# 导弹跟踪问题

5122409009 王力功

5121209117 徐小博

1. 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？

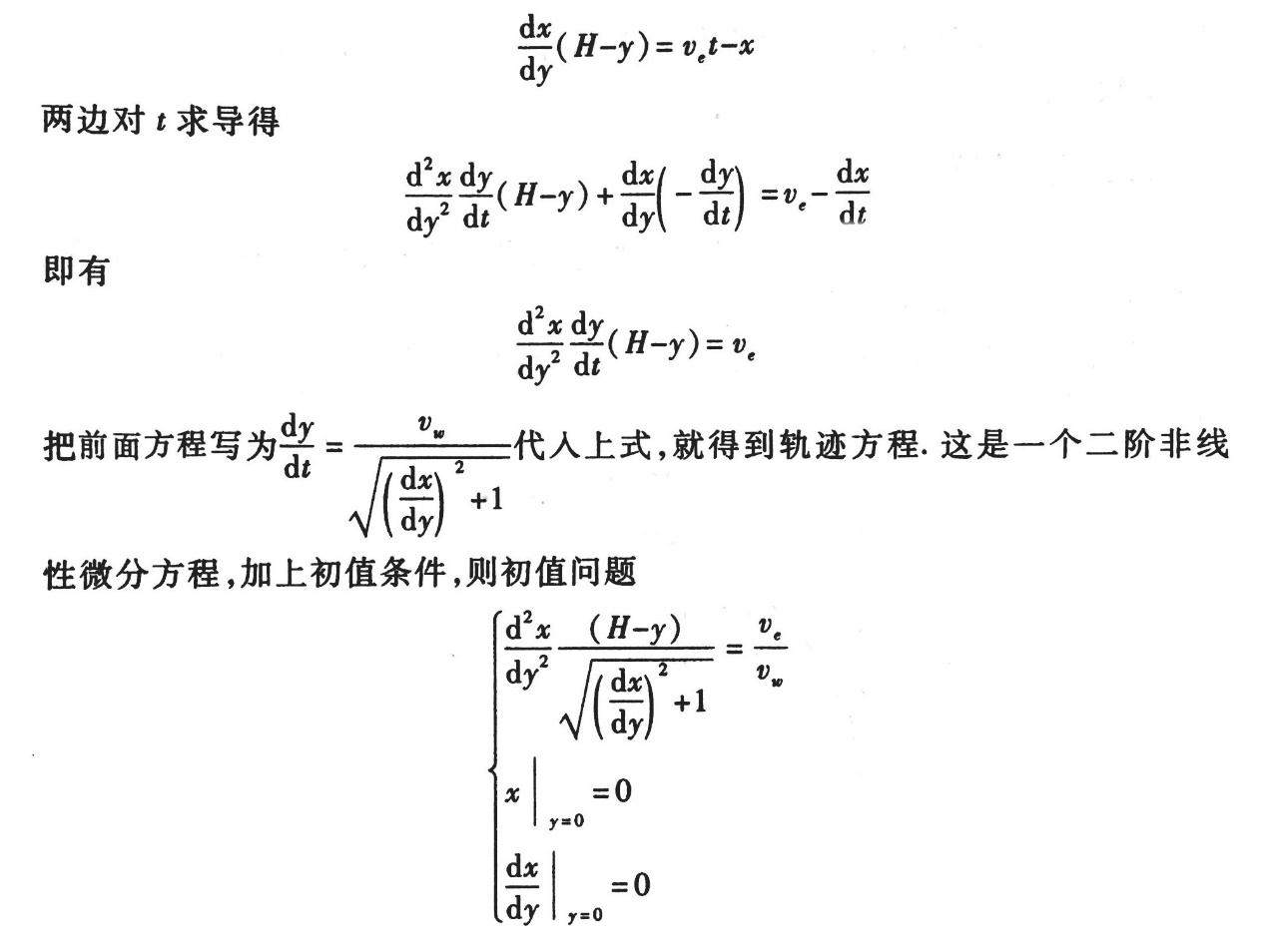
1. 数学模型

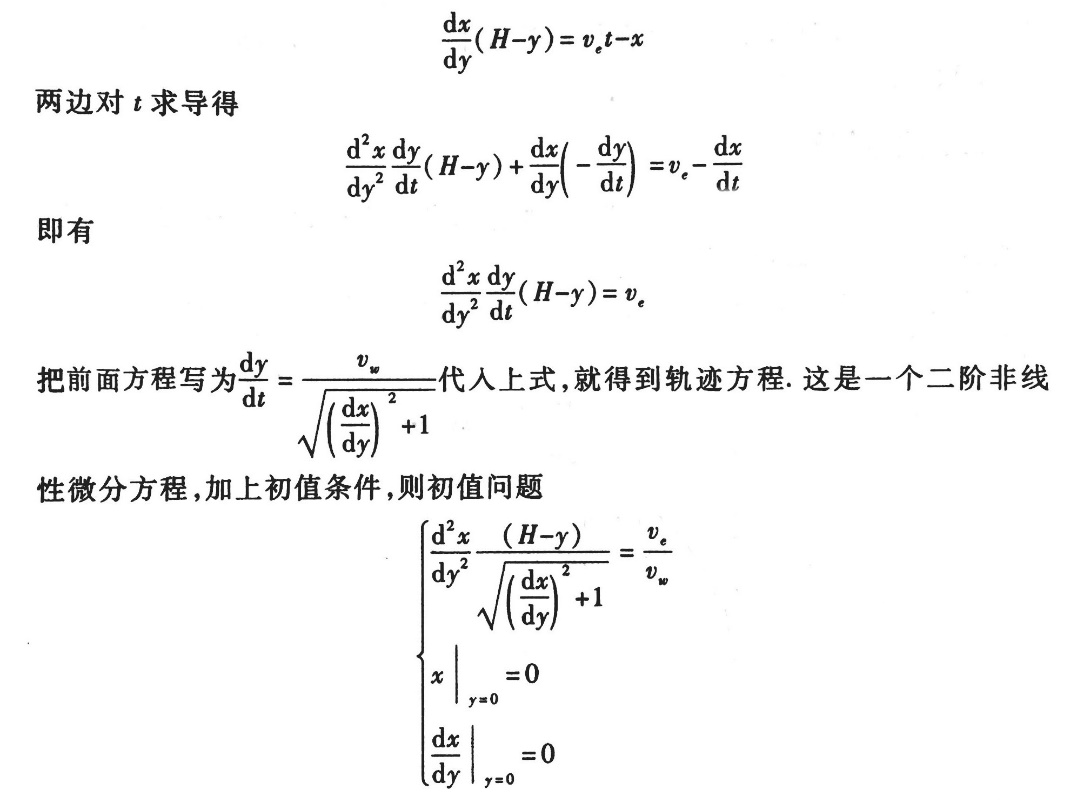
设导弹在原点O(0,0)，x轴指向正东，y轴指向正北。t=0时，导弹位于O，敌舰位于A(0,H)，其中H=120(km)。设导弹在t时刻的位置为P(x(t), y(t))，由题意得

其中。在t时刻，敌舰位置M(vet,H)，其中。

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

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，





即为导弹轨迹的数学模型。

* 1. **解析法**

令，记，则可以化为一阶可分离变量方程

参考教材上的推导过程，最终可以求得

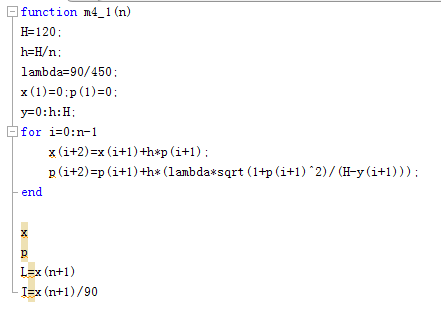
把数据代入，得

* 1. **Euler法**

设导弹到达处的时刻为，那么得到计算的迭代格式为

其中

使用MATLAB，编制文件：



尝试步长为4，得到n=4时的计算结果为

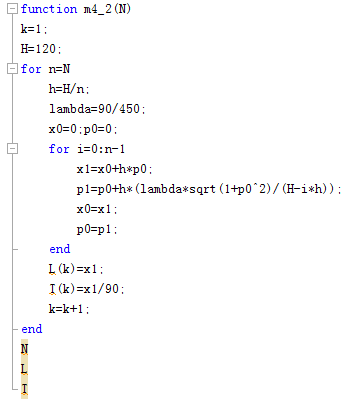
x = 0 0 1.5000 5.0025 11.5254

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：



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

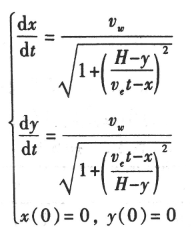
将步长继续增加，观察步长n=[480,600,960,1560,1920,2000]时的结果，如表2所示。

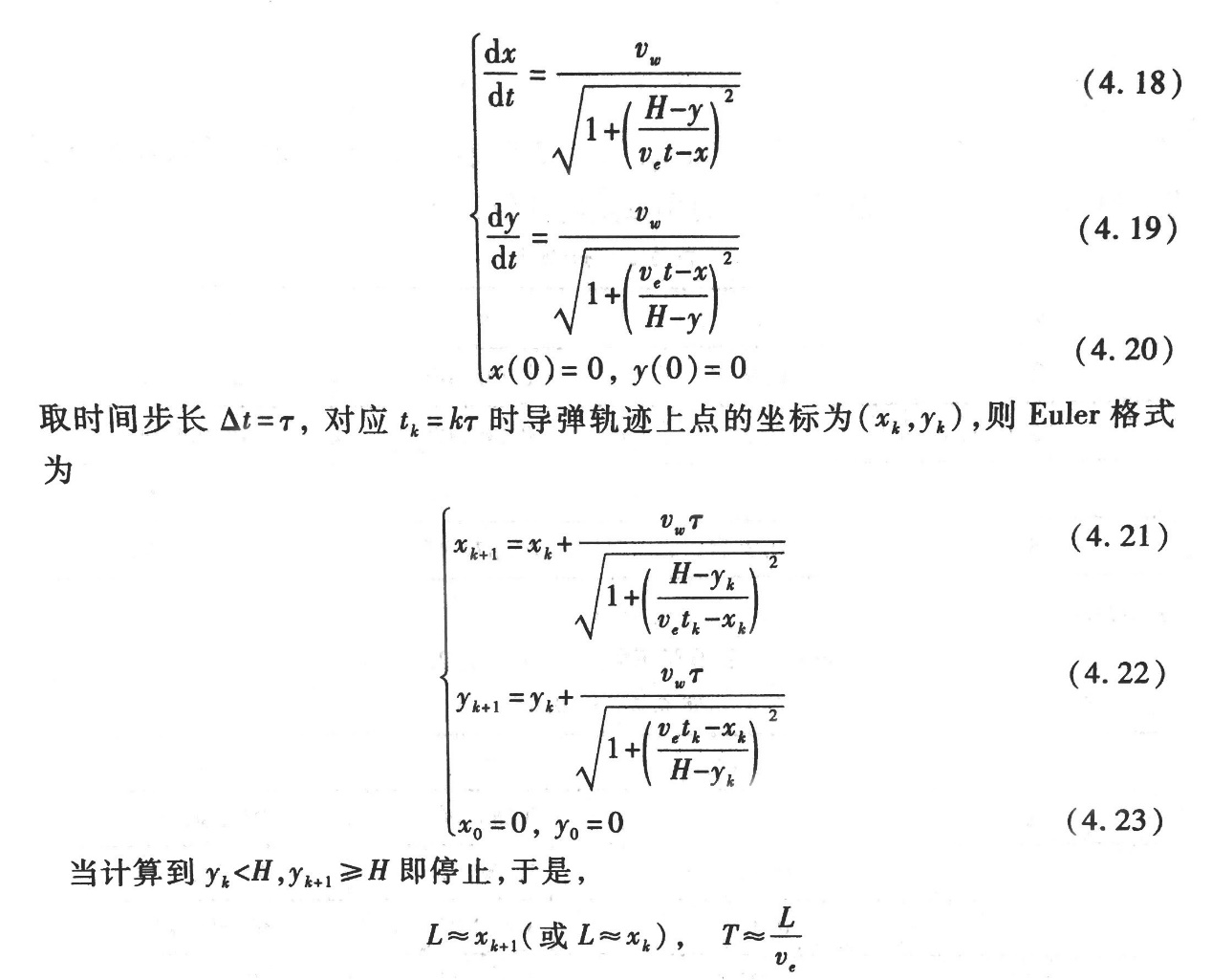
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| n | 480 | 600 | 960 | 1560 | 1920 | 2000 |
| L | 24.4965 | 24.5751 | 24.7034 | 24.7958 | 24.8261 | 24.8315 |
| T | 0.2722 | 0.2731 | 0.2745 | 0.2755 | 0.2758 | 0.2759 |

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





使用MATLAB，代码如下：

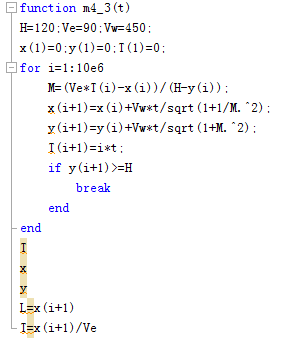


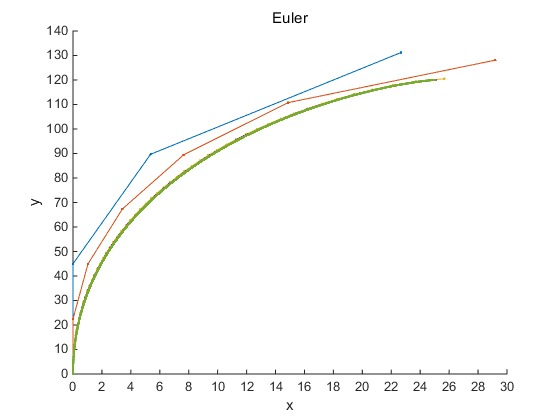
表3是对应不同的步长，用Euler法所得相应的步长推进次数n和计算结果。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 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 |

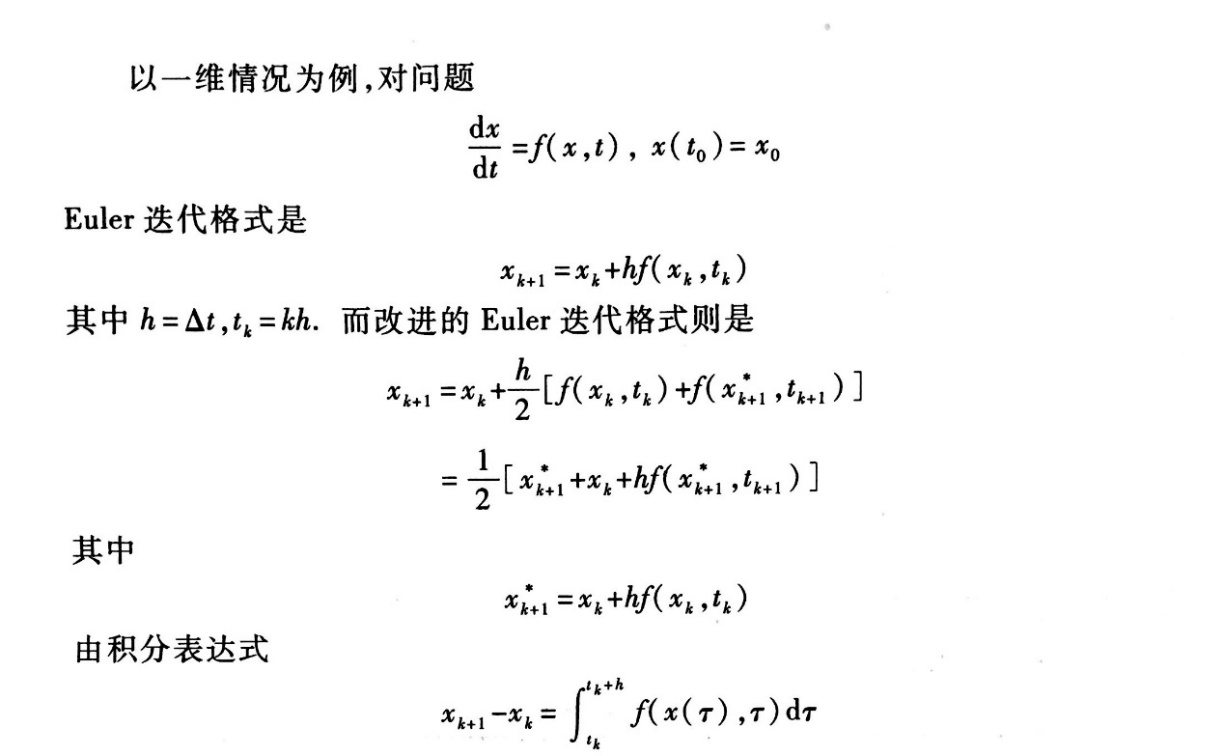
表3

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。



* 1. **改进的Euler法（预报 — 矫正法）**



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

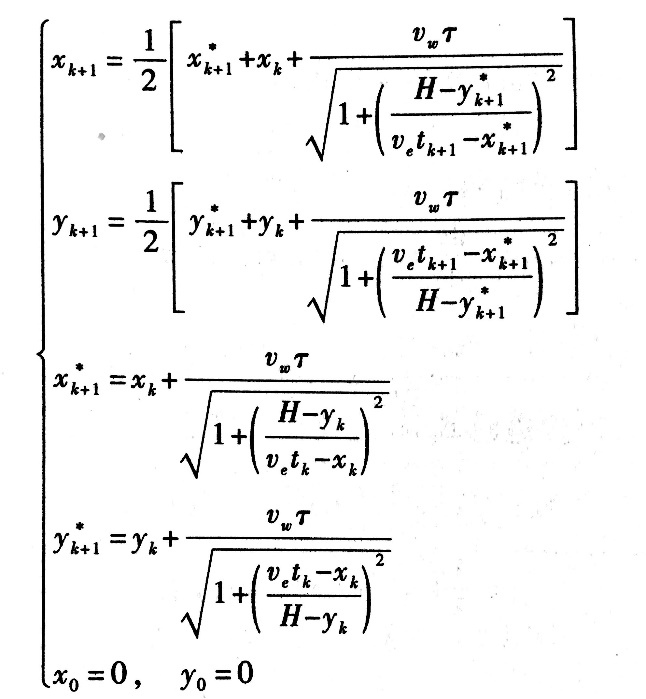
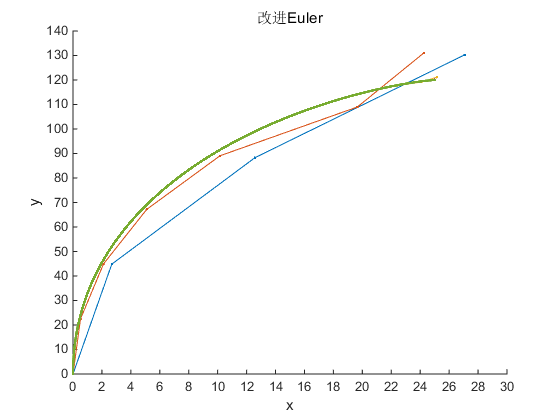


表4是对应不同的步长，用改进的Euler法所得相应的步长推进次数n和计算结果。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 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 |

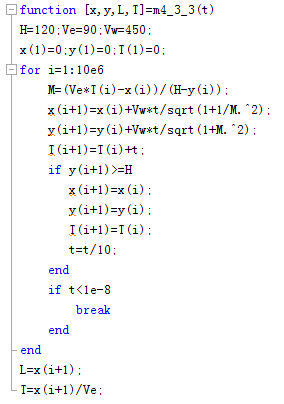
表4



【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，是计算到即停止，然后取，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，而远大于H时，可缩小步长（例如原来的十分之一）以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：



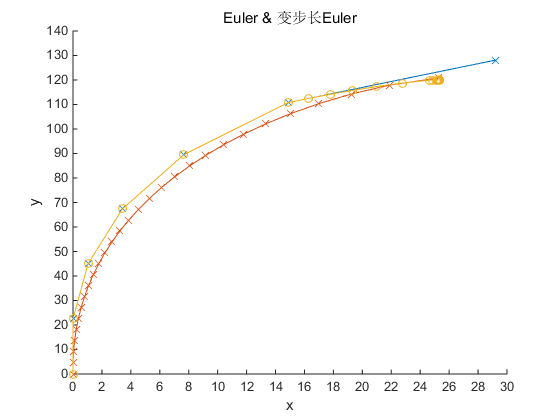
注意到t会在特判下缩小步长，我们依然使用不同的初始步长进行测试，所得相应的步长推进次数n和计算结果如表5所示。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 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，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

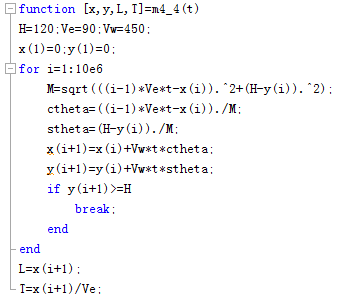
注意到改变步长法在初始步长为0.05时的迭代效果和标准Euler方法步长在0.01（前者的五分之一）时的迭代效果几乎相当，由此可见，改变步长的Euler方法相较于标准Euler方法优化效果显著。

* 1. **仿真方法**

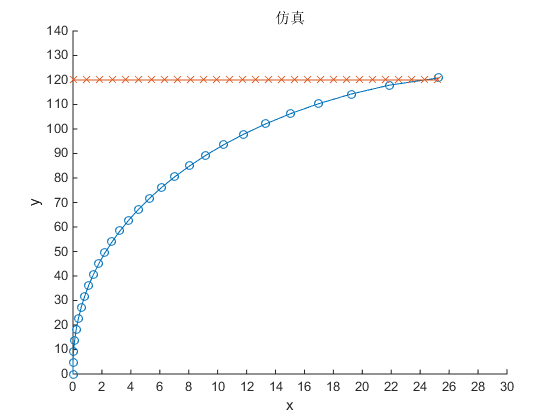
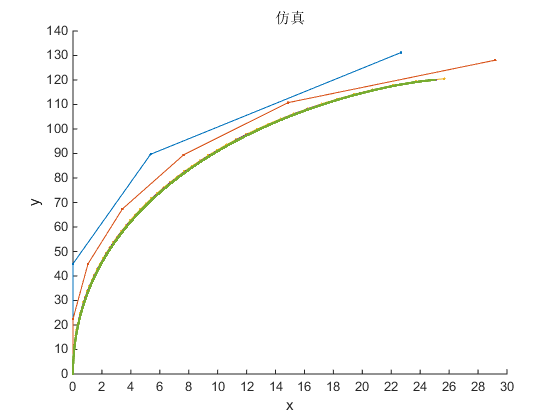
仿真方法同教材，此处略。

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：



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】

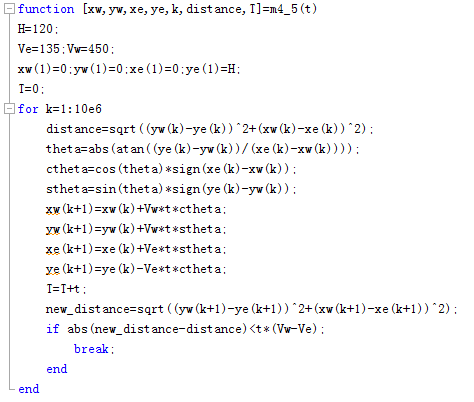
当基地发射导弹时，敌舰立即用仪器发觉。假定敌舰为一高速快艇，它即刻以135km/h的速度与导弹方向垂直的方向逃逸。问导弹何时何地击中敌舰？试建立数学模型并求解，用仿真方法进行计算。

设导弹和敌舰在初始时刻（t=0）时分别位于（0,0）和（0,H），，H=120(km)，敌舰和导弹在k时刻的所在位置分别用和来表示。

其中为导弹飞行方向的倾角

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：

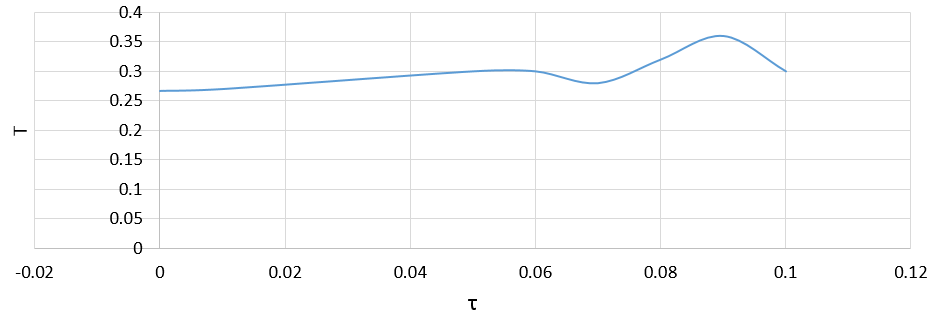


通过改变步长，我们可以观察到该方法的运行结果，在步长较大的时候T会发生不稳定的波动，但当步数逐渐的时候，T趋于稳定。具体数据如表6所示

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

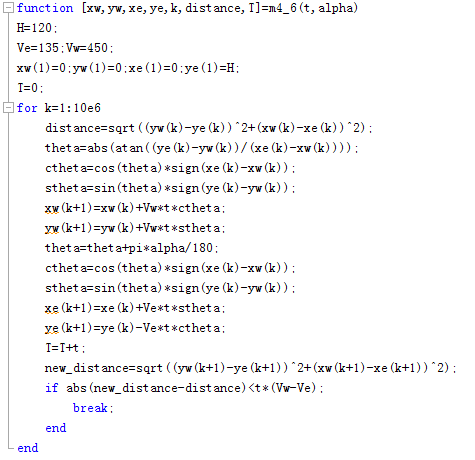
表6

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。



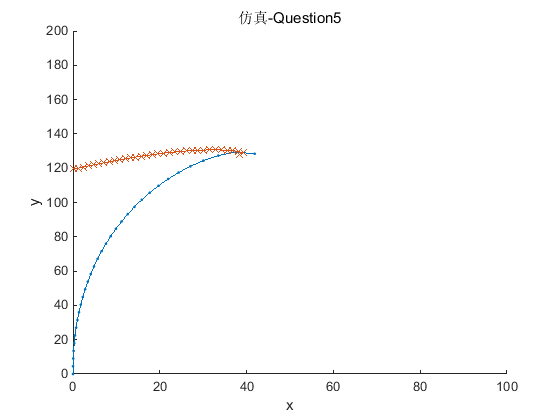
【任务5 & 6】

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？



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：



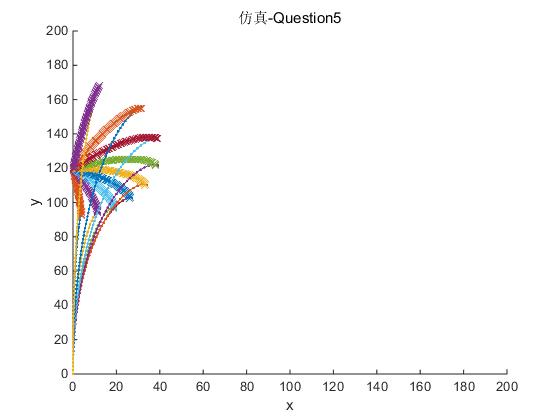
**导弹运动轨迹**

**与导弹垂直的逃逸方向**

**敌舰运动轨迹**

α

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。

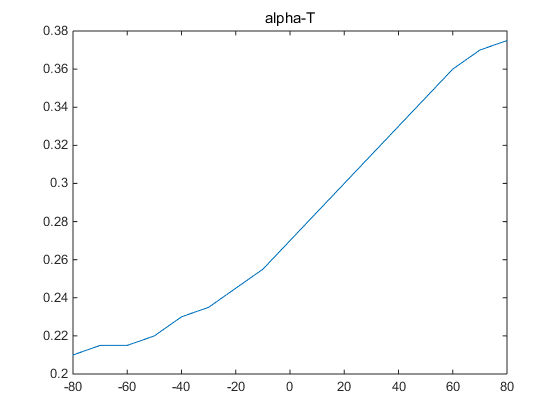
从图中可以看出，从轨迹的长短来看，明显朝北方逃逸可以跑得更远（轨迹更长），也就是敌舰越晚被导弹击中，这与我们直观上的直觉是保持一致的。

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。



1. 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。

1. 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