1. 拦截预测

1.1 拦截导弹方程

拦截导弹方程可以通过解析求解

$$m\frac{dv}{dt}=-c\frac{dm}{dt}$$

解为: $\Delta v=c\ln\frac{m_0}{m_f}$ （m_0拦截导弹加燃料的初始重量，m_f拦截导弹最终重量）

加上重力，以及速度和高度相关的阻力

$$m\frac{dv}{dt}=F-c\frac{dm}{dt}\hat{v}$$

$F=-g\hat{z}-D\hat{v}$

其中阻力$D=\frac{1}{2}pv^2AC_D$（A 为横截面积，C_D阻力系数）

因为阻力与高度有关，所以$p(z)=p_0e^{-\frac{z}{H}}$

仿真的时候，把 H 设为 8000

综上我们可以得到如下方程

$$m\frac{dv_x}{dt}=(-\frac{1}{2}p_0e^{-\frac{z}{H}}v^2AC_D-c\frac{dm}{dt})\hat{v}_x$$

$$m\frac{dv_y}{dt}=(-\frac{1}{2}p_0e^{-\frac{z}{H}}v^2AC_D-c\frac{dm}{dt})\hat{v}_y$$

$$m\frac{dv_z}{dt}=(-mg-\frac{1}{2}p_0e^{-\frac{z}{H}}v^2AC_D-c\frac{dm}{dt})\hat{v}_z$$

采用正向欧拉数值解法

$$v_{x,t+1}=v_{x,t}+\frac{dt}{m}(-\frac{1}{2}p_0e^{-\frac{z}{H}}v^2AC_D-c\frac{dm}{dt})\hat{v}_{x,t}$$

$$v_{y,t+1}=v_{y,t}+\frac{dt}{m}(-\frac{1}{2}p_0e^{-\frac{z}{H}}v^2AC_D-c\frac{dm}{dt})\hat{v}_{y,t}$$

$$v_{z,t+1}=v_{z,t}+\frac{dt}{m}(-mg-\frac{1}{2}p_0e^{-\frac{z}{H}}v^2AC_D-c\frac{dm}{dt})\hat{v}_{z,t}$$

单位向量 v^x、v^y 和 v^z 用于控制拦截导弹发射的方向

1.2 运动学方程与有限差分

运动学方程用于在模拟过程中更新拦截导弹的位置，以及预测敌方导弹的位置

$$\begin{cases} x = x_0 + vt \\ v = v_0 + at + \frac{1}{2} j^2 \\ a = a_0 + jt \end{cases}$$

拦截导弹更新位置: $x = x_0 + vt$

敌方导弹更新位置:采集到 4 个数据点后，使用有限差分法近似计算了速度、加速度和时间导数 j

$$\begin{cases} x = x_0 + \frac{\mathrm{v}vt}{g}(1 - e^{-\frac{gt}{vt}}) \\ \mathrm{v} = \mathrm{v}_0 + at + \frac{1}{2} j^2 \\ a = a_0 + jt \end{cases}$$

2. 拦截

2.1 基于优化的方法

使用了一种简单的基于优化的基本方法，使导弹能够尽早拦截敌方导弹

$$\min_{t,v'} \quad \frac{1}{2} t^2$$

$$\begin{aligned} s.t. \quad & x'_0 + v'_x (t - \delta) = x_0 + v_x t \\ & y'_0 + v'_y (t - \delta) = y_0 + v_y t \\ & z'_0 + v'_z (t - \delta) - \frac{1}{2} g (t - \delta^2) = z_0 + v_z t + \frac{1}{2} g t^2 \\ & t - \delta > 0 \\ & \| v' \| \leq V \end{aligned}$$

（x, y, z)对应于敌方导弹，（x’，y’，z’）对应于反应导弹

δ是敌我发射导弹的时间差

V 是我方导弹可以达到的最大速度

其实这个方法就是找到一个拦截导弹的速度最大限度地减少敌方导弹飞行的时间，在其到达区域之前拦截它

2.2 数值方法

执行了一系列数据收集、投影和近似拦截步骤来提高精度问题

数值方法算法

输入：拦截导弹的初始位置

输出：敌方导弹和拦截导弹诡计

....

```
观测敌方导弹;
从敌方导弹弹道上收集四个位置点;
当没有拦截成功时
    前推敌方导弹 N 个 timestep 的轨迹
    for k in [1, ..., N] do
        for t in [1, ..., k] do
            拦截导弹加一个 timestep
            敌方导弹加一个 timestep
            dt=拦截导弹与敌方导弹之间的距离;
           如果拦截预测成功，则
                k_optimum = k;
                Break k loop;
           如果 d_t>d_t-1
                Store d_t-1;
                Store k;
                Break t loop;
       End
    End
    如果没有拦截成功
        k_optimum = k 对应 min d_t-1;
   前推拦截导弹 N 个 timestep
   从敌方导弹弹道上收集四个位置点
End
```