

# 基于行星历表的太阳系空间显示系统实现

任营营, 王解先

(同济大学 测绘与地理信息学院, 上海 200092)

**摘要:** 为了研究太阳系中太阳和八大行星在不同参考系下实际的运动状态和运行轨迹, 采用 DE430 行星历表计算出各天体太阳质心系中实际的状态向量, 介绍了不同参考系之间的转换关系, 并通过坐标转换进一步得到地球质心惯性系和地固系下天体的状态量; 利用 C# 和 SharpGL 工具设计开发出太阳系三维空间显示系统, 实现了不同参考系下各天体的坐标计算、动态显示、轨迹绘制和视点变化等功能, 并进一步对系统进行了优化设计, 软件运行结果表明, 各天体在不同参考系中的运行状态不同, 且各有各的运行规律。

**关键词:** SharpGL; DE 历表; 行星坐标; 坐标系转换

## Realization of Solar System Display System Based on DE

Ren Yingying, Wang Jiexian

(College of Surveying and GeoInformatics, Tongji University, Shanghai 200092, China)

**Abstract:** In order to study the actual state of motion and trajectory of the sun and its eight planets in different reference systems, this paper uses the Development Ephemerides 430 to calculate the actual state vectors in the celestial centroid system of each celestial body, and introduces the conversion methods among different reference systems then gets the coordinates of the planets in the Conventional Inertial System and Conventional Terrestrial System. Based on C# and SharpGL, this paper designs a solar system space display system to realize location calculation, the dynamic display, trajectory mapping and viewpoint change of the celestial bodies in different reference systems, and the system is further optimized. The result shows that the celestial bodies have different operating status in different reference systems and have their own law.

**Keywords:** SharpGL; development ephemerides (DE); planetary coordinates; coordinate transformation

## 0 引言

在太阳质心惯性系中, 八大行星围绕着太阳做近似圆周运动<sup>[1]</sup>。但是由于地球自身有公转、自转、极移、章动、岁差等运动现象, 居住在地球上的人类是无法真实地观察到这样的运动状态的。因此, 研究太阳系中各天体在地心参考系(包括地心惯性系和地心地固系)中实际的运动规律具有重要的意义。文献[2-7]只是从不同的层面模拟出个别天体在地心惯性系或地固系中的运动规律, 并未利用真实的天体坐标对其进行研究。本文正是基于 DE430 计算出天体的实际状态向量, 并利用 C# 和 SharpGL 对不同参考系中太阳系天体的实际运动状态和运行轨迹进行研究。

行星/月球(DE/LE)历表是美国喷气实验室(JPL)于20世纪60年代开始研发的星历表。该历表被广泛应用于星际探测、太空导航、天体测量、引力定律的检验、空间大地测量、地球物理等诸多相关学科领域<sup>[11]</sup>。截止目前, JPL网站已经发布了19份不同版本的DE/LE历表, 不同版本的历表在其内容和精度上都有一定的区别, 各版JPL行星历表的比较可参考JPL网站的介绍以及参考文献[8-9]。

C#是由C与C++语言派生出来的, 在吸收两者的优

点的同时, 弥补了两者在界面开发等方面的缺点。与C和C++相比, C#的语法简单易学, 配置与调试简单和开发周期短等优点, 如今被广泛应用于工程开发。SharpGL是进行三维显示的函数库, 是一种特殊的OpenGL库, 专门供C#调用<sup>[10]</sup>。与OpenGL相比, SharpGL在C#开发中更具吸引力, 它可以通过控件直接将绘图界面拖拽在窗体上, 并进一步设置其属性事件, 省去了一系列在OpenGL中的繁琐程序。

## 1 DE 星历表

本文中采用的是DE430历表, 该历表于2013年4月发布, 包含章动和天平动。起始历元为1549-12-21(儒略日: 2287184.5), 截止历元为2650-11-25(儒略日: 2688976.5), 涵盖了约1000年的时长。

为了高精度地描述各天体的空间位置, 反映其真实的运动状态, 需利用DE历表计算出天体的位置和速度。那么, 如何根据历表来获取行星的状态量呢? 通常可利用JPL网站上提供了Fortran版本的子程序进行计算, 其原理是通过切比雪夫插值多项式<sup>[8]</sup>插值得到。

Fortran版的PLEPH子程序可以计算出某一时刻目标天体相对于参照天体的状态量(6个参数包括位置和速度)。除Fortran版本外, 还有其他网站上提供的C版本程序, 大部分是由Fortran版本翻译而成。PLEPH(tdb, npl, nctr, pv)子程序包含4个参数, 其中tdb为儒略日, npl为行星

收稿日期: 2019-02-26; 修回日期: 2019-03-18。

作者简介: 任营营(1994-), 男, 河南郑州人, 硕士研究生, 主要从事大地测量数据处理与算法研究。

的编号, nctr 为中心行星编号 (如果计算章动和天平动, 中心天体编号为 0), 具体行星编号可参见表 1, pv 为天体的状态向量, 包含天体的三维坐标和速度, 单位分别是 au 和 au/day。调用时输入 4 个参数便可得到相应的天体运动状态向量。

表 1 天体(包含章动和天平动)编号

编号	天体(包含章动和天平动)	备注
1	Mercury	水星
2	Venus	金星
3	Earth	地球
4	Mars	火星
5	Jupiter	木星
6	Saturn	土星
7	Uranus	天王星
8	Neptune	海王星
9	Pluto	冥王星
10	Moon	月球
11	Sun	太阳
12	Solar-system barycenter	太阳系质心
13	Earth-moon barycenter	地月系质心
14	Nutations in longitude and obliquity	章动
15	Librations	天平动

## 2 坐标系统转换

本文除了研究太阳质心系中天体的运行规律外, 还重点研究天体在地心参考系中的运行规律, 因为人类正是处于地心系中观察各天体运行状态的。地心系包括地心惯性系和地心地固系, 两者的坐标系原点都是地球的质量中心, 不同的是各自的坐标轴指向, 因此, 实现地心惯性系与地固系转换的关键问题就是求出两者坐标轴指向的旋转矩阵。下面为地固系与地心惯性系的相互转换关系<sup>[12-13]</sup>。

考虑在某一历元时刻  $t$  下, 协议地球坐标系 (CTS) 到 J2000.0 协议地心惯性系 (CIS) 的旋转, 依次对协议地球系进行极移、地球自转、章动和岁差旋转变换可以转换至 J2000.0 协议惯性系。综上所述, 从地固系到 J2000.0 惯性系的坐标转换表达为:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{CIS} = PNSU \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{CTS} \quad (1)$$

由于旋转矩阵为正交矩阵, 正交矩阵的逆矩阵是自身的转置矩阵, 所以要想实现逆变, 即 J2000.0 惯性系到地固系的转换, 可表达为:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{CTS} = (PNSU)^T \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{CIS} \quad (2)$$

式中,  $U$  为极移矩阵, 它将协议历元平地球坐标系转换至瞬时极地球坐标系;  $S$  为地球的自转矩阵, 它将瞬时极地球坐标系转换至真天球坐标系;  $N$  为章动矩阵, 它将真天球坐标系转换至  $t$  历元时刻的测瞬平天球坐标系;  $P$  为岁差矩

阵, 它将  $t$  历元时刻的测瞬平天球坐标系转换到 J2000.0 地心惯性系。PNSU 具体的计算可参考文献 [12-13]。

## 3 系统设计与实现

### 3.1 功能设计

上文中提到, 居住在地球上的人类是无法真实的看到八大行星围绕着太阳做近似圆周运动情景的, 那是因为地球自身有公转、自转、极移、章动、岁差等运动现象, 如果在以地球为参考原点的坐标系中, 太阳和其他行星的运动状态是怎样的呢?

以地球质心为原点, 如果仅考虑地球公转, 可以采用地心惯性系, 如果进一步考虑地球自转、极移、章动、岁差等现象, 可以采用地固系。实际上居住在地球上的人类就是处在地固坐标系中观察各个行星的运动状态的。为观察太阳和八大行星的运行规律, 本文作者拟开发设计一套太阳系空间显示系统, 该系统的主要的设计功能有: 行星的坐标计算、天体的显示、轨道的绘制、天体运动的动态显示、不同参考系的转换以及观察角度的变换。图 1 为太阳系空间显示系统设计开发界面。



图 1 太阳系空间显示系统设计运行界面

### 3.2 结构设计

上文中已经给出了太阳系空间显示系统的具体设计功能, 那么如何实现这些功能呢? 本文将采用逆向思维法一步步进行分析, 给出详细的设计思路, 简略结构图见图 2。

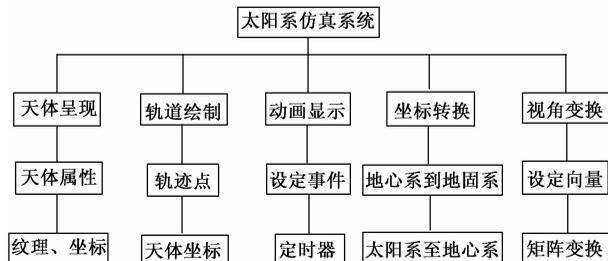


图 2 软件设计结构图

1) 天体的呈现: 要在屏幕上呈现出各天体的信息, 需要考虑各天体的大小、形状、表面、位置等信息。首先考虑到太阳和八大行星的体积比数量级太大, 如太阳和地球的体积比约为 30 万, 为了在屏幕上展示行星, 无法按照实际体积比设定天体的求半径大小, 因此可根据天体的体积比适当确定天体的大小; 虽然各天体的形状不是球体, 但是在屏幕坐标系下, 各个天体的体积被适当缩小, 完全可

以用球体代替各个天体的形状; 此外, 各个天体的表面颜色也有不同, 如太阳是橘黄色的、地球是蓝色的, 如何将真实的行星表面信息反应在模拟的天体上呢? 这里可以采用纹理贴图的方式将不同的行星图片粘贴在相应的球体上; 最后考虑如何描述行星的位置, 可以采用 DE 历表获取行星的状态向量。

2) 轨道的绘制: 轨道是由线形成的, 而轨迹线是由天体的运动轨迹点连接形成的, 因此只要获得不同历元下天体的轨迹点的三维坐标, 而且这些轨迹点足够密集, 通过点连成线, 即可形成天体的运行轨道。

3) 动画显示: 以动态的方式展示天体的运动, 首先考虑的是如何让天体自己动起来, 如果以一定的时间间隔触发一次事件, 这个事件包括重画计算机屏幕和重新设定新的天体坐标, 那么设置适当的时间间隔后触发该事件, 天体的位置不同, 便实现了天体的运动。C# 中的定时器(timer) 便有这样的功能, 它可以设定一定的时间间隔后不断触发事件<sup>[14]</sup>。

4) 坐标系转换: 坐标系转换主要涉及太阳质心系、地球质心系和地固系的转换。太阳质心系中的天体坐标可利用 DE 历表得到。地球质心系中的天体坐标可通过简单的坐标平移转换得到, 即利用天体在太阳质心系中的坐标减去中心天体在太阳质心系中坐标得到。地固系中的天体坐标需要考虑地球自转、极移、章动和岁差, 构造旋转矩阵可地心惯性系转换到地固系。

5) 视角变换: 不同的方向和角度下观察到天体的运动状态也是不同的, 如从 Z 轴上方沿下方看, 看到的是 XOY 平面的天体的运行状态。实现此功能首先要设定一组眼睛、目标和上方向的向量, 利用此向量通过矩阵变换可以得到不同视角下的天体运动状态。

6) 其他功能: 除上述主要功能外, 为提高软件的人机交互性能, 软件设计功能还包括天体演示过程暂停与播放功能、行星是否显示功能、轨迹是否显示功能以及帮助等其他功能。

### 3.4 系统实现

1) 定义绘制行星的方法, 可以设置行星的纹理、坐标、旋转角度。

```
public void drawplanet(string bmppath, float size, int number,
float rot, float x, float y, float z)
{
    OpenGL gl = openGLControl.OpenGL;
    gl.LoadIdentity(); //加载单位矩阵
    gl.Translate(x, y, z); //行星平移坐标
    gl.Rotate(rotation[number], 0.0f, 0.0f, 1.0f); //绕 Z 轴
    旋转
    texture.Create(gl, bmppath); //设置纹理
    gl.Sphere(ptrBall, size, 100, 100); //绘制星球
    rotation[number] += rot;
}
```

2) 绘制行星轨道, 每天画一个点, 将其连接起来。

Pleph(jd, target, center, R) 方法可以得到某一历元下目标行星相对于参照行星的相对状态量。

```
gl.Begin(OpenGL.GL_LINE_STRIP); //开始绘制线
for(int i=0; j < 365 * jplear; i++)
{
    jd = 2415020.5 + i; //改变儒略日
    ClassPLEPH.Pleph(jd, target, center, R); //获得天体的相对
    坐标
    gl.Vertex(R[0], R[1], R[2]); //绘制行星的位置点
}
gl.End(); //结束绘制
```

3) 设置定时器, 定时器每触发一次, 儒略日增加一日, 便重新绘制一次行星, 行星的坐标与上次不同, 实现了行星的运动。

```
private void timer1_Tick(object sender, EventArgs e)
{
    double jd;
    OpenGL gl = openGLControl.OpenGL;
    gl.Clear(OpenGL.GL_COLOR_BUFFER_BIT | OpenGL.GL_
    DEPTH_BUFFER_BIT);
    //清除颜色缓冲以及深度缓冲
    jd = 2415080.5 + ii; //儒略日变化
    ClassPLEPH.Pleph(jd, 1, 12, PO1); //新儒略日的行星相对
    坐标
    drawplanet(bmppath[0], 0.1f, 0, 0.103f, (float)PO1[0],
    (float)PO1[1], (float)PO1[2]);
    //绘制行星
    ii++;
}
```

4) 如要得到地心惯性系的天体坐标, 通过 Pleph(jd, target, center, R) 方法可将地球质心设置为中心参照体, 从而得到天体在地心惯性系中的坐标, 其原理是将各天体质心在太阳质心系中的状态量减去地球质心在太阳质心系中的状态量。把地心惯性系的坐标转换到地固系中, 需要进行极移、地球自转、章动和岁差旋转变换, 作者定义的 Frametrans(year, month, day, h, m, s, style, x0, y0, z0, x, y, z) 方法可以实现地固系和地心系的相互转换。

图 3 为天体在太阳质心系、地心系和地固系中的运动状态, 由图可知各天体在不同参考系下的运行状态和运行轨迹是完全不同的。在太阳质心惯性系中, 八大行星围绕太阳质心做近似圆周运动; 而在地心惯性系中, 由于地球的公转现象, 各天体(除太阳外)围绕地球质心运动的同时还伴有回旋现象发生, 且天体离地球的距离越近, 回旋现象越明显; 在地心地固系中, 正如人类在地球真实观察到的样子, 各天体围绕地心做螺旋曲线运动, 例如太阳以地球为参照中心, 人类可观察到太阳每天东升西落的现象。

5) 获取鼠标的屏幕坐标, 将其传递至函数 gl.LookAt(eye[0], eye[1], eye[2], center[0], center[1], center[2], up[0], up[1], up[2]) 中作为参数值, 从而实现视角变化, 函数中的三组参数分别代表眼睛的位置、瞄



图 3 不同参考系下天体的运动

准的位置和向上的方向。该函数实际上定义了一个视图矩阵，与当前矩阵相乘实现视点转换。

```
private void openGLControl_MouseDown(object sender, MouseEventArgs e)
{
    //拾取鼠标的屏幕坐标
    center[1] = e.X;
    center[2] = e.Y;
}
```

除了利用鼠标控制视点变换外，也可以通过键盘按键改变 gl.LookAt 的其他参数值实现场景的旋转和平移等功能。图 4 为不同视角下太阳系中天体的运动状态：

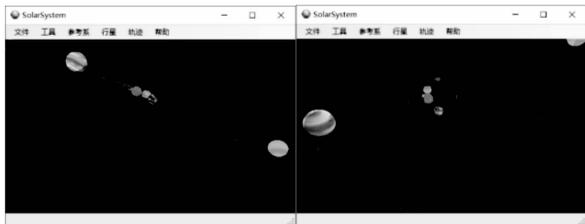


图 4 不同视点下天体的运动

### 3.5 系统优化

为了观察较长时间天体运动轨迹的变化情况，在绘制轨道的过程中，如设置绘制 20 年的轨道，一天绘制一个天体位置点，绘制八条轨道大约需要 58 000 个点，因此每次屏幕重画过程中，轨道的绘制占用了较长的时间，导致程序运行效率不高。同时还应考虑到天体的运动具有周期性，每天相同的时刻绘制天体的位置点可能会导致周期点位重合，造成一定的资源重复和浪费。

在此情况下，考虑使用切比雪夫插节点，摒弃等距节点。使用切比雪夫插值点可以保留更多更有效的轨道信息，更加真实地反映出行星的运动轨迹，避免了资源的重复浪费。在任意区间  $[\ ]$  上的  $n+1$  个切比雪夫节点可通过线性变换得到<sup>[15]</sup>：

$$x_i = \frac{1}{2} (a + b) + \frac{1}{2} (b - a) \cos \left[ \left( \frac{i}{n} \right) \pi \right] \quad (3)$$

## 4 结语

本文首先基于 DE430 得到太阳系各天体的空间状态量，然后利用 C# 和 SharpGL 混合开发出太阳系三维空间显示系统，动态显示太阳和八大行星在不同参考系中的运动状态和运行轨迹。显示结果表明，不同参考系下天体的运行状态和轨迹是不同的，在研究天体时若选择太阳质心系，各天体的围绕太阳做近似圆周运动；若选择地心惯性系，各天体围绕地球运动，除太阳外各天体还发生回旋现象，这是由于地球的公转产生的；若选择地固系，地球为参考原点且保持不动，各天体沿螺旋曲线绕地球进行运转。由此可见，选择不同的参考系对于研究目的有着极其重要的意义。

### 参考文献：

- [1] 程稼夫. 力学 [M]. 北京：科学出版社，2000.
- [2] 张三慧，王殊珠. 大学物理学—力学 [M]. 北京：清华大学出版社，1980.
- [3] 高炳坤. 三体问题的受力分析—兼谈失重 [J]. 大学物理，2004，23 (5)：7-8.
- [4] 高炳坤. 人造地球卫星与月亮在地心参考系中的受力分析 [J]. 大学物理，2004，23 (11)：3-5.
- [5] 杨立志，胡 斌. 以地球为参考系下太阳系各行星运动轨迹的探讨 [J]. 物理之友，2015，31 (10)：9-13.
- [6] 赵凯华，罗蔚茵. 新概念物理教程力学 [M]. 北京：高等教育出版社，1995.
- [7] 陶贤赞. 以地球视角看太阳运动 [J]. 高考，2017，(3)：159-161.
- [8] 雷伟伟，李 凯，张捍卫. DE 历表的结构、计算与比较 [J]. 飞行器测控报，2016，35 (5)：375-384.
- [9] 马高峰. JPL 行星/月球星历 [A]. 中国宇航学会深空探测技术专业委员会，2005：7.
- [10] 王晓松，徐 妍，田董伟，等. SharpGL 三维建模技术实现 [J]. 软件导刊，2017，16 (4)：205-208.
- [11] 邓雪梅，樊 敏，谢 懿. JPL 行星历表的比较及评估 [J]. 天文学报，2013，54 (6)：550-561.
- [12] 王解先. GPS 精密定轨与定位 [M]. 上海：同济大学出版社，1997.
- [13] 李克昭，杨 力，柴霖等. GNSS 定位原理 [M]. 北京：煤炭工业出版社，2014.
- [14] 宋文强，熊 壮. C# 程序设计 [M]. 北京：高等教育出版社，2010.
- [15] 同济大学计算数学教研室. 现代数值计算 [M]. 上海：人民邮电出版社，2014.