基于HTML5的“三体”运动仿真

16302010009 张皓通

摘要：本文利用HTML5设计了二维下，基于经典力学的三体运动数值模拟，建立三体运动模型。程序根据质点的初始运动参数，运用数值模拟迭代出之后任意时刻各个质点的运动参数，并将结果可视化。本程序可以用于质点力学或基础天体力学的教学演示等相关领域。

一. 引言：

大约一百年前，数学家希尔伯特提出了“完美数学问题”准则：问题要既能简明扼要表达出来，但问题的解决方法却又非常困难因此需要全新的方法去解决。希尔伯特举了两个例子，第一个是费马猜想，第二个就是N体问题[1]，具体来说，就是空间中N个可视为质点的恒质星体在某一时刻运动状态均已知，求之后任意时刻任意时刻此N个质点的速度与位置。当N<3时，牛顿力学就可将这个问题完美解决。但当N大于等于3时，经典力学体系便无法求得准确解。本文模拟N=3时的情形，三体系统[2]是一个混沌系统，如图1，并不存在解析解。其重要特征之一就是误差的累积性，即计算数据的无理性以及对微扰的敏感性。因此，三体系统并不存在一个普遍的解析解，甚至连预测长期的行为都是困难的，这也是三体问题成为困扰了数学家几个世纪之久的重要原因。

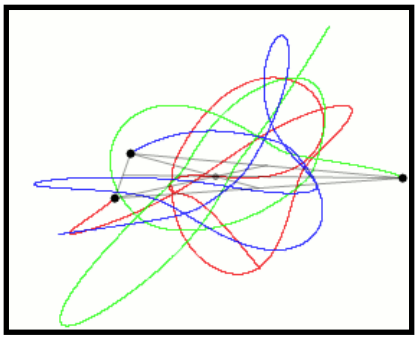

图1-三体系统往往是一个混沌系统

本实验拟在经典力学的框架中，通过迭代进行对三体运动的模拟，并进行可视化。当然，三体系统的混沌性质决定了这种方式仅仅能在宏观角度粗糙的模拟三体运动，且可信度随时间下降。但三体也有至少十六个稳定解[3]的存在，其中也有不少存在解析解，本文也将模拟几种稳定解的情形。

二. 实验原理：

设空间中存在有三个质量、初始位置、初始初速度已知的三个质点，根据牛顿运动定律，可以得到加速度a[4]：

$$\ddot{\mathbf{r}}_1 = -Gm_2\frac{\mathbf{r}_1-\mathbf{r}_2}{\left|\mathbf{r}_1-\mathbf{r}_2\right|^3} - Gm_3\frac{\mathbf{r}_1-\mathbf{r}_3}{\left|\mathbf{r}_1-\mathbf{r}_3\right|^3}$$
$$\ddot{\mathbf{r}}_2 = -Gm_3\frac{\mathbf{r}_2-\mathbf{r}_3}{\left|\mathbf{r}_2-\mathbf{r}_3\right|^3} - Gm_1\frac{\mathbf{r}_2-\mathbf{r}_1}{\left|\mathbf{r}_2-\mathbf{r}_1\right|^3}$$
$$\ddot{\mathbf{r}}_3 = -Gm_1\frac{\mathbf{r}_3-\mathbf{r}_1}{\left|\mathbf{r}_3-\mathbf{r}_1\right|^3} - Gm_2\frac{\mathbf{r}_3-\mathbf{r}_2}{\left|\mathbf{r}_3-\mathbf{r}_2\right|^3}$$

其中，r指质点的位移，为一矢量，G是万有引力常量，m为质点质量。我们在二维平面中进行矢量分解，如

$$\mathbf{r}_1-\mathbf{r}_2 = \sqrt{\left(r_{1x}-r_{2x}\right)^2+\left(r_{1y}-r_{2y}\right)^2}$$

在程序中，使用迭代的方法计算速度和位置。

$$v_1 = v_{10} + a_1\Delta t$$
$$v_2 = v_{20} + a_2\Delta t$$
$$v_3 = v_{30} + a_3\Delta t$$
$$x_1 = x_{10} + v_1\Delta t$$
$$x_2 = x_{20} + v_2\Delta t$$
$$x_3 = x_{30} + v_3\Delta t$$

以上均为矢量，需要做到x轴和y轴的矢量分解。

三. 实验过程

整个程序代码的编写主要分为HTML5界面和Javascript程序，HTML5界面负责程序的总体结构框架、文字说明与用户交互。

HTML5（Hyper Text Mark Language）是主流的编写网页的语言。使用其编写物理课件的一大好处就是网页格式的物理课件更加方便于分享、演示与传播。但相比Python、Matlab等一系列已经拥有较为完善库的工具，HTML5的库比较少，功能不完善，实现大多需要自己手写而没有封装好的函数可以使用。

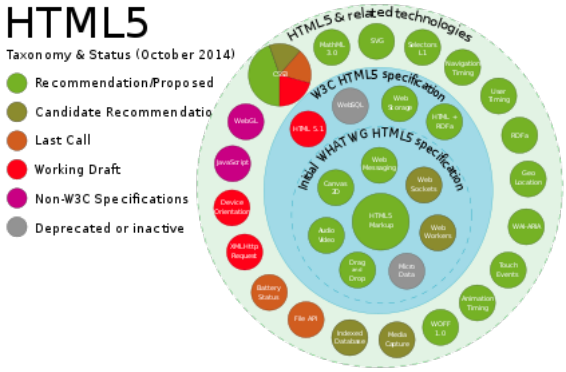

图2-HTML5的主要API库（科学库较少）

对于HTML5部分，总体框架和文字说明在老师布置的教程之中已经给出，只需要再做一些微小的修改即可。我认为主要的难点是我设计了很多用户交互的功能，此部分的设计与Debug占用了该程序大部分的精力。

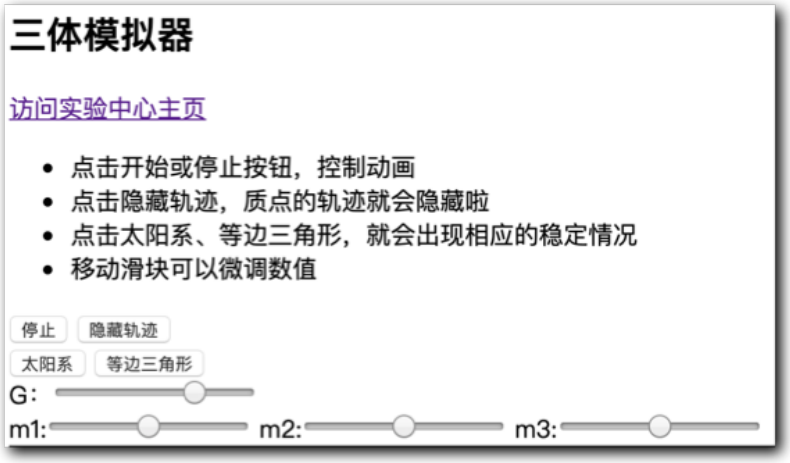

图3-HTML5界面中的一部分

JavaScript部分是实现三体运动轨迹的主要部分，算法在第二部分：

首先要定义全局变量，如图4:

```
var animateButton = document.getElementById("animateButton"),
    paused = false;
var animateButton1 = document.getElementById("animateButton1"),
    trace = true;
var animateButton2 = document.getElementById("animateButton2"),
    special = 0;
var animateButton3 = document.getElementById("animateButton3");
var radiusRange = document.getElementById("radiusRange");

var inputm1 = document.getElementById("inputm1");
var inputm2 = document.getElementById("inputm2");
var inputm3 = document.getElementById("inputm3");

var myCanvas = document.getElementById("space");
var ball1 = myCanvas.getContext("2d");
var ball2 = myCanvas.getContext("2d");
var ball3 = myCanvas.getContext("2d");
var m = [0, 5, 5, 5];

var x1 = 100, y1 = 370;
var x2 = 200, y2 = 200;
var x3 = 300, y3 = 350;

var vx1 = 100, vy1 = 50;
var vx2 = 30, vy2 = -40;
var vx3 = 20, vy3 = -30;

var ax1 = 0, ay1 = 0;
var ax2 = 0, ay2 = 0;
var ax3 = 0, ay3 = 0;

var G = 6.67 * Math.pow(10,4.5),
    t = 0.001;
```

图4-JavaScript中的全局变量

主要设计逻辑如下：

- 主体部分是一层大循环，首先清空画布。
- 当开始按钮为关闭时，则会直接跳至循环结束；当开始按钮开启时，在第一次循环中，将会根据小球的初始位置和初始速度进行初始化，随后进行迭代。
- 更新画布，进入下一次循环。

高级功能：

- 显示/隐藏轨迹：当隐藏轨迹时，每次循环开始时会清空画布；显示轨迹时，将不会清空画布，因此轨迹会变得可见。
- 参数改变：可在程序运行过程中，改变G、各个星体质量，来观察轨迹的改变。实现方法是在循环开始时，通过滑块触发监视器来改变全局变量。
- 特殊情况：程序设定了两个具有稳定解的特殊情况，分别为太阳系情形和等边三角情形。

四. 实验结果与分析

在仿真模拟中，可通过稳定解来验证模型的准确性：

太阳系情形：此种情况下，有一质点质量极大，另外两个质量较小的质点围绕大质量质点运动，轨迹如图5：

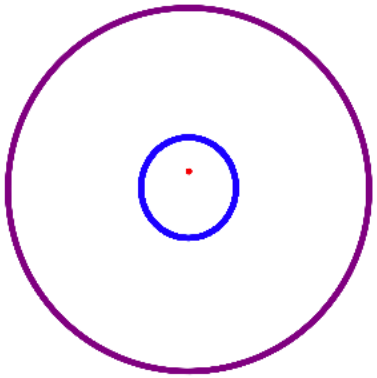

图5-太阳系情形下的三体轨迹

等边三角形情形：在任意时刻，三个质点都构成一个等边三角形，轨迹如图6：

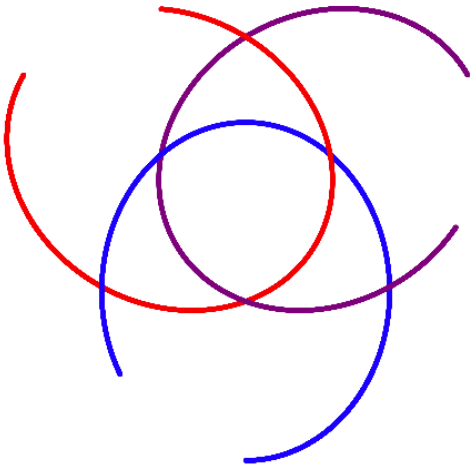

图6-等边三角形轨迹

但是，由于三体系统的误差积累性，该模拟在45s后基本就会出现偏离稳定解的行为，但在这之前还是能够基本模拟三个质点的理论轨迹，这说明该仿真模型是成功的，也侧面论证了三体系统是一个混沌系统，对于微扰极其敏感的特点。

五. 实验结论

本实验成功建立了基于经典力学的二维自由空间三体模型，并利用了HTML5与JavaScript完成了可视化的实现，并验证了几个三体系统的特殊解。本文粗略验证了三体系统属于经典混沌系统的事实，直观上体现了该系统无解析解且有误差积累性的特点。

六. 参考文献

[1] Florin Diacu. "The Solution of the *n*-body Problem", *The Mathematical Intelligencer*, 1996.
[2]Barrow-Green, June (1997). *Poincaré and the Three Body Problem*. American Mathematical Soc. pp. 8–12. Bibcode:1997ptbp.book.....B. ISBN 978-0-8218-0367-7.
[3]Barrow-Green, J. (2010). The dramatic episode of Sundman, Historia Mathematica 37, pp. 164–203.
[4]中文维基百科 三体系统

附录

```
<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8">
    <title>三体</title>
    <style type="text/css">
        canvas {
            border-bottom: 5px solid black;
            border-top: 5px solid black;
            border-left: 5px solid black;
            border-right: 5px solid black;
        }
    </style>
</head>

<body>
<h2>三体模拟器</h2>

<a href="http://phylab.fudan.edu.cn" target="_blank">访问实验中心主页</a>

<ul>
    <li>点击开始或停止按钮，控制动画</li>
    <li>点击隐藏轨迹，质点的轨迹就会隐藏啦</li>
    <li>点击太阳系、等边三角形，就会出现相应的稳定情况</li>
    <li>移动滑块可以微调数值</li>
</ul>

<input id="animateButton" class="controls" type="button" value="开始"/>
<input id="animateButton1" class="controls" type="button" value="隐藏轨迹"/>
<br />
<input id="animateButton2" class="controls" type="button" value="太阳系"/>
```

```
<input id="animateButton3" class="controls" type="button" value="等边三角形"/>
<br />
G:
<input id="radiusRange" class="controls" type="range" min="-100" max="100" value="
45"/>
<br />
m1:<input id="inputm1" class="controls" type="range" min="0" max="10" value="5"/
>
m2:<input id="inputm2" class="controls" type="range" min="0" max="10" value="5"/
>
m3:<input id="inputm3" class="controls" type="range" min="0" max="10" value="5"/
>
<br />
<canvas id="space" width="1000" height="600"></canvas>
<script type="text/javascript">
    var animateButton = document.getElementById("animateButton"),
        paused = false;
    var animateButton1 = document.getElementById("animateButton1"),
        trace = true;
    var animateButton2 = document.getElementById("animateButton2"),
        special = 0;
    var animateButton3 = document.getElementById("animateButton3");
    var radiusRange = document.getElementById("radiusRange");

    var inputm1 = document.getElementById("inputm1");
    var inputm2 = document.getElementById("inputm2");
    var inputm3 = document.getElementById("inputm3");

    var myCanvas = document.getElementById("space");
    var ball1 = myCanvas.getContext("2d");
    var ball2 = myCanvas.getContext("2d");
    var ball3 = myCanvas.getContext("2d");
    var m = [0, 5, 5, 5];

    var x1 = 100, y1 = 370;
    var x2 = 200, y2 = 200;
    var x3 = 300, y3 = 350;

    var vx1 = 100, vy1 = 50;
    var vx2 = 30, vy2 = -40;
    var vx3 = 20, vy3 = -30;

    var ax1 = 0, ay1 = 0;
    var ax2 = 0, ay2 = 0;
    var ax3 = 0, ay3 = 0;

    var G = 6.67 * Math.pow(10,4.5),
        t = 0.001;


    function gravity() {
        if(!paused){
```

```
            if(!trace){
                ball2.clearRect(0, 0, 1000, 600);
            }
            //太阳系
            if(special == 1){
                special = 0;
                ball2.clearRect(0, 0, 1000, 600);
                m = [0, 1000000, 10, 10];
                x1 = 500, y1 = 300;
                x2 = 500, y2 = 350;
                x3 = 500, y3 = 450;

                vx1 = 0, vy1 = 0;
                vx2 = 300, vy2 = 0;
                vx3 = 200, vy3 = 0;

                ax1 = 0, ay1 = 0;
                ax2 = 0, ay2 = 0;
                ax3 = 0, ay3 = 0;

                G = 6.67 * Math.pow(10,0), t = 0.001;
            }
            //等边三角形
            if(special == 2){
                special = 0;
                ball2.clearRect(0, 0, 1000, 600);
                m = [0, 2, 2, 2];
                x1 = 300, y1 = 150;
                x2 = 500, y2 = 496.4102;
                x3 = 700, y3 = 150;

                vx1 = -20, vy1 = 34.641;
                vx2 = 40, vy2 = 0;
                vx3 = -20, vy3 = -34.641;

                ax1 = 0, ay1 = 0;
                ax2 = 0, ay2 = 0;
                ax3 = 0, ay3 = 0;

                G = 6.67 * Math.pow(10,5), t = 0.001;
            }

            ball1.beginPath();
            ball1.arc(x1, y1, 2, 0, Math.PI * 2);
            ball1.closePath();
            ball1.fillStyle = "red";
            ball1.fill();

            ball2.beginPath();
            ball2.arc(x2, y2, 2, 0, Math.PI * 2);
            ball2.closePath();
            ball2.fillStyle = "blue";
```

```
            ball2.fill();

            ball3.beginPath();
            ball3.arc(x3, y3, 2, 0, Math.PI * 2);
            ball3.closePath();
            ball3.fillStyle = "purple";
            ball3.fill();

            ax1 = G * m[2] * (x2 - x1)/ Math.pow((Math.sqrt(Math.pow((x1 - x2),2) +Math.pow((y1 - y2),2))),3) + G * m[3]* (x3 - x1)/ Math.pow((Math.sqrt(Math.pow((x1 - x3),2)+ Math.pow((y1 - y3),2))),3);
            ay1 = G * m[2] * (y2 - y1)/ Math.pow((Math.sqrt(Math.pow((x1 - x2),2) +Math.pow((y1 - y2),2))),3) + G * m[3]* (y3 - y1)/ Math.pow((Math.sqrt(Math.pow((x1 - x3),2)+ Math.pow((y1 - y3),2))),3);
            ax2 = G * m[1] * (x1 - x2)/ Math.pow((Math.sqrt(Math.pow((x1 - x2),2) +Math.pow((y1 - y2),2))),3) + G * m[3]* (x3 - x2)/ Math.pow((Math.sqrt(Math.pow((x2 - x3),2)+ Math.pow((y3 - y2),2))),3);
            ay2 = G * m[1] * (y1 - y2)/ Math.pow((Math.sqrt(Math.pow((x1 - x2),2) +Math.pow((y1 - y2),2))),3) + G * m[3]* (y3 - y2)/ Math.pow((Math.sqrt(Math.pow((x3 - x2),2)+ Math.pow((y3 - y2),2))),3);
            ax3 = G * m[1] * (x1 - x3)/ Math.pow((Math.sqrt(Math.pow((x1 - x3),2) +Math.pow((y1 - y3),2))),3) + G * m[2]* (x2 - x3)/ Math.pow((Math.sqrt(Math.pow((x3 - x2),2)+ Math.pow((y2 - y3),2))),3);
            ay3 = G * m[1] * (y1 - y3)/ Math.pow((Math.sqrt(Math.pow((x1 - x3),2) +Math.pow((y1 - y3),2))),3) + G * m[2]* (y2 - y3)/ Math.pow((Math.sqrt(Math.pow((x3 - x2),2)+ Math.pow((y2 - y3),2))),3);

            vx1 += ax1 * t;
            vy1 += ay1 * t;
            vx2 += ax2 * t;
            vy2 += ay2 * t;
            vx3 += ax3 * t;
            vy3 += ay3 * t;

            x1 += vx1 * t;
            y1 += vy1 * t;
            x2 += vx2 * t;
            y2 += vy2 * t;
            x3 += vx3 * t;
            y3 += vy3 * t;
        }
        window.setTimeout(gravity, t );
        //window.requestAnimationFrame(gravity);
    }

    /*function myFunction() {
        gravity();
    }
    document.getElementById("myBtn").addEventListener("click", myFunction);*/
    animateButton.onclick = function(e) {
        if (paused) {
            paused = false;
```

```
            animateButton.value = "停止";
        } else {
            paused = true;
            animateButton.value = "开始";
            gravity();
        }
    }

    animateButton1.onclick = function (e) {
        if (trace) {
            trace = false;
            animateButton1.value = "显示轨迹";
        } else {
            trace = true;
            animateButton1.value = "隐藏轨迹";
            //gravity();
        }
    }

    animateButton2.onclick = function (e){
        special = 1;
    }
    animateButton3.onclick = function (e){
        special = 2;
    }
    radiusRange.onchange = function(e) {
        G = 6.67 * Math.pow(10,4.5+radiusRange.value/100);
    }

    inputm1.onchange = function(e) {
        m[1] = inputm1.value;
    }

    inputm2.onchange = function(e) {
        m[2] = inputm2.value;
    }

    inputm3.onchange = function(e) {
        m[3] = inputm3.value;
    }

</script>

</body>

</html>

```