立相关模型时，也存在如下着缺陷与不足。

- (1) 在仿真过程中发现，计算结果受迭代步长的影响很大。在源程序中将步长改为 0.001，会发现小球运动轨迹与步长为 0.01 时完全不一样，可见迭代步长在很大程度上影响了运算精度；
- (2) 对实际情况的模拟不够全面，如宇宙空间尺度、天体之间的距离、运行速度等，这方面还需要查找更多的数据来完善。

同时，本文中相关模型尚有开发的余地，如引入相对论效应的影响及考虑到天体之间距离足够小时应使用微积分计算万有引力等。并且，如果计算机处理速度足够快，则可以设置很小的迭代步长和很高的画面刷新频率，这样计算出来的结果将会更加完美。

参考文献

- [1] S. 温伯格. 引力论和宇宙[M]. 北京: 科学出版社, 1980
- [2] 易照华. 天体力学基础[M]. 北京: 高等教育出版社, 1993
- [3] 陈锺贤. 计算物理学[M]. 哈尔滨: 哈尔滨工业大学出版社, 2001
- [4] 高炳坤. 三体问题的受力分析——兼谈失重[J]. 大学物理, 2004. 5
- [5] 孙义燧, 罗定军. 一般三体问题中轨道形状和位置的变化 [J]. 南京大学学报, 1980. 4
- [6] 薛定宇, 陈阳泉. 高等应用数学问题的 MATLAB 求解[M]. 北京: 清华大学出版社, 2008
- [7] 陈圭光, 毛涛涛, 王正林, 王玲. 精通 MATLAB GUI 设计[M]. 北京: 电子工业出版社, 2008. 2
- [8] 罗华飞. MATLAB GUI 设计学习手记[M]. 北京: 北京航空航天大学出版社, 2009. 8
- [9] 刘志勇. 基于 MATLAB 的动画制作[J]. 江西电力职业技术学院学报, 2008. 9

附录

附录 I

万有引力定律:

$$F_{x12} = G \frac{m^2}{(x_1 - x_2)^2}$$

$$F_{y12} = G \frac{m^2}{(y_1 - y_2)^2}$$

$$F_{z12} = G \frac{m^2}{(z_1 - z_2)^2}$$

$$F_{x23} = G \frac{m^2}{(x_2 - x_3)^2}$$

$$F_{y23} = G \frac{m^2}{(y_2 - y_3)^2}$$

$$F_{z23} = G \frac{m^2}{(z_2 - z_3)^2}$$

$$F_{x31} = G \frac{m^2}{(x_3 - x_1)^2}$$

$$F_{y31} = G \frac{m^2}{(y_3 - y_1)^2}$$

$$F_{z31} = G \frac{m^2}{(z_3 - z_1)^2}$$

牛顿第二定律:

$$a_{x1} = \frac{F_{x12} + F_{x31}}{m}$$

$$a_{y1} = \frac{F_{y12} + F_{y31}}{m}$$

$$a_{z1} = \frac{F_{z12} + F_{z31}}{m}$$

$$a_{x2} = \frac{F_{12} + F_{23}}{m}$$

$$a_{y2} = \frac{F_{12} + F_{23}}{m}$$

$$a_{z2} = \frac{F_{12} + F_{23}}{m}$$

$$a_{x3} = \frac{F_{23} + F_{31}}{m}$$

$$a_{y3} = \frac{F_{23} + F_{31}}{m}$$

$$a_{z3} = \frac{F_{23} + F_{31}}{m}$$

速度迭代:

$$u_1 = u_1 + a_{x1} \times t$$

$$v_1 = v_1 + a_{y1} \times t$$

$$w_1 = w_1 + a_{z1} \times t$$

$$u_2 = u_2 + a_{x2} \times t$$

$$v_2 = v_2 + a_{y2} \times t$$

$$w_2 = w_2 + a_{z2} \times t$$

$$u_3 = u_3 + a_{x3} \times t$$

$$v_3 = v_3 + a_{y3} \times t$$

$$w_3 = w_3 + a_{z3} \times t$$

坐标值迭代:

$$x_1 = x_1 + u_1 \times t + \frac{1}{2} \times a_{x1} \times t^2$$

$$y_1 = y_1 + v_1 \times t + \frac{1}{2} \times a_{y1} \times t^2$$

$$z_1 = z_1 + w_1 \times t + \frac{1}{2} \times a_{z1} \times t^2$$

$$x_2 = x_2 + u_2 \times t + \frac{1}{2} \times a_{x2} \times t^2$$

$$y_2 = y_2 + v_2 \times t + \frac{1}{2} \times a_{y2} \times t^2$$

$$z_2 = z_2 + w_2 \times t + \frac{1}{2} \times a_{z2} \times t^2$$

$$x_3 = x_3 + u_3 \times t + \frac{1}{2} \times a_{x3} \times t^2$$

$$y_3 = y_3 + v_3 \times t + \frac{1}{2} \times a_{y3} \times t^2$$

$$z_3 = z_3 + w_3 \times t + \frac{1}{2} \times a_{z3} \times t^2$$

附录 II

$$\begin{aligned}
 E_1 &= \frac{1}{2}m \times \left[\left(\frac{u_2 + u_3}{2} - u_1 \right)^2 + \left(\frac{v_2 + v_3}{2} - v_1 \right)^2 + \left(\frac{w_2 + w_3}{2} - w_1 \right)^2 \right] - G \\
 &\quad \times \frac{m^2}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}} - G \\
 &\quad \times \frac{m^2}{\sqrt{(x_1 - x_3)^2 + (y_1 - y_3)^2 + (z_1 - z_3)^2}} \\
 E_2 &= \frac{1}{2}m \times \left[\left(\frac{u_1 + u_3}{2} - u_2 \right)^2 + \left(\frac{v_1 + v_3}{2} - v_2 \right)^2 + \left(\frac{w_1 + w_3}{2} - w_2 \right)^2 \right] - G \\
 &\quad \times \frac{m^2}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}} - G \\
 &\quad \times \frac{m^2}{\sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2 + (z_3 - z_2)^2}} \\
 E_3 &= \frac{1}{2}m \times \left[\left(\frac{u_1 + u_2}{2} - u_3 \right)^2 + \left(\frac{v_1 + v_2}{2} - v_3 \right)^2 + \left(\frac{w_1 + w_2}{2} - w_3 \right)^2 \right] - G \\
 &\quad \times \frac{m^2}{\sqrt{(x_1 - x_3)^2 + (y_1 - y_3)^2 + (z_1 - z_3)^2}} - G \\
 &\quad \times \frac{m^2}{\sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2 + (z_3 - z_2)^2}}
 \end{aligned}$$

附录III

```

function varargout = panel2(varargin)
% PANEL2 M-file for panel2.fig
%     PANEL2, by itself, creates a new PANEL2 or raises the existing
%     singleton*.
%
%     H = PANEL2 returns the handle to a new PANEL2 or the handle to
%     the existing singleton*.
%
%     PANEL2('CALLBACK', hObject,eventData,handles,...) calls the local
%     function named CALLBACK in PANEL2.M with the given input arguments.
%
%     PANEL2('Property','Value',...) creates a new PANEL2 or raises the
%     existing singleton*. Starting from the left, property value pairs
are
%     applied to the GUI before panel2_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to panel2_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only
one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help panel2

% Last Modified by GUIDE v2.5 16-May-2010 14:02:49

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @panel2_OpeningFcn, ...
                  'gui_OutputFcn',  @panel2_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin

```

```

    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before panel2 is made visible.
function panel2_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to panel2 (see VARARGIN)

% Choose default command line output for panel2
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes panel2 wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = panel2_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton_start.
function pushbutton_start_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton_start (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
str={'G','m','x1,y1,z1,u1,v1,w1','x2,y2,z2,u2,v2,w2','x3,y3,z3,u3,v3,w3','time'};
%设定输入对话框中供输入的变量名称

```

```

strdef={'6.63*10^-23','1.989*10^30'
,'5,6,7,0.1,-0.1,0.1','25,10,15,-0.1,0.1,0.1','10,7,4,0.1,0.1,-0.1','
0.1'};
%默认显示与使用的参数
DataInput=InputDlg(str,'input parameters',1,strdef);
% 对话框的标题
    Data=str2num(DataInput{1,1});
    g=Data(1,1);
    Data=str2num(DataInput{2,1});
    m=Data(1,1);
    Data=str2num(DataInput{3,1});
    x1=Data(1,1); y1=Data(1,2); z1=Data(1,3);
    u1=Data(1,4); v1=Data(1,5); w1=Data(1,6);
    Data=str2num(DataInput{4,1});
    x2=Data(1,1); y2=Data(1,2); z2=Data(1,3);
    u2=Data(1,4); v2=Data(1,5); w2=Data(1,6);
    Data=str2num(DataInput{5,1});
    x3=Data(1,1); y3=Data(1,2); z3=Data(1,3);
    u3=Data(1,4); v3=Data(1,5); w3=Data(1,6);
    Data=str2num(DataInput{6,1});
    time=Data(1,1);
%获得默认或用户输入的参数值，包括万有引力常量、小球质量、初始位置与速度、运动时间
h=line('Color',[1 0
0],'Marker','.', 'MarkerSize',40,'erasemode','normal');
k=line('Color',[0 1
0],'Marker','.', 'MarkerSize',40,'erasemode','normal');
l=line('Color',[0 0
1],'Marker','.', 'MarkerSize',40,'erasemode','normal');
%设定小球颜色、点型、大小、擦除模式，每次擦除时重置整个画面
grid on
%加网格线
axis([-10^6 10^6 -10^6 10^6 -10^6 10^6])
%坐标轴显示的空间范围
title('three-body movement')
%动画标题
for t=0:0.001:time
%以0.01的时间精度迭代
fx12=g*m.^2./(x1-x2).^3.*abs(x1-x2);
fx23=g*m.^2./(x2-x3).^3.*abs(x2-x3);
fx31=g*m.^2./(x3-x1).^3.*abs(x3-x1);
%万有引力定律
ax1=(fx12+fx31)/m;
ax2=(fx12+fx23)/m;
ax3=(fx23+fx31)/m;

```

```

%牛顿第二定律
u1=u1+ax1*t;
u2=u2+ax2*t;
u3=u3+ax3*t;
%速度值迭代
x1=x1+u1.*t+1/2*ax1.*t.^2;
x2=x2+u2.*t+1/2*ax2.*t.^2;
x3=x3+u3.*t+1/2*ax3.*t.^2;
%坐标值迭代
%以上为对x轴方向的计算
fy12=g*m.^2./(y1-y2).^3.*abs(y1-y2);
fy23=g*m.^2./(y2-y3).^3.*abs(y2-y3);
fy31=g*m.^2./(y3-y1).^3.*abs(y3-y1);
ay1=(fy12-fy31)/m;
ay2=(fy12-fy23)/m;
ay3=(fy23-fy31)/m;
v1=v1+ay1*t;
v2=v2+ay2*t;
v3=v3+ay3*t;
y1=y1+v1.*t+1/2*ay1.*t.^2;
y2=y2+v2.*t+1/2*ay2.*t.^2;
y3=y3+v3.*t+1/2*ay3.*t.^2;
%以上为对y轴方向的计算
fz12=g*m.^2./(z1-z2).^3.*abs(z1-z2);
fz23=g*m.^2./(z2-z3).^3.*abs(z2-z3);
fz31=g*m.^2./(z3-z1).^3.*abs(z3-z1);
az1=(fz12-fz31)/m;
az2=(fz12-fz23)/m;
az3=(fz23-fz31)/m;
w1=w1+az1*t;
w2=w2+az2*t;
w3=w3+az3*t;
z1=z1+w1.*t+1/2*az1.*t.^2;
z2=z2+w2.*t+1/2*az2.*t.^2;
z3=z3+w3.*t+1/2*az3.*t.^2;
%以上为对z轴方向的计算
set(h,'xdata',x1,'ydata',y1,'zdata',z1);
set(k,'xdata',x2,'ydata',y2,'zdata',z2);
set(l,'xdata',x3,'ydata',y3,'zdata',z3);
%重置三个小球的位置
drawnow
%刷新画面
pause(0.01)
%控制速度

```

```

end
x1
y1
z1
x2
y2
z2
x3
y3
z3
%在Command Window中显示每个球末状态的坐标
E1=1/2*m*((u2+u3)/2-u1).^2+((v2+v3)/2-v1).^2+((w2+w3)/2-w1).^2)-g*m.
.^2/sqrt((x2-x1).^2+(y2-y1).^2+(z2-z1).^2)-g*m.^2/sqrt((x1-x3).^2+(y1-
y3).^2+(z1-z3).^2)
E2=1/2*m*((u1+u3)/2-u2).^2+((v1+v3)/2-v2).^2+((w1+w3)/2-w2).^2)-g*m.
.^2/sqrt((x2-x1).^2+(y2-y1).^2+(z2-z1).^2)-g*m.^2/sqrt((x3-x2).^2+(y3-
y2).^2+(z3-z2).^2)
E3=1/2*m*((u1+u2)/2-u3).^2+((v1+v2)/2-v3).^2+((w1+w2)/2-w3).^2)-g*m.
.^2/sqrt((x1-x3).^2+(y1-y3).^2+(z1-z3).^2)-g*m.^2/sqrt((x3-x2).^2+(y3-
y2).^2+(z3-z2).^2)
%计算每个小球动能与势能之差并显示出来
if E1<0&E2<0&E3<0
    hmb=msgbox('It is an engage state,which means that the body will come
back if time is long enough.','','')
%如果所有小球动能都小于势能，则是束缚态
else
    hmb=msgbox('It is not an engage state,which means that the body will
not come back no matter how much time is spent.','','');
%如果并非所有小球动能都小于势能，则不是束缚态
end

% --- Executes on button press in close.
function close_Callback(hObject, eventdata, handles)
% hObject    handle to close (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close
%关闭界面

```