二体模型的建立：

由于在主减速段嫦娥三号运动时间较短，可以忽略月球的自转，不妨假设从近月点到预着陆点在一个平面内，建立如图极坐标系，其中O为月球球心，为飞船当前位置与y轴的夹角，F为发动机的脉冲推力，为飞船转动角速度，r为飞船距离月球球心的距离， 为推力与水平速度反方向的夹角，v为飞船竖直方向分速度，为制动发动机比冲。

可推得如下动力学模型：

已知嫦娥三号在近月点的初始状态为：

假设发动机推力与速度方向始终呈夹角不变，求解该模型即在满足系统终端条件下，使主减速过程中消耗燃料尽量小，即到达终端时剩余m尽量大。由于精确遍历所有角度可能复杂度较高，故考虑采用免疫优化算法求解，其算法步骤如下：

1.随机生成初始抗体群（解空间），利用全局变量temp\_fai来传递抗体求解微分方程组并验证其可行性从而得到满足条件（末状态在允许误差范围内）的初始抗体（可行解）。

2.按亲和度（解的误差和末状态下剩余质量）排序。

3.选取亲和度前一半的抗体进行免疫处理。先利用repmat函数仿真抗体克隆，再对克隆后的抗体群进行变异处理（加上一个在区间[-0.01, 0.01]均匀分布的随机数，注意保留父本即从第二个抗体开始变异以防止出现退化现象）。

4.进行克隆抑制，筛选出每一个克隆抗体群中的最优抗体加入记忆细胞库中。

5.随机生成满足条件的抗体加入记忆细胞库以取代第3步中筛去的亲和度低的另一半抗体，并对记忆细胞库中的抗体按亲和度进行排序。

6.当达到最大迭代次数时终止循环，此时记忆细胞库中排名第一的便是近似全局最优解。

通过matlab编程求解得，合末速度为57.4038m/s，其中水平分速度为53.8855m/s，竖直分速度为19.7874m/s指向月球球心，最终用时为409s，剩余质量，燃料消耗为1040.8163kg，末转动角度。

假设嫦娥三号轨道与经线面重合并自南向北主减速，则可得近月点坐标为19.51W，31.47N，由于月球对称性，易得远月点坐标为160.49E，31.47S。



matlab代码：

fun2\_solve:

clear, clc;

t\_sum = 450; % 最大总用时

vx0 = 1692.53; % 水平初速度

v0 = 0; % 竖直初速度

r0 = 1749373; % 初始高度

omega = vx0 / r0; % 初始角速度

m0 = 2400; % 初始质量

theta = 0; % 初始转动角度

global temp\_fai; % 定义全局变量‘临时fai值’

Antibody\_Num = 20; % 定义抗体数

Iter\_Max = 10; % 定义最大迭代数

H\_eps = inf;

V\_eps = 2;

flag = 0;

Clone\_Num = 10; % 克隆抗体数

Iter\_Num = 0; % 当前克隆次数

fai\_Antibody = zeros(Antibody\_Num, 4);

%%%%%随机生成满足条件的初始抗体群%%%%%

for i = 1 : Antibody\_Num

while true

temp\_fai = rand(1, 'double')\*2-1;

[t, temp\_Mat] = ode45(@fun2, [0: 1 :t\_sum], [r0 v0 theta omega m0]);

temp\_H = temp\_Mat(:, 1) - 1737013;

temp\_V = sqrt(temp\_Mat(:, 2).^2 + (temp\_Mat(:, 1).\*temp\_Mat(:, 4)).^2);

if find(abs(temp\_V + -57) < V\_eps)

fai\_Antibody(i, 1) = temp\_fai;

fai\_Antibody(i, 2) = temp\_Mat(find(abs(temp\_V + -57) < V\_eps, 1, 'first'), 5);

fai\_Antibody(i, 3) = abs(temp\_V(find(abs(temp\_V + -57) < V\_eps, 1, 'first')) + -57);

break;

end

end

end

%%%%%进行免疫处理%%%%%

fai\_Antibody\_sorted = sortrows(fai\_Antibody, [-2, 3]); % 排序

while Iter\_Num < Iter\_Max

% 选取亲和度前一半的抗体进行免疫处理

for i = 1 : Antibody\_Num/2

% 进行抗体克隆

Antibody\_Clone = repmat(fai\_Antibody\_sorted(i, 1), Clone\_Num, 1);

% 进行抗体变异

for j = 2 : Clone\_Num

Antibody\_Clone(j) = Antibody\_Clone(j) + randn(1)\*0.01;

end

for j = 1 : Clone\_Num

temp\_fai = Antibody\_Clone(j);

[t, temp\_Mat] = ode45(@fun2, [0: 1 :t\_sum], [r0 v0 theta omega m0]);

temp\_H = temp\_Mat(:, 1) - 1737013;

temp\_V = sqrt(temp\_Mat(:, 2).^2 + (temp\_Mat(:, 1).\*temp\_Mat(:, 4)).^2);

if find(abs(temp\_V + -57) < V\_eps)

Antibody\_Clone(i, 2) = temp\_Mat(find(abs(temp\_V + -57) < V\_eps, 1, 'first'), 5);

Antibody\_Clone(i, 3) = abs(temp\_V(find(abs(temp\_V + -57) < V\_eps, 1, 'first')) + -57);

Antibody\_Clone(i, 4) = 1;

else

Antibody\_Clone(i, 2) = 0;

Antibody\_Clone(i, 3) = inf;

Antibody\_Clone(i, 4) = 0;

end

end

Antibody\_Clone\_sorted = sortrows(Antibody\_Clone, [-4, -2, 3]);

temp\_Antibody = zeros(Antibody\_Num/2, 4);

temp\_Antibody(i, :) = Antibody\_Clone\_sorted(1, :);

end

fai\_Antibody(1 : Antibody\_Num/2, :) = temp\_Antibody(1 : Antibody\_Num/2, :);

for i = Antibody\_Num/2 + 1 : Antibody\_Num

while true

temp\_fai = rand(1, 'double')\*2-1;

[t, temp\_Mat] = ode45(@fun2, [0: 1 :t\_sum], [r0 v0 theta omega m0]);

temp\_H = temp\_Mat(:, 1) - 1737013;

temp\_V = sqrt(temp\_Mat(:, 2).^2 + (temp\_Mat(:, 1).\*temp\_Mat(:, 4)).^2);

if find(abs(temp\_V + -57) < V\_eps)

fai\_Antibody(i, 1) = temp\_fai;

fai\_Antibody(i, 2) = temp\_Mat(find(abs(temp\_V + -57) < V\_eps, 1, 'first'), 5);

fai\_Antibody(i, 3) = abs(temp\_V(find(abs(temp\_V + -57) < V\_eps, 1, 'first')) + -57);

break;

end

end

end

fai\_Antibody\_sorted = sortrows(fai\_Antibody, [-4, -2, 3]); % 排序

Iter\_Num = Iter\_Num + 1;

end

Best\_fai = fai\_Antibody\_sorted(1, 1);

[t, Mat] = ode45(@fun2, [0: 1 :t\_sum], [r0 v0 theta omega m0]);

%v\_sum = zeros(t\_sum, 3600);

% for sub = 1 : 3600

% fai = pi/1800\*sub;

% [t, A] = ode45(@fun2, [0: 1 :t\_sum], [r0 v0 theta omega m0]);

%

% for i = 1 : t\_sum + 1

% v\_sum(i, sub) = sqrt(A(i, 2)^2 + (A(i, 4)\*A(i, 1))^2);

% H(i, sub) = A(i, 1) - 1737013;

% theta\_sum(i, sub) = A(i, 3);

% m\_sum(i, sub) = A(i, 5);

% end

% end

%

% plot(t, H(:,1))

%

% min(min(v\_sum))

fun2.m:

function f = fun2(t, x)

G = 6.67259e-11;

M = 7.3477e+22;

F = 7500;

ve = 2940;

global temp\_fai;

f = zeros(5, 1);

f(1) = x(2);

f(2) = F/x(5)\*sin(temp\_fai)-G\*M/(x(1)^2)+x(4)^2\*x(1);

f(3) = x(4);

f(4) = -1/x(1)\*(F/x(5)\*cos(temp\_fai)+2\*x(2)\*x(4));

f(5) = -F/ve;

end