导弹跟踪问题

5122409009 王力功
5121209117 徐小博

一． 实际问题概述

某军的导弹基地发现正北方向 120km 处海面上有敌舰一艘以 90km/h 的速度向正东方向行驶。该基地立刻发射导弹追踪，导弹速度 450km/h，自动导航系统使导弹在任一时刻都能对准敌舰。试问导弹在何时何处击中敌舰？

二．数学模型

设导弹在原点 O(0,0)，x 轴指向正东，y 轴指向正北。t=0 时，导弹位于 O，敌舰位于 A(0,H)，其中 H=120(km)。设导弹在 t 时刻的位置为 P(x(t), y(t))，由题意得

$$\left(\frac{dx}{dt}\right)^2+\left(\frac{dy}{dt}\right)^2=v_w^2$$

其中$v_w=450(km/h)$。在 t 时刻，敌舰位置 M(v_et,H)，其中$v_e=90(km/h)$。

由题意得，导弹切线方向指向敌舰，所以直线 PM 方向就是导弹轨迹上 P 的切线方向，于是

$$\begin{cases}\dfrac{dy}{dx}=\dfrac{H-y}{v_e t-x}\\ x(0)=0, y(0)=0\end{cases}$$

得出一个关于时间变量 t 的一阶微分方程组的初值问题。我们消去变量 t，

$$\frac{\mathrm{d}x}{\mathrm{d}y}(H-y)=v_e t-x$$

两边对 t 求导得

$$\frac{\mathrm{d}^2x}{\mathrm{d}y^2}\frac{\mathrm{d}y}{\mathrm{d}t}(H-y)+\frac{\mathrm{d}x}{\mathrm{d}y}\left(-\frac{\mathrm{d}y}{\mathrm{d}t}\right)=v_e-\frac{\mathrm{d}x}{\mathrm{d}t}$$

即有

$$\frac{\mathrm{d}^2x}{\mathrm{d}y^2}\frac{\mathrm{d}y}{\mathrm{d}t}(H-y)=v_e$$

把前面方程写为$\dfrac{\mathrm{d}y}{\mathrm{d}t}=\dfrac{v_w}{\sqrt{\left(\dfrac{\mathrm{d}x}{\mathrm{d}y}\right)^2+1}}$代入上式，就得到轨迹方程. 这是一个二阶非线性微分方程，加上初值条件，则初值问题

$$\begin{cases} \frac{d^2x}{dy^2}\frac{(H-y)}{\sqrt{\left(\frac{dx}{dy}\right)^2+1}}=\frac{v_e}{v_w} \\ x\Big|_{y=0}=0 \\ \frac{dx}{dy}\Big|_{y=0}=0 \end{cases}$$

即为导弹轨迹的数学模型。

a) 解析法

令$p=\frac{dx}{dy}$，记$\lambda=\frac{v_e}{v_w}$，则可以化为一阶可分离变量方程 $\frac{dp}{dy}=\frac{\lambda\sqrt{p^2+1}}{H-y}$

参考教材上的推导过程，最终可以求得

$$L=\frac{\lambda H}{1-\lambda^2}=\frac{Hv_wv_e}{{v_w}^2-{v_e}^2}$$

把数据$H=120(km/h),v_w=450(km/h),v_e=90(km/h)$代入，得

$$L\approx 25(km),\ T\approx 0.2778(h)$$

b) Euler 法

设导弹到达(x_k,y_k)处的时刻为t_k，那么得到计算的迭代格式为

$$\begin{cases} x_{k+1}=x_k+hp_k \\ p_{k+1}=p_k+h\frac{\lambda\sqrt{1+{p_k}^2}}{H-y_k} \\ x_0=0,p_0=0 \end{cases}$$

$$\text{其中}h=\frac{H}{n},L\approx x_n,T\approx\frac{x_n}{v_e}$$

使用 MATLAB，编制文件：

```
function m4_1(n)
H=120;
h=H/n;
lambda=90/450;
x(1)=0;p(1)=0;
y=0:h:H;
for i=0:n-1
    x(i+2)=x(i+1)+h*p(i+1);
    p(i+2)=p(i+1)+h*(lambda*sqrt(1+p(i+1)^2)/(H-y(i+1)));
end

x
p
L=x(n+1)
T=x(n+1)/90
```

尝试步长为 4，得到 n=4 时的计算结果为

x =　　0　　0　　1.5000　　5.0025　　11.5254

p =　　0　　0.0500　　0.1167　　0.2174　　0.4221

L =　　11.5254

T =　　0.1281

下面，我们针对不同的 n 值计算结果，MATLAB 代码如下：

```
function m4_2(N)
k=1;
H=120;
for n=N
    h=H/n;
    lambda=90/450;
    x0=0;p0=0;
    for i=0:n-1
        x1=x0+h*p0;
        p1=p0+h*(lambda*sqrt(1+p0^2)/(H-i*h));
        x0=x1;
        p0=p1;
    end
    L(k)=x1;
    I(k)=x1/90;
    k=k+1;
end
N
L
I
```

先观察步长 n=[4,8,12,24,48,96,120,240]时的结果，如表 1 所示。

n	4	8	12	24	48	96	120	240
L	11.525	15.964	17.973	20.551	22.249	23.329	23.580	24.151
T	0.1281	0.1773	0.1997	0.2283	0.2472	0.2592	0.2620	0.2683

表 1

将步长继续增加，观察步长 n=[480,600,960,1560,1920,2000]时的结果，如表 2 所示。

n	480	600	960	1560	1920	2000
L	24.4965	24.5751	24.7034	24.7958	24.8261	24.8315
T	0.2722	0.2731	0.2745	0.2755	0.2758	0.2759

表 2

【任务 1.1】从表 1 及表 2 的比较中可见 n 越大，也即 h 越小时，结果越精确。

接下来，我们对初值问题进行数值处理，将问题变为

$$\begin{cases} \dfrac{dx}{dt} = \dfrac{v_w}{\sqrt{1+\left(\dfrac{H-y}{v_e t-x}\right)^2}} \\ \dfrac{dy}{dt} = \dfrac{v_w}{\sqrt{1+\left(\dfrac{v_e t-x}{H-y}\right)^2}} \\ x(0)=0,\ y(0)=0 \end{cases}$$

取时间步长 $\Delta t=\tau$，对应 $t_k=k\tau$ 时导弹轨迹上点的坐标为 (x_k,y_k)，则 Euler 格式为

$$\begin{cases} x_{k+1}=x_k+\dfrac{v_w\tau}{\sqrt{1+\left(\dfrac{H-y_k}{v_e t_k-x_k}\right)^2}} & (4.21) \\ y_{k+1}=y_k+\dfrac{v_w\tau}{\sqrt{1+\left(\dfrac{v_e t_k-x_k}{H-y_k}\right)^2}} & (4.22) \\ x_0=0,\ y_0=0 & (4.23) \end{cases}$$

当计算到 $y_k<H, y_{k+1}\geqslant H$ 即停止，于是，

$$L\approx x_{k+1}(\text{或 } L\approx x_k),\quad T\approx\frac{L}{v_e}$$

使用 MATLAB，代码如下：

```
function m4_3(t)
H=120;Ve=90;Vw=450;
x(1)=0;y(1)=0;I(1)=0;
for i=1:10e6
    M=(Ve*I(i)-x(i))/(H-y(i));
    x(i+1)=x(i)+Vw*t/sqrt(1+1/M.^2);
    y(i+1)=y(i)+Vw*t/sqrt(1+M.^2);
    I(i+1)=i*t;
    if y(i+1)>=H
        break
    end
end
I
x
y
L=x(i+1)
I=x(i+1)/Ve
```

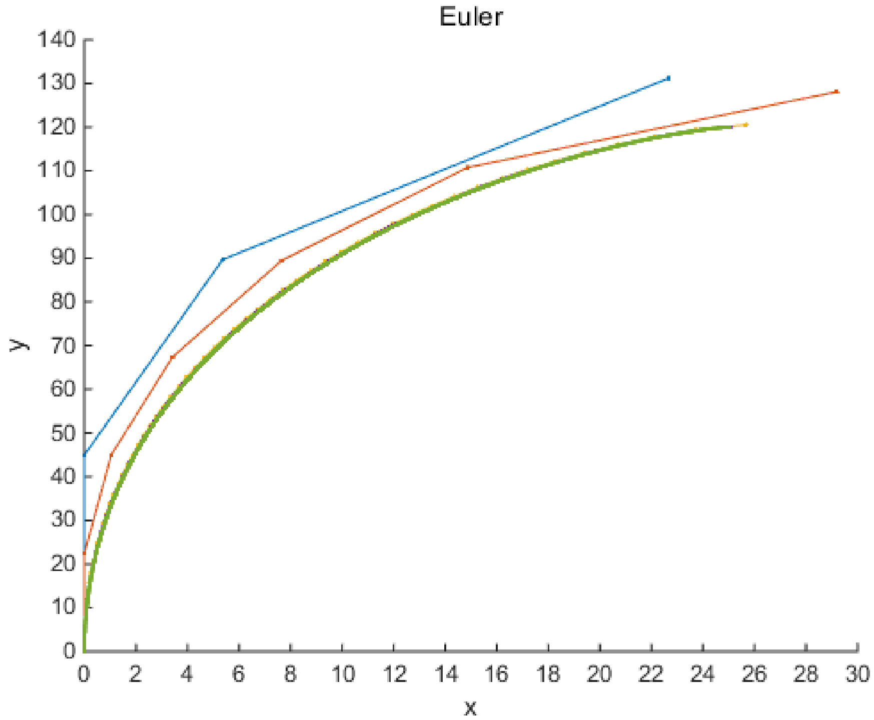
表 3 是对应不同的步长τ，用 Euler 法所得相应的步长推进次数 n 和计算结果。

τ	0.1	0.01	0.001	0.0001
n	3	29	279	2779
L	22.67495	25.2645	25.0493	25.0058
T	0.25194	0.280722	0.2783	0.2778

表 3

表 3 所显示的结果，和之前的结论保持一致：步长越小，计算结果更精确。

我们使用 MATLAB 作图，显示了不同步长的结果。当步长足够小时，精确程度在图上不够显著，但对于制导这样的精细操作来说，多精确 1 位小数都至关重要。



c) 改进的 Euler 法（预报 — 矫正法）

以一维情况为例，对问题

$$\frac{dx}{dt}=f(x,t),\ x(t_0)=x_0$$

Euler 迭代格式是

$$x_{k+1}=x_k+hf(x_k,t_k)$$

其中 $h=\Delta t, t_k=kh$. 而改进的 Euler 迭代格式则是

$$x_{k+1}=x_k+\frac{h}{2}[f(x_k,t_k)+f(x_{k+1}^*,t_{k+1})]$$

$$=\frac{1}{2}[x_{k+1}^*+x_k+hf(x_{k+1}^*,t_{k+1})]$$

其中

$$x_{k+1}^*=x_k+hf(x_k,t_k)$$

由积分表达式

$$x_{k+1}-x_k=\int_{t_k}^{t_k+h}f(x(\tau),\tau)d\tau$$

的几何意义来看，Euler 法是用矩形来代替曲边梯形，而改进的 Euler 法则是用梯形来代替曲边梯形。

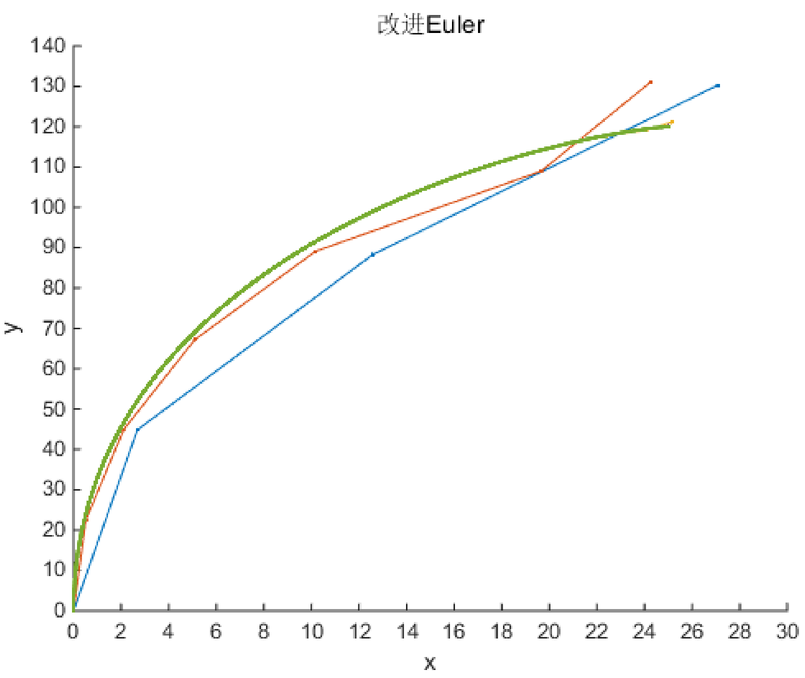
我们写出相应的改进 Euler 迭代格式

$$
\begin{cases}
x_{k+1}=\dfrac{1}{2}\left[x_{k+1}^{*}+x_k+\dfrac{v_w\tau}{\sqrt{1+\left(\dfrac{H-y_{k+1}^{*}}{v_e t_{k+1}-x_{k+1}^{*}}\right)^2}}\right] \\
y_{k+1}=\dfrac{1}{2}\left[y_{k+1}^{*}+y_k+\dfrac{v_w\tau}{\sqrt{1+\left(\dfrac{v_e t_{k+1}-x_{k+1}^{*}}{H-y_{k+1}^{*}}\right)^2}}\right] \\
x_{k+1}^{*}=x_k+\dfrac{v_w\tau}{\sqrt{1+\left(\dfrac{H-y_k}{v_e t_k-x_k}\right)^2}} \\
y_{k+1}^{*}=y_k+\dfrac{v_w\tau}{\sqrt{1+\left(\dfrac{v_e t_k-x_k}{H-y_k}\right)^2}} \\
x_0=0,\quad y_0=0
\end{cases}
$$

表 4 是对应不同的步长τ，用改进的 Euler 法所得相应的步长推进次数 n 和计算结果。

τ	0.1	0.01	0.001	0.0001
n	3	29	279	2779
L	27.0724	23.9104	24.9811	25.0050
T	0.3008	0.2657	0.2776	0.2778

表 4



【任务 1.2】从表 4 中我们可以看到，在步长相等的情况下，改进 Euler 法的计算结果总是比 Euler 法要更加精确，说明此改进确实有效。但是当步长足够小，即达到 0.0001 级别的时候，Euler 法（0.277842314589063）和改进 Euler 法（0.277833129115080）的差别已经不显著了，仅在五位小数以后才有体现，几乎可忽略不计。

【任务 2】我们在之前的计算过程中，是计算到$y_k < H, y_{k+1} \geq H$即停止，然后取$L \approx x_{k+1}$，这样做法可能会有不小的误差，有时甚至会出现整体步长改小而结果却未必能改进的情况。由于 Euler 法或者改进的 Euler 法的计算格式中每一步的值的取得仅仅依赖上一步的值，因此在计算过程中改变步长是可行的，即当计算到$y_k < H, y_{k+1} \geq H$，而y_{k+1}远大于 H 时，可缩小步长（例如原来的十分之一）以x_k, y_k作为新起点继续进行迭代。我们使用这种变长方法来改进任务 1 中得到的结果。

MATLAB 代码如下所示：

```
function [x,y,L,I]=m4_3_3(t)
H=120;Ve=90;Vw=450;
x(1)=0;y(1)=0;I(1)=0;
for i=1:10e6
    M=(Ve*I(i)-x(i))/(H-y(i));
    x(i+1)=x(i)+Vw*t/sqrt(1+1/M.^2);
    y(i+1)=y(i)+Vw*t/sqrt(1+M.^2);
    I(i+1)=I(i)+t;
    if y(i+1)>=H
        x(i+1)=x(i);
        y(i+1)=y(i);
        I(i+1)=I(i);
        t=t/10;
    end
    if t<1e-8
        break
    end
end
L=x(i+1);
I=x(i+1)/Ve;
```

注意到 t 会在特判下缩小步长，我们依然使用不同的初始步长进行测试，所得相应的步长推进次数 n 和计算结果如表 5 所示。

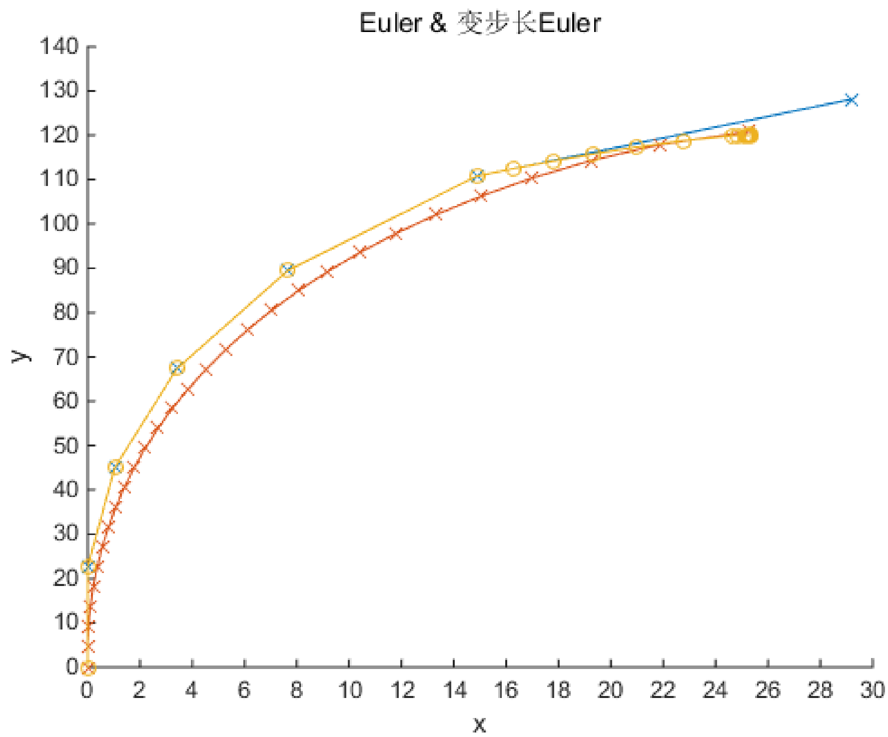
步长	Euler 法		改进 Euler 法		改变步长法	
	L	T	L	T	L	T
0.05	29.1948	0.3244	24.24089	0.26934	25.33684	0.28152
0.01	25.2645	0.2807	23.9104	0.26567	25.07523	0.27861
0.001	25.0493	0.2783	24.9811	0.27757	25.0088	0.27788

表 5

从表 5 中我们可以进行横向、纵向对比。在同一步长的情况下，改进 Euler 法和改变步长法都比 Euler 法更精确，而且在步长较大（0.05）的时候，改进 Euler 法和改变步长法

已经达到了一个较为精确的状态，但是 Euler 法还在差距很远的地方。

下面我们用 MATLAB 作图来着重对比改变步长 Euler 方法和标准 Euler 方法。



图中大叉的曲线代表了标准 Euler 方法，两个曲线的步长分别为 0.05 和 0.01，计算结果如下

t = 0.0500，L = 29.1948，T = 0.3244

t = 0.0100，L = 25.2645，T = 0.2807

图中圆圈的曲线代表改变步长 Euler 方法，初始步长为 0.05，每当$y_{k+1} \geq H$，则将步长减小十倍，直至步长小于 10^{-8} 终止迭代，计算结果如下：

t = 0.0500，L = 25.3368，T = 0.2815

注意到改变步长法在初始步长为 0.05 时的迭代效果和标准 Euler 方法步长在 0.01（前者的五分之一）时的迭代效果几乎相当，由此可见，改变步长的 Euler 方法相较于标准 Euler 方法优化效果显著。

d） 仿真方法

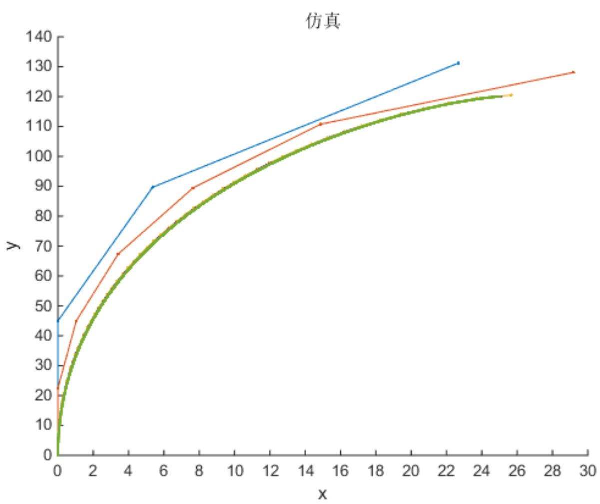
仿真方法同教材，此处略。

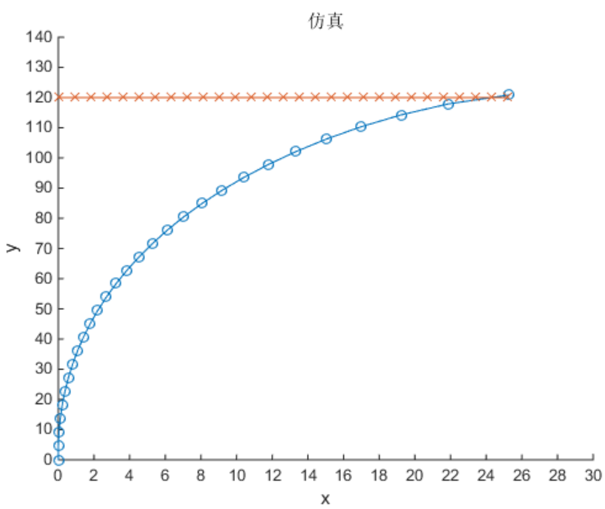
根据相应的 MATLAB 代码计算，输入相应的步长，发现得到的结果与 Euler 迭代的结果完全一致。这两种迭代格式实质上是相通的，但在仿真方法中，我们没有使用微分方程，而是用离散形式进行模拟。

MATLAB 代码如下：

```matlab
function [x,y,L,T]=m4_4(t)
H=120;Ve=90;Vw=450;
x(1)=0;y(1)=0;
for i=1:10e6
    M=sqrt(((i-1)*Ve*t-x(i)).^2+(H-y(i)).^2);
    ctheta=((i-1)*Ve*t-x(i))./M;
    stheta=(H-y(i))./M;
    x(i+1)=x(i)+Vw*t*ctheta;
    y(i+1)=y(i)+Vw*t*stheta;
    if y(i+1)>=H
        break;
    end
end
L=x(i+1);
T=x(i+1)/Ve;
```

分别取步长为 0.1, 0.05, 0.01, 0.001，得到的结果如图左所示。





可以看出仿真的图像和 Euler 方法图像确实吻合，步长越小越精确。这再一次印证了这两种迭代在实质上是一致的。

为体现仿真过程，我们又取 0.01 的步长，作出导弹运行轨迹和敌舰运行轨迹，如图右所示。

【任务 3 & 4】

当基地发射导弹时，敌舰立即用仪器发觉。假定敌舰为一高速快艇，它即刻以 135km/h 的速度与导弹方向垂直的方向逃逸。问导弹何时何地击中敌舰？试建立数学模型并求解，用仿真方法进行计算。

设导弹和敌舰在初始时刻（t=0）时分别位于（0,0）和（0,H），$v_w = 450(km/h)$，$v_e = 135(km/h)$，H=120(km)，敌舰和导弹在 k 时刻的所在位置分别用(x_{ek}, y_{ek})和(x_{wk}, y_{wk})来表示。

$$\begin{cases} x_{wk+1} = x_{wk} + v_w \cdot \tau \cdot \cos\theta_k \\ y_{wk+1} = y_{wk} + v_w \cdot \tau \cdot \sin\theta_k \end{cases}$$

$$\begin{cases} x_{ek+1} = x_{ek} + v_e \cdot \tau \cdot \sin\theta_k \\ y_{ek+1} = y_{ek} - v_e \cdot \tau \cdot \cos\theta_k \end{cases}$$

其中θ_k为导弹飞行方向的倾角

$$\theta_k = arctan\frac{y_{ek} - y_{wk}}{x_{ek} - x_{wk}}$$

$$\cos\theta_k = \frac{x_{ek} - x_{wk}}{\sqrt{(x_{ek} - x_{wk})^2 + (y_{ek} - y_{wk})^2}}$$

$$\sin\theta_k = \frac{y_{ek} - y_{wk}}{\sqrt{(x_{ek} - x_{wk})^2 + (y_{ek} - y_{wk})^2}}$$

每次迭代以后，敌舰和导弹之间的距离都会被更新。关于仿真的终止条件，原来是简单的两者间距离小于一个固定的常数，后来我们发现步长不同，这个量没法控制，因为距离的缩减是和步长有关的。最终不断尝试方案，得到了现在的算法：当前后两次迭代对应的两个敌我距离差值，缩小到敌我双方的单位速度差值之内时，仿真停止，输出击中所需时间 T。

MATLAB 代码如下所示：

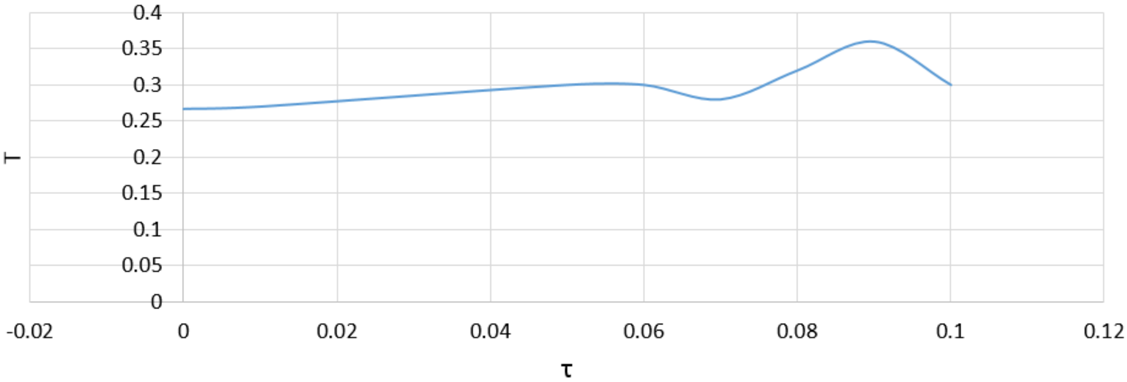
```
function [xw,yw,xe,ye,k,distance,I]=m4_5(t)
H=120;
Ve=135;Vw=450;
xw(1)=0;yw(1)=0;xe(1)=0;ye(1)=H;
I=0;
for k=1:10e6
    distance=sqrt((yw(k)-ye(k))^2+(xw(k)-xe(k))^2);
    theta=abs(atan((ye(k)-yw(k))/(xe(k)-xw(k))));
    ctheta=cos(theta)*sign(xe(k)-xw(k));
    stheta=sin(theta)*sign(ye(k)-yw(k));
    xw(k+1)=xw(k)+Vw*t*ctheta;
    yw(k+1)=yw(k)+Vw*t*stheta;
    xe(k+1)=xe(k)+Ve*t*stheta;
    ye(k+1)=ye(k)-Ve*t*ctheta;
    I=I+t;
    new_distance=sqrt((yw(k+1)-ye(k+1))^2+(xw(k+1)-xe(k+1))^2);
    if abs(new_distance-distance)<t*(Vw-Ve);
        break;
    end
end
```

通过改变步长，我们可以观察到该方法的运行结果，在步长较大的时候 T 会发生不稳定的波动，但当步数逐渐的时候，T 趋于稳定。具体数据如表 6 所示

τ	0.1	0.09	0.08	0.07	0.05	0.01	0.001	0.0005	0.0001	0.00001	0.000001
T	0.3	0.36	0.32	0.28	0.3	0.27	0.267	0.267	0.2668	0.26668	0.266668

表 6

下图直观反映了步长和算得时间之间的关系，在 0.05 以上的步长中明显存在一个波动，之后算法就趋于收敛。



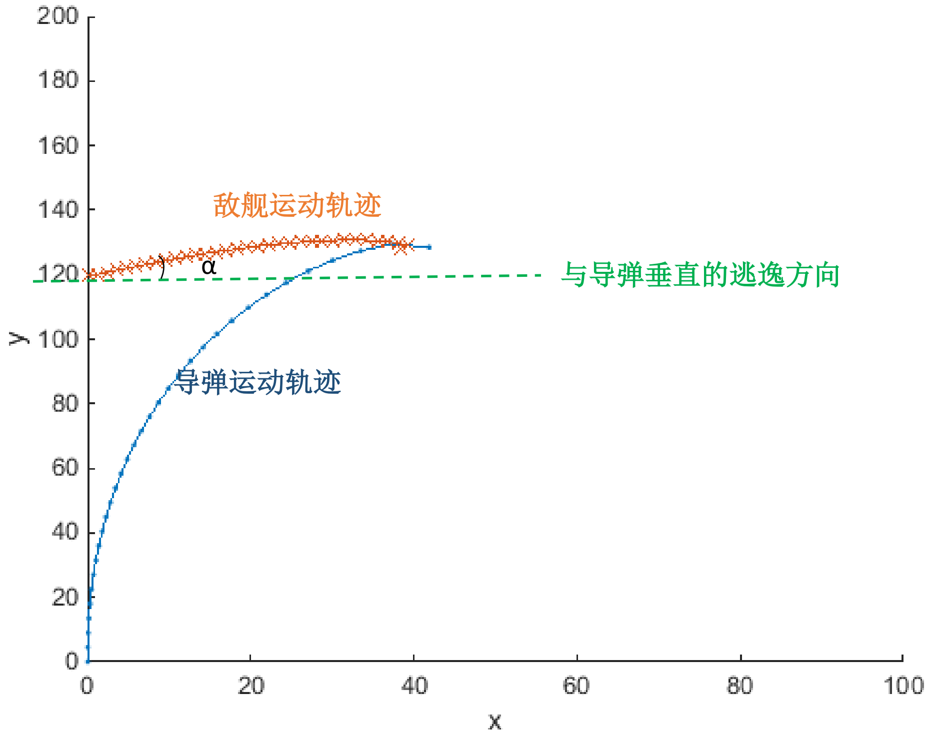
【任务 5 & 6】

如果敌舰以 135km/h 的速度与导弹方向成固定夹角的方向逃逸，问导弹何时何地击中敌舰？建立数学模型，并选择若干特殊角度进行计算，求出怎样的角度逃逸较好？

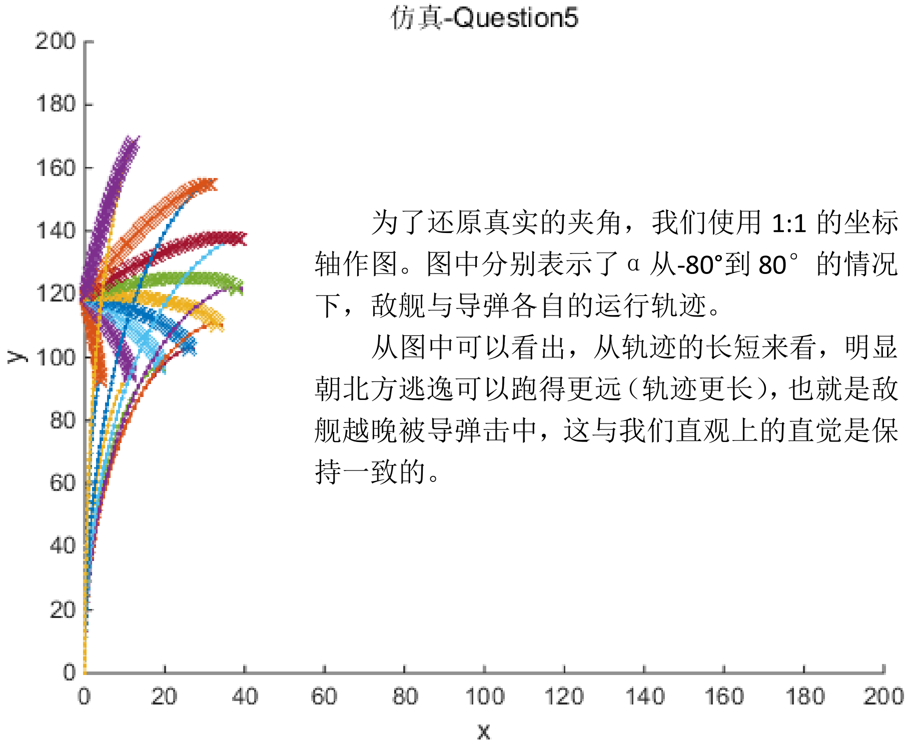
```
function [xw,yw,xe,ye,k,distance,I]=m4_6(t,alpha)
H=120;
Ve=135;Vw=450;
xw(1)=0;yw(1)=0;xe(1)=0;ye(1)=H;
I=0;
for k=1:10e6
    distance=sqrt((yw(k)-ye(k))^2+(xw(k)-xe(k))^2);
    theta=abs(atan((ye(k)-yw(k))/(xe(k)-xw(k))));
    ctheta=cos(theta)*sign(xe(k)-xw(k));
    stheta=sin(theta)*sign(ye(k)-yw(k));
    xw(k+1)=xw(k)+Vw*t*ctheta;
    yw(k+1)=yw(k)+Vw*t*stheta;
    theta=theta+pi*alpha/180;
    ctheta=cos(theta)*sign(xe(k)-xw(k));
    stheta=sin(theta)*sign(ye(k)-yw(k));
    xe(k+1)=xe(k)+Ve*t*stheta;
    ye(k+1)=ye(k)-Ve*t*ctheta;
    I=I+t;
    new_distance=sqrt((yw(k+1)-ye(k+1))^2+(xw(k+1)-xe(k+1))^2);
    if abs(new_distance-distance)<t*(Vw-Ve);
        break;
    end
end
```

MATLAB 的代码如上图所示。

对比前面的【任务 3 & 4】来说，现在的情况与之不同的仅仅在于逃逸方向。所以除此之外的模型建立过程与之前保持一致，不再赘述。我们的代码在原本垂直的基础上再加了一个夹角 α 。例如，若 α =20°，则敌舰与导弹的实际夹角为 110° 。该角度的几何意义如下图所示：



接下来，我们取不同的 α 值，分别作出夹角不同时敌我双方的运行轨迹，并记录敌舰被击中所需的时间。



为了还原真实的夹角，我们使用 1:1 的坐标轴作图。图中分别表示了 α 从-80°到 80° 的情况下，敌舰与导弹各自的运行轨迹。

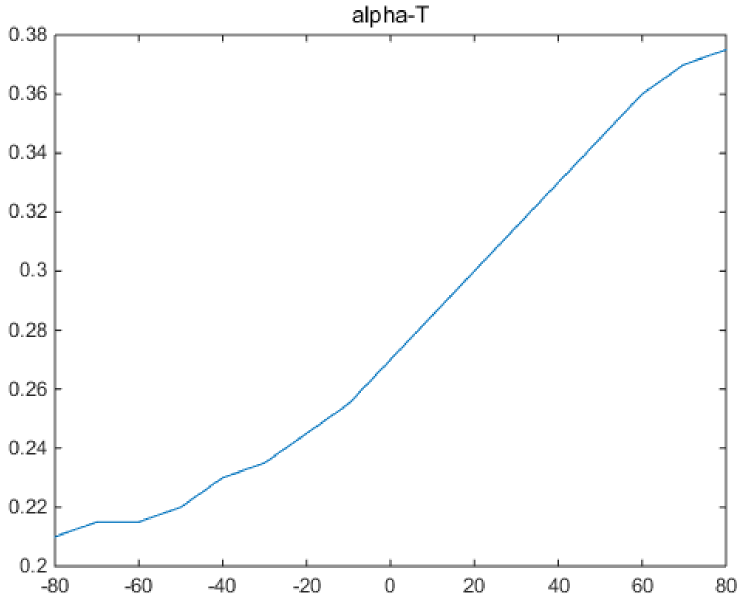
从图中可以看出，从轨迹的长短来看，明显朝北方逃逸可以跑得更远（轨迹更长），也就是敌舰越晚被导弹击中，这与我们直观上的直觉是保持一致的。

在从图中获得一个大概的感受以后，我们再来看记录的不同 α 所对应的时间 T。如表 7 所示。

α	-80	-70	-60	-50	-40	-30	-20	-10
T	0.22	0.26	0.22	0.23	0.23	0.24	0.25	0.26
α	10	20	30	40	50	60	70	80
T	0.39	0.30	0.32	0.33	0.35	0.36	0.37	0.38

表 7

下图是由表中数据所绘制的 α-T 图，我们可以看到，α 越大，时间越长，逃离效果越好。需要提及的是，由于-90° 和 90° 这两个角度太特殊，模拟出的效果在这两点发生了突变，所以我们把这两种情况排除在外。在[-80,80]的范围下，T 随 α 单调递增，在 80°时逃离的时间最长（此时夹角是 170° ）。事实上，在 α=-90°时，敌舰直面导弹追踪方向而开，显然违背常识，直接排除。在 α=90° 时，导弹和敌舰都是沿正北方向匀速直线运动，该问题转化为简单的追及问题，通过 135T+120=450T 的方程直接可以求解 T=0.38095，相对于 α=80°时更长一些。所以沿正北方向逃逸是最好的。



三．结果分析

对于这一实际问题建模得出的结果，我们认为是非常符合直觉的。

因为作为逃离方，总是要背对着追赶方向运动，但我们原本以为有一些夹角逃离会比较好，让导弹有调整追击方向的时间损耗。但其实调转方向时在现实世界中会有空气阻力、水的阻力等，在建模中被我们理想化地暂时忽略了。在理想模型的情况下，根据我们的结果来看，朝背离导弹的方向直线行驶，是最简单且有效的逃离方式。

四．分工情况

王力功：分析问题，数学建模，编程实现及作图

徐小博：组织文章结构，整理数据及编写，结果分析