1 问题分析

此问题共可以分为 3 个阶段：

1. 利用数据解出未知参量；
2. 利用已知参量，画出相对于太阳的小行星轨道图；
3. 做出小行星运动动图，进一步分析小行星运动。

2 模型原理

问题指出，小行星的运动轨迹是一个椭圆，且已知以太阳为平面直角坐标系原点的五个观测数据。那么第一阶段此问题完全可以抽象为一个平面解析几何问题，即：已知二次曲线的五点坐标，求此二次曲线的解析方程并进行一系列的分析。下面开始逐步解决问题：

2.1 利用数据解出未知参量

二次曲线的方程为：

$$a_1x^2 + 2a_2xy + a_3y^2 + 2a_4x + 2a_5y + 1 = 0 \tag{1}$$

将数据直接代入曲线方程可以得到一个线性方程组：

$$AX = B \tag{2}$$

其中 A 为已代入数据的系数矩阵且 rank(A)=5，X 为未知数向量，B 为非齐次项。使用 matlab 解出这个线性方程组立即得到系数向量

$$[a_1\ \ a_2\ \ \ a_3\ \ \ a_4\ \ \ a_5] = [-0.3378\ \ 0.1892\ \ \ -0.3818\ \ 0.4609\ \ 0.4104]$$

2.2 画出相对于太阳的小行星轨道图

使用 matlab 画出以此二次曲线的小行星相对图，利用 matlab 函数 fimplicit 即可画出隐函数（方程）图像，再经过一系列标点处理后得到图 1

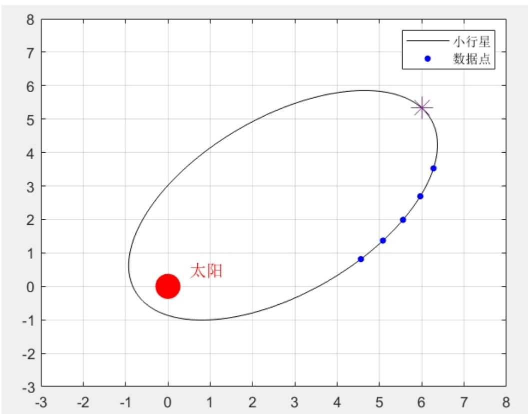

Figure 1: 小行星轨道相对图

可以发现这个小行星运动轨道是一个椭圆，这符合开普勒第一定律。

2.3 做出小行星运动动图，进一步分析小行星运动

基本思路是使用函数 set 与 for 循环，做出小行星运动动图。注意到问题：set 函数仅支持一维向量表示的坐标点，这意味着要计算出小行星绕轨道运动的基本数据，此时若再使用二元变量的坐标，则会出现映射中“多对一”的情况，将无法满足 set 函数使用的条件。这里选择将原二次曲线方程通过解析计算参数化，以达到用单变量表示运动数据的目的。

画出原二次曲线的图像

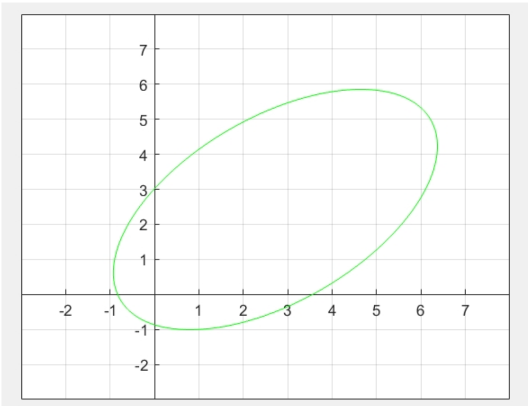

Figure 2: 轨道二次曲线图像

很明显此曲线存在交叉项 xy，接下来的任务是将曲线旋转至长轴平行于 x 或 y 轴，以消除交叉项；然后将曲线平移至原点处，最后取参数变换

$$\begin{cases} x = a\cos t \\ y = b\sin t \end{cases} \qquad t \in [\,0,\ 2\pi]$$

将双变量转化为单变量。最后再取逆变换，得到原二次方程关于 t 参数的参数方程

2.3.1 二次曲线的变换

二次曲线的平移变换分为平移变换和旋转变换

首先引入旋转变换

$$\begin{cases} x = x'\cos\alpha - y'\sin\alpha \\ y = x'\cos\alpha + y'\cos\alpha \end{cases}$$

其中 α 是将图像顺时针旋转的角度，使用这个公式可以将二次曲线的方程转化成

$$a_1'x'^2 + 2a_2'x'y' + a_3'y'^2 + 2a_4'x' + 2a_5'y' + 1 = 0 \tag{3}$$

转化后方程的常数项分别为

$$\begin{cases} a_1' = a_1\cos^2\alpha + 2a_2\sin\alpha\cos\alpha + a_3\sin^2\alpha \\ a_2' = (a_3 - a_1)\sin\alpha\cos\alpha + a_3(\cos^2\alpha - \sin^2\alpha) \\ a_3' = a_1\sin^2\alpha - 2a_2\sin\alpha\cos\alpha + a_3\cos^2\alpha \\ a_4' = a_4\cos\alpha + a_5\sin\alpha \\ a_5' = -a_4\sin\alpha + a_5\cos\alpha \end{cases}$$

由于首要目的是消去方程的交叉项，以此我们令 $a_3' = 0$，即

$$a_3' = a_1 \sin^2\alpha - 2a_2 \sin\alpha\cos\alpha + a_3\cos^2\alpha = 0$$

经过一系列的推导不难得到下面关于 $\tan\alpha$ 的一元二次方程

$$a_2\tan^2\alpha + (a_1 - a_3)\tan\alpha - a_2 = 0 \tag{4}$$

由此得到旋转角度的正切值 $\tan\alpha$ 由于是顺时针转动，故 (4) 的根取正值。

接下来再根据三角变换公式

$$\sin\alpha = \frac{\tan\alpha}{\sqrt{1+\tan^2\alpha}}, \quad \cos\alpha = \frac{1}{\sqrt{1+\tan^2\alpha}}$$

得到旋转变换中所需要的三角函数值。

现在我们已经使得交叉项为零，且由于二次曲线的长轴是与 x 轴对齐的，所以方程 () 4 中关于 y 的项一定为 0。那么，我们就得到化简后的（4）：

$$a_1'x'^2 + a_3'y'^2 + 2a_4'x' + 1 = 0 \tag{5}$$

进一步考虑平移。将（3）进行配方得到

$$a_1'(x' + \frac{a_4'}{a_1'})^2 + a_3'y'^2 = \frac{a_4'^2}{a_1'} - 1$$

取平移变换

$$\begin{cases} x'' = x' + \dfrac{a_4'}{a_1'} \\ y'' = y' \end{cases}$$

得到只关于二次项的曲线方程

$$a_1'x''^2 + a_3'y''^2 = \frac{a_4'^2}{a_1'} - 1 \tag{6}$$

接下来只需要简单的归一化就可以将方程化为标准方程

$$a_1''x''^2 + a_3''y''^2 = 1 \tag{7}$$

其中 $a_1'' = \dfrac{a_1'}{\dfrac{a_4'^2}{a_1'} - 1}, \quad a_2'' = \dfrac{a_2'}{\dfrac{a_4'^2}{a_1'} - 1}$ 现在再使用参数变换即得

$$\begin{cases} x' = \sqrt{\dfrac{\dfrac{a_4'^2}{a_1'} - 1}{a_1'}}\cos t - \dfrac{a_4'}{a_1'} \\ y' = \sqrt{\dfrac{\dfrac{a_4'^2}{a_3'} - 1}{a_1'}}\sin t \end{cases} \quad t \in [\,0,\ 2\pi] \tag{8}$$

现在，再使用旋转公式就可以得到原二次曲线方程的参数形式，也就是

$$F: \begin{cases} x = x'\cos\alpha - y'\sin\alpha \\ y = x'\cos\alpha + y'\cos\alpha \end{cases} \tag{9}$$

式（9）中的三角项由式（4）决定，变换项由式（8）决定。至此，我们已经把问题所需要的理论依据阐述完毕，现在开始具体的算法实现。

结合第一问中得到的参数向量 X，将其中的第 1，2，3 项代入式子（4）中得到旋转角度的正切值

$$\tan\alpha = 0.8905$$

再根据三角变换得，旋转所需要的三角函数值

$$\sin\alpha = 0.6651, \quad \cos\alpha = 0.7468$$

现在根据旋转公式得到变换后的常数项：

$$X' = [\,-0.1694\ \ 0\ \ -0.5502\ \ 0.6171\ \ 0]$$

接下来根据 2.3.1 的固定流程不难得到关于参数 t 的参数方程

$$F : \begin{cases} x = x'\cos\alpha - y'\sin\alpha \\ y = x'\cos\alpha + y'\cos\alpha \end{cases}$$

这就是 set 函数所需要的两个向量，另外，为了检验是否正确，使用 plot 函数以分别一上面两个向量组为横坐标和纵坐标，得到图像

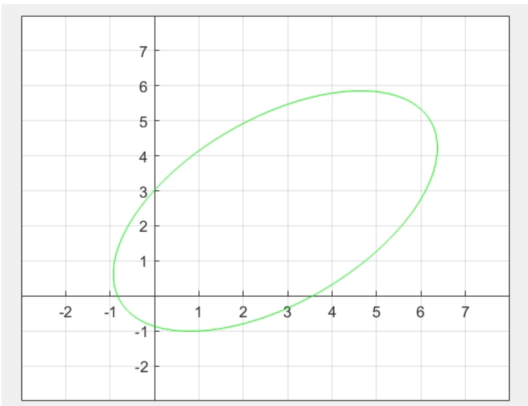

Figure 3: 参数曲线图像

发现和 Figure 2 完全一致，这证明上述思路正确，同时也再次证明了二次曲线确实可以用单参数表示。

现在已经完全满足了做出动图的条件，使用 for 循环和 drawdnow 命令即可画出动态图（见文件夹附件 point_animation.gif）

3 文件代码

asteroid.m

```
function A=asteroid(X,Y)
f=@(x,y)[x^2,2*x*y,y^2,2*x,2*y];
C=zeros(5,5);
for i=1:5
    C(i,:)=f(X(i),Y(i));
end
B=-ones(1,5);
A=B/C.';
end
```

problem.m

```
clc
X=[4.5596 5.0816 5.5546 5.9636 6.2756];
Y=[0.8145 1.3685 1.9895 2.6925 3.5265];
Initial=[0 0];
A=asteroid(X,Y);

f=@(x,y)A(1)*x^2+2*A(2)*x*y+A(3)*y^2+2*A(4)*x+A(5)*2*y+1;

b=A(2);
a=A(1);
c=A(3);
co=[b,a-c,-b];
r=roots(co);
t=r(2);
sin1= t/sqrt(1 +t^2);
cos1= 1/sqrt(1 +t^2);
A2=[A(1)*cos1^2+2*A(2)*sin1*cos1+A(3)*sin1^2
    0
    A(1)*sin1^2-2*A(2)*sin1*cos1+A(3)*cos1^2
    A(4)*cos1+A(5)*sin1
    -A(4)*sin1+A(5)*cos1];

t=0:0.01:2*pi;
x1=cos(t)*sqrt((A2(4)^2/A2(1)+A2(5)^2/A2(3)-1)/A2(1))-A2(4)/A2(1);
y1=sin(t)*sqrt((A2(4)^2/A2(1)+A2(5)^2/A2(3)-1)/A2(3))-A2(5)/A2(3);

x=cos1*x1-y1*sin1;
y=sin1*x1+y1*cos1;

figure;
```

```
fimplicit(f, [-3, 8, -3, 8], 'black-',LineWidth=0.2);
grid on
hold on
plot(X,Y,'.', 'MarkerSize', 15,Color="blue")
p=plot(0,0,'.', 'MarkerSize', 60,Color="red");
text(0.5,0.5,'太阳','FontSize',12,'Color',"red")
h= plot(x(1), y(1), '*', 'MarkerSize', 15, 'MarkerFaceColor', '#77AC30');
legend("小行星",'数据点')

filename = 'point_animation.gif';
for k = 1:length(t)
    set(h, 'XData', x(k), 'YData', y(k));
    drawnow;
    pause(0.0001);

    %frame = getframe(gcf);
    %im = frame2im(frame);
    %[imind, cm] = rgb2ind(im, 256);
    %if k == 1
    %     imwrite(imind, cm, filename, 'gif',
    %'Loopcount', inf, 'DelayTime', 0.1);
    %else
    %     imwrite(imind, cm, filename, 'gif',
    %'WriteMode', 'append', 'DelayTime', 0.1);
    %end
end
```