# **反潜深弹命中概率研究**

摘要

反潜深弹引爆深度对潜艇命中可能性影响较大，本文依据深弹落点位置，建立概率统计模型，探究深弹引爆深度与潜艇命中概率关系，主要解决以下问题。

针对问题一：首先，本文引进一种新的探索深弹落点杀伤范围内与潜艇最近点的欧式距离，根据此距离，得出潜艇可能被深弹命中的所有平面区域，见图二；其次，深弹引爆深度不同，潜艇可能被命中平面区域不同，找出不同深度下潜艇可能被命中的对应平面区域，建立概率统计模型，在找出的平面区域下对概率密度函数求积分即为潜艇被命中的概率，并给出潜艇被命中概率与引爆深度函数关系。这里需注意，积分区域分为矩形与圆弧两种，需分别计算，最终潜艇被命中概率为所有平面区域概率之和；最后，随着引爆深度增大，潜艇被命中区域逐渐增大，直至图二整个平面区域，此时深弹命中概率最大，具体命中概率趋势见图四。根据题目已知数据，将其代入模型，计算出深弹最大命中概率为8.536%，这种情况是本文建立模型的某些引爆深度情况下的一种。

针对问题二：首先，考虑潜艇在垂直方向上存在定位误差，单一应用问题一模型计算不准确，需要对其改进；其次，只考虑潜艇深度影响，根据潜艇中心单尾正态分布密度函数给出潜艇垂直方向上被命中概率模型。这样，将其与问题一模型五相乘即为潜艇被命中概率模型；最后，问题一中已求给出水平最大命中概率，要想潜艇被命中概率最大，只需垂直命中概率最大即可，采用python网格搜索法，当引爆深度为149.5米时，潜艇垂直被击中概率最大，为0.317，最终计算出问题二中潜艇被击中概率为2.71%。

针对问题三，首先确定一枚深弹命中概率，确定发射九枚时至少一枚命中潜艇的概率；其次，应用python网格搜索法，搜寻概率最大的设计方案；最后，根据搜索方案，给出最优投弹方案，当引爆深度为137.78 m，投弹距离间隔a、b均为30米时，深弹命中概率最大，概率为55.2%。

**关键词：**欧式距离、概率密度函数、命中概率，网格搜索、python

# **问题重述**

反潜作战中，深水炸弹（深弹）在二战时期大放光彩。随着技术不断改进发展，鱼雷逐渐替代深弹，并成为主要反潜作战武器。然而，在某些特定的复杂海域下，深弹具有成本低廉、并且抗外部干扰性强的优点，目前许多国家依旧采用深弹反潜技术。

在一般反潜作战中，首先侦察飞机利用通讯侦查设备对目标潜艇进行精确定位，然后将潜艇具体信息传给反潜飞机，此时，反潜飞机进行投弹。通常情况下，潜艇发现被跟踪时立即静默并就地隐蔽。

结合问题具体背景，根据已知条件建立合适的数学模型，解决下面三个问题。

问题一：潜艇中心位置其深度定位准确，没有误差，潜艇的3个坐标相互独立，水平坐标服从正态分布。探究深弹的最大命中率与落点水平坐标和定深引信引爆的深度之间联系，给出深弹命中概率最大的方案，并列出最大命中率表达式。

问题二：在问题一的前提下，潜艇3个方向位置定位都存在误差，深度标准差为40m，实际深度最小值为120m，且潜艇中心深度符合单边截尾正态分布。请确定合理的引爆深，使深弹命中率最大，并写出深弹命中率表达式。

问题三：实际情况中，单枚深弹命中率偏低，杀伤力较弱。要提高命中率，一般采用投放多枚的方式。若同时投放九枚引爆深度都相同的深弹，落点的形状为一个矩形，请给出一种投弹设计方案，使得至少一枚命中的概率最大。

# 问题分析

**2.1 问题一的分析：**

主要考虑深弹在潜艇平面尺度区域范围内引爆，还是潜艇平面尺度区域范围外引爆，分为三种情形。

（1）深弹落点在平面尺度范围内：如果深弹与潜艇垂直距离在深弹杀伤半径范围之内，命中概率为1；如果深弹与潜艇垂直距离在深弹杀伤半径范围之外，命中概率为0；深弹能否命中潜艇，取决于深弹所处深度位置。根据潜艇平面尺度范围，应用联合概率密度函数计算深弹命中潜艇概率。

（2）深弹落点在平面尺度范围外：深弹垂直下落过程中，如果深弹与潜艇水平距离在深弹杀伤半径范围之内（落点位于潜艇四个侧面平面区域），命中概率为1；深弹垂直下落过程中，如果深弹与潜艇水平距离在深弹杀伤半径范围之外（落点位于潜艇四个侧面平面区域），命中概率为0；根据四个侧面能命中尺度范围，应用联合概率密度函数计算深弹命中潜艇概率。

（3）深弹落点在平面尺度范围外：当深弹落点位于潜艇上表面上方或下表面下方时，如果下落过程中，深弹与潜艇上最近的点的距离如果在深弹杀伤半径范围之内，则命中概率为1,，找到这部分区域，应用联合概率密度函数计算深弹命中潜艇概率（该部分为圆弧，积分上下限要特别注意）。

结合三种情形，给出投弹命中概率最大方案，并计算最大命中率。

**2.2 问题二的分析：**

在问题一基础上，引入了潜艇深度误差，要研究深度误差对潜艇命中概率的影响，主要考虑以下三种情形。

（1）水平命中概率与问题一相同，包含三种情况。

（2）需研究潜艇深度命中概率。由于潜艇实际深度服从截尾正态分布，所以，在深度命中概率计算中，要结合深度位置单边截尾正态分布密度函数来分析深度命中概率。结合问题一分析中三种情况，分别计算出对应情况下深度命中概率，将结论与问题一分析中三种情况下水平概率分别乘积再求和，即为投弹最大命中概率表达式，并计算出投弹最大命中概率。

（3）由于深度存在不确定性，最佳的定深引信引爆深度不再固定，在投弹命中基础上，本文通过网格搜索法寻出不同深度下投弹最大命中概率。

**2.3 问题三的分析：**

结合问题二，设计一个投弹方案，使得至少一枚深弹命中概率最大，主要考虑以下两种情形。

1. 结合问题二，确定不同深度下发射一枚深弹的命中概率。
2. 通过网格搜法优化阵列的行间距和列间距，以及引爆深度，找到使得整体命中概率最大化的方案，使得至少一枚深弹时命中概率最大。这涉及到同时优化平面布局和垂直引爆深度两个维度。优化投弹阵列的间隔，要满足两点，既要确保深弹覆盖足够大的平面区域，又要避免投弹点过于分散，导致某个区域的命中概率过低。此外，在垂直方向上，定深引信的引爆深度也需要调整。

# 模型假设

1、假设侦察机发现潜艇时，潜艇意识到被发现。

2、假设潜艇静默后，位置不发生变化。

3、假设深弹为一质点。

4、假设在深弹杀伤半径范围内即被视为命中，且杀伤效果一样。

5、假设深弹投落位置准确无误。

6、假设深弹在水下垂直下落过程中方向不会受外部因素影响。

7、假设深弹正常引爆。

8、假设单枚深弹命中概率相互独立。

9、假设海底地形等复杂情况对深弹命中率无影响。

10、假设潜艇水平位置误差服从正态分布，深度定位服从截尾正态分布。

# 符号说明

|  |  |  |
| --- | --- | --- |
| **符号** | **含义** | **单位** |
|  | 潜艇长度 | m |
|  | 潜艇宽度 | m |
|  | 潜艇高度 | m |
|  | 潜艇中心的预估深度 | m |
|  | 潜艇中心实际深度 | m |
|  | 水平定位误差标准差 |  |
|  | 深度定位误差标准差 |  |
| R | 深弹的最大杀伤半径 |  |
|  | 潜艇本体区域命中概率 |  |
|  | 潜艇右侧区域命中概率 |  |
|  | 潜艇上方区域命中概率 |  |
|  | 潜艇右上角四分之一圆区域命中概率 |  |
|  | 定深引信引爆深度 | m |
|  | 标准正态分布的密度函数 |  |
|  | 标准正态分布的分布函数 |  |
|  | 深弹引爆深度 | m |
|  | 垂直命中概率，基于深度误差 |  |
| P0 | 某一枚深弹的总命中概率 |  |
|  | 垂直方向的命中概率 |  |
|  | 水平方向的命中概率 |  |

# 模型建立与求解

## 问题一模型的建立与求解

5.1.1搜索潜艇边界与深弹距离最近的点的欧式距离

为了探索深弹投弹区域，并能计算投弹命中最大概率，本文给出一种新的探索潜艇边界与深弹距离最近的点的距离计算方法。

（1）

说明：

，是立方体在方向上的最小值和最大值.

，是立方体在方向上的最小值和最大值

，是立方体在方向上的最小值和最大值,

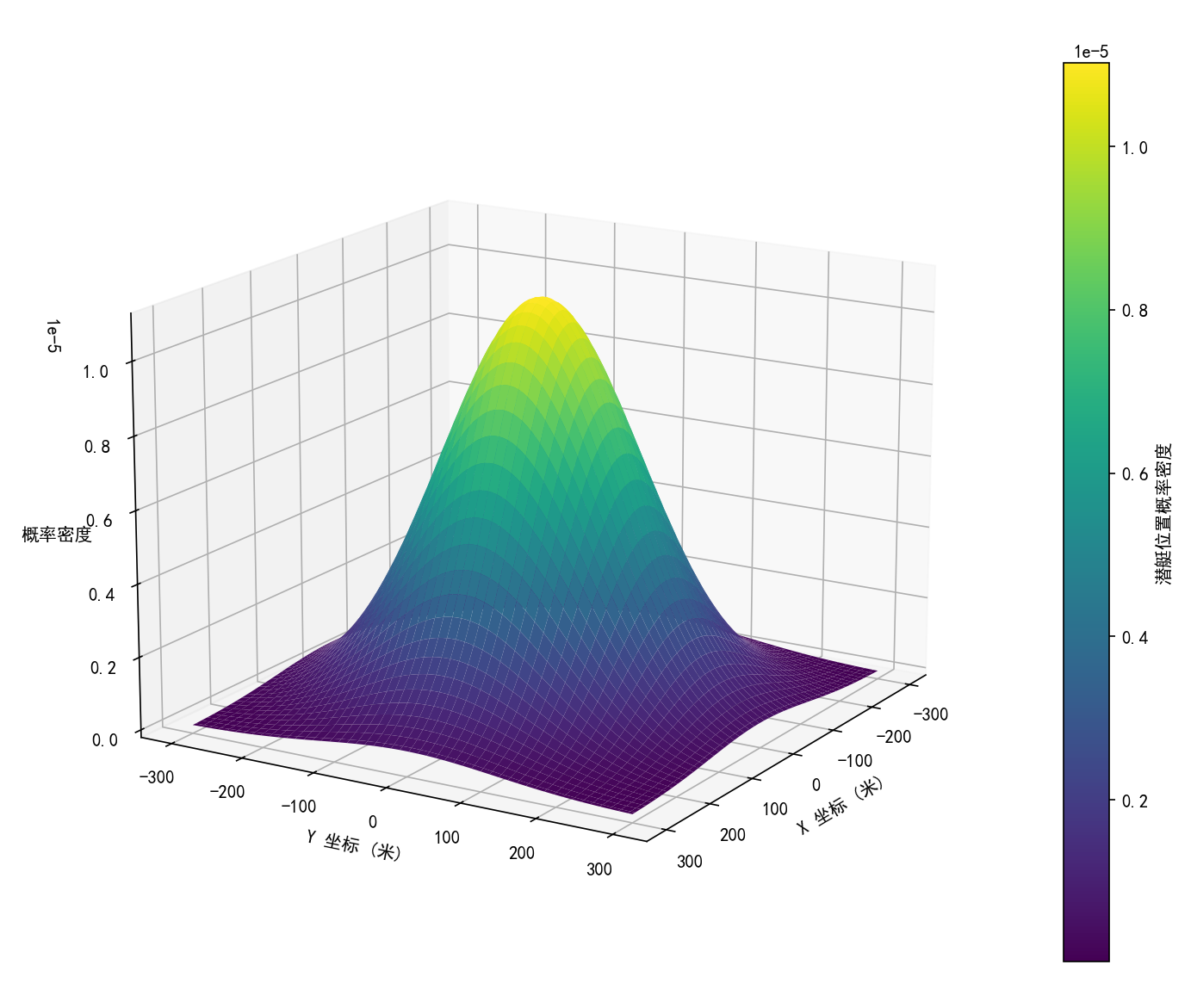
5.1.2预备知识

定义[1]：设二维随机变量服从正态分布，且互相独立，当，则概率密度函数为：

（2）

5.1.3模型建立

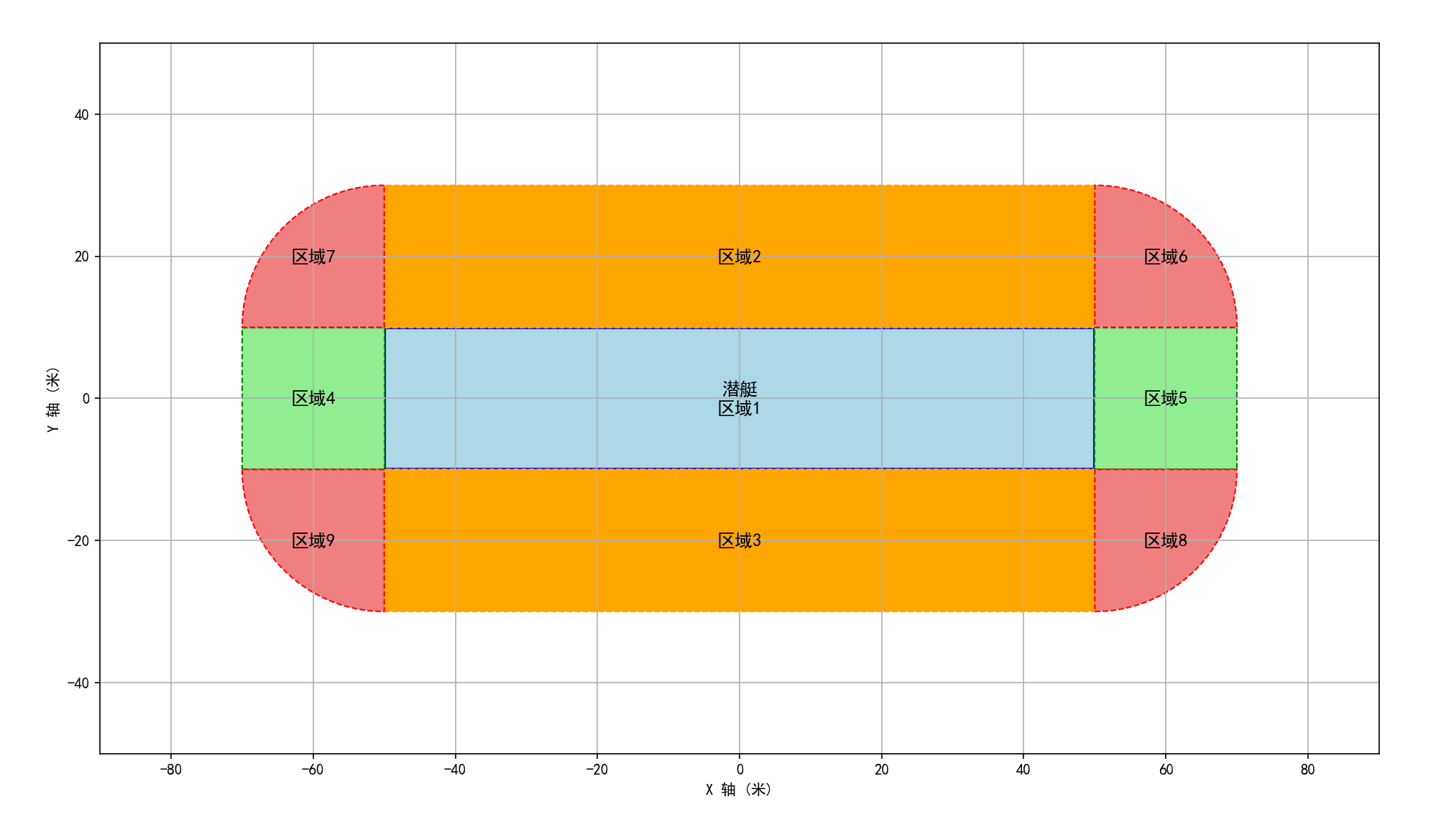
（1）由于潜艇中心位置坐标是3个相互独立的随机变量，水平坐标服从正态分布，在深度定位没有误差的情况下，要分析投弹最大命中率与落点平面坐标及定深引信引爆深度关系，优先观察游艇中心位置在水平区域可能出现最大概率，从而确定投弹落点水平位置坐标。结合游艇中心位置坐标（0,0,150），绘制出游艇中心位置正态分布图，如图所示，图一。



图一 潜艇中心位置三维概率密度分布

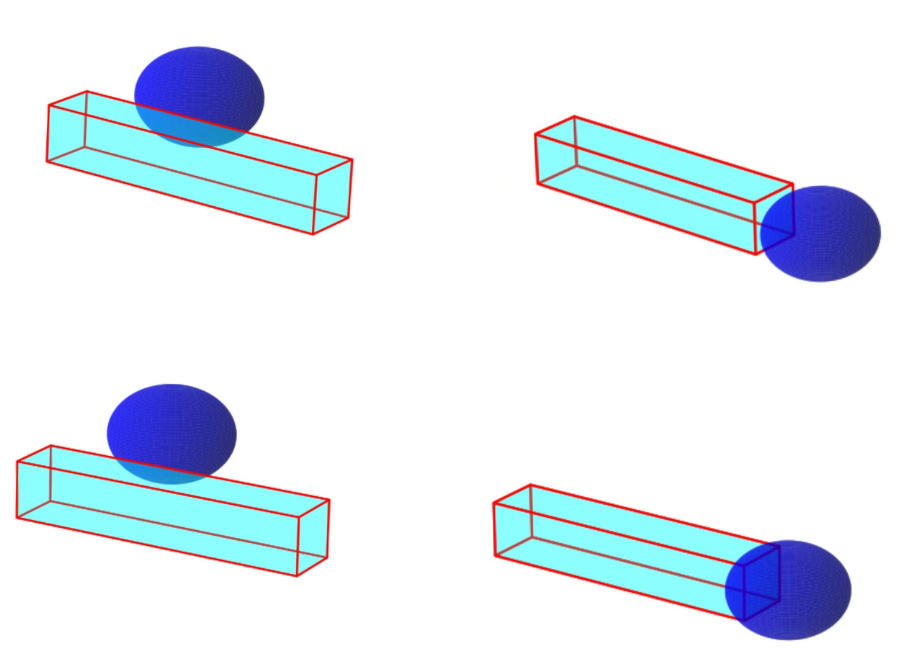
（2）对于正态分布来说，概率密度最大的点（峰值）是位于其均值的位置，所以两个相互独立的随机变量的正态分布，概率密度函数的最大值也出现在两个随机变量分别等于它们的均值时。结合图一可以得出，投弹落点位于潜艇中心的水平坐标，即（0，0）时命中潜艇概率最大。

要分析命中概率模型，首先要确定投弹命中潜艇水平区域。根据本文建立的欧式距离（1）式，可得出深弹落点命中潜艇时的最大水平区域，如图二:



图二 深弹命中潜艇落点平面区域

1. 根据问题一所给引爆条件，分析可得出深弹引信引爆深度在[117.5,182.5]（单位：米）范围内才有可能命中潜艇，其它深度命中概率为0。已知潜艇在水平面上可能被深弹命中区域，且水平区域上水平坐标互相独立，服从正态分布，要计算潜艇可能被命中概率，只需对水平坐标概率密度函数求积分即可。但是，深弹在不同深度引爆，潜艇被命中区域是不同的，如图三。



图三 深弹落点情况

当引爆深度为117.5米时，潜艇可能被命中区域为图二区域1，引爆深度增加时，潜艇被命中的区域向外延伸，直至图二全部区域。针对不同的引爆深度，潜艇被命中概率不同，计算潜艇被命中概率，需找到不同深度下潜艇被命中积分区域。

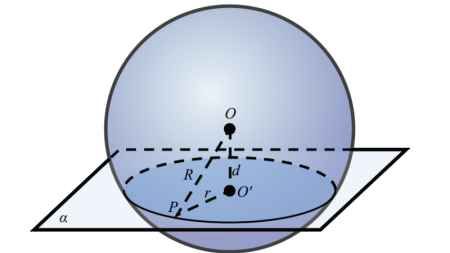
结合图二分析可以发现，潜艇表面积分区域的外界拓展线长与爆炸半径和潜艇上表面截出圆的半径相同（如图四），截出圆的半径为，其中：

当引爆深度在潜艇上下表面20米外，则无法命中潜艇，因此不作考虑。

当引爆深度在潜艇上下表面内，此时的潜艇被命中积分区域最大，此时Q=20。

当引爆深度在潜艇上下表面32米内，且不再潜艇上下表面内，则

。以此可以确定不同引爆深度下的命中潜艇积分区域。



图四 潜艇表面区域延伸拓展线长

（4）具体概率模型分两种情况来求解。第一种情况，命中概率区域为矩形。第二种情况，命中概率区域为四分之一圆。

第一种情形：命中概率区域为矩形

模型一：区域1（潜艇平面)，深弹落点区域，潜艇可能被命中概率为：

 （5）

模型二：区域2、3，上下对称，计算其中一个即可。

深弹落点区域，潜艇可能被命中概率为：

 （6）

模型三：区域4、5，左右对称，计算其中一个即可。

深弹落点区域，潜艇可能被命中概率为：

 （7）

第二种情形，命中概率区域为四分之一圆。

需根据所在圆的方程，依据积分优先原则（先后或先后x）确定积分上下限,进而计算出命中概率。

模型四：区域6、7、8、9，均为四分之一圆，计算其中一个即可。圆的方程为：

（8）

深弹落点区域，潜艇可能被命中概率为：

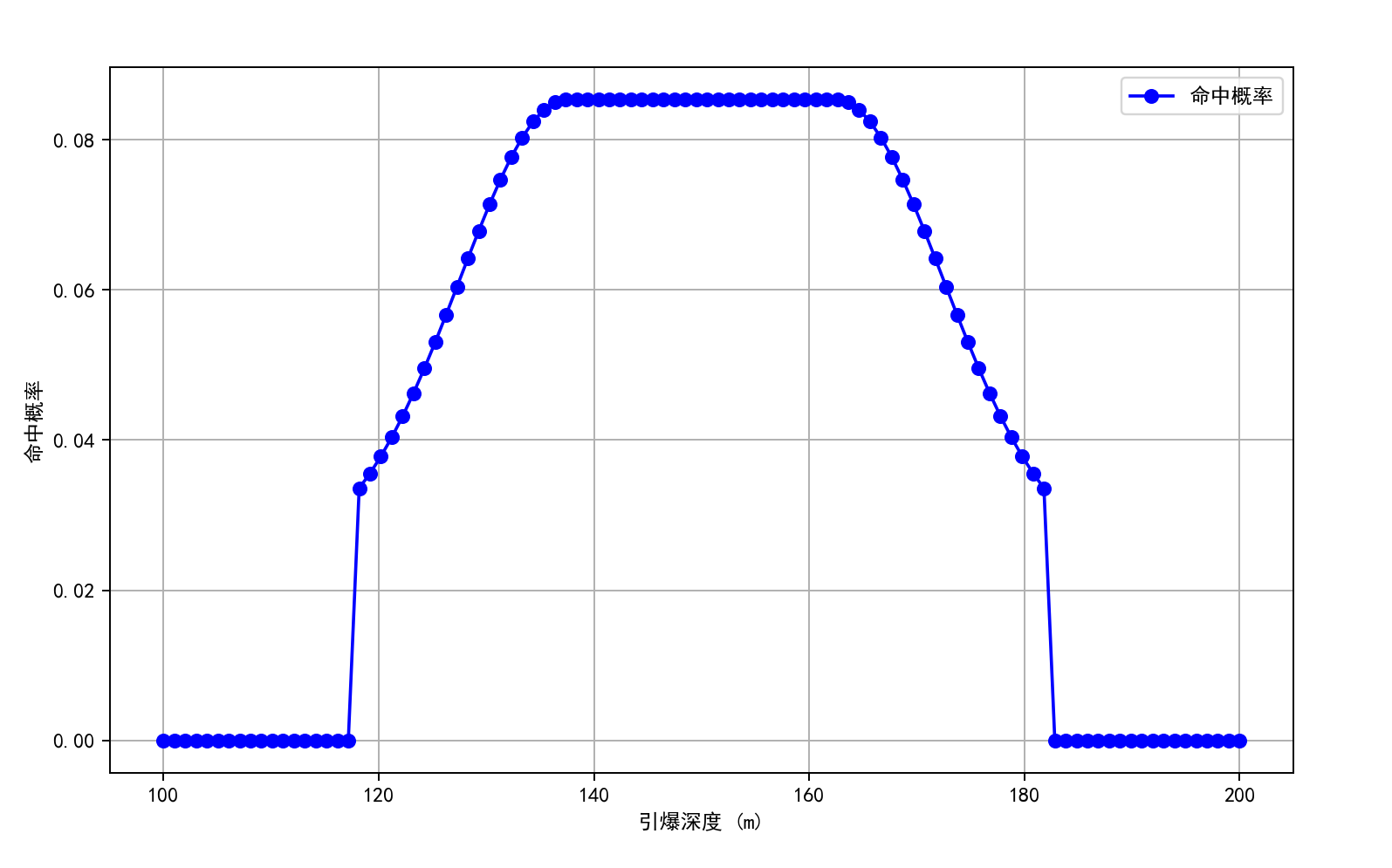
 (9)

深度不同，潜艇被命中概率不同。所以，潜艇被命中的概率与深度之间关系为：

模型五：潜艇被命中的概率

(10)

计算任意深度命中概率，只需将深度值代入模型五计算即可得出潜艇命中概率。其深弹落点不同引爆深度与潜艇命中概率关系如图四,详见python程序[2]附录1。



图四 不同引爆深度潜艇被命中概率

说明：

深弹的引爆深度小于潜艇的上表面深度，即定深深度为，则可能会发生触发引爆深弹，也可能发生触发定深引爆。深弹定深位置处于潜艇上方，且潜艇在深弹的杀伤范围内。即引爆深度和，只有定深引爆才能命中。

深弹不在潜艇的投影平面范围内，但当潜艇在任意一处的表面在深弹的爆炸范围内。那么深弹仍然可能伤害潜艇。

深弹的引爆深度位于潜艇上表面深度到深弹杀伤半径之内，则引爆，最大深度应为潜艇上表面深度加上杀伤半径。

 (11)

（5）最大命中概率计算

由图四可以看出，深弹下落过程中，随着引爆深度增大，潜艇被命中概率逐渐增大，达到一定引爆深度时，潜艇被命中概率不再增加，这时潜艇被命中概率达到最大。此时，计算潜艇被命中概率只需对图二整个落点区域进行积分，这是模型五一种情况。根据题目已知条件，将对应具体参数代入模型一、二、三、四，其结果如下：

 （12）

 （13）

 （14）

 （15）

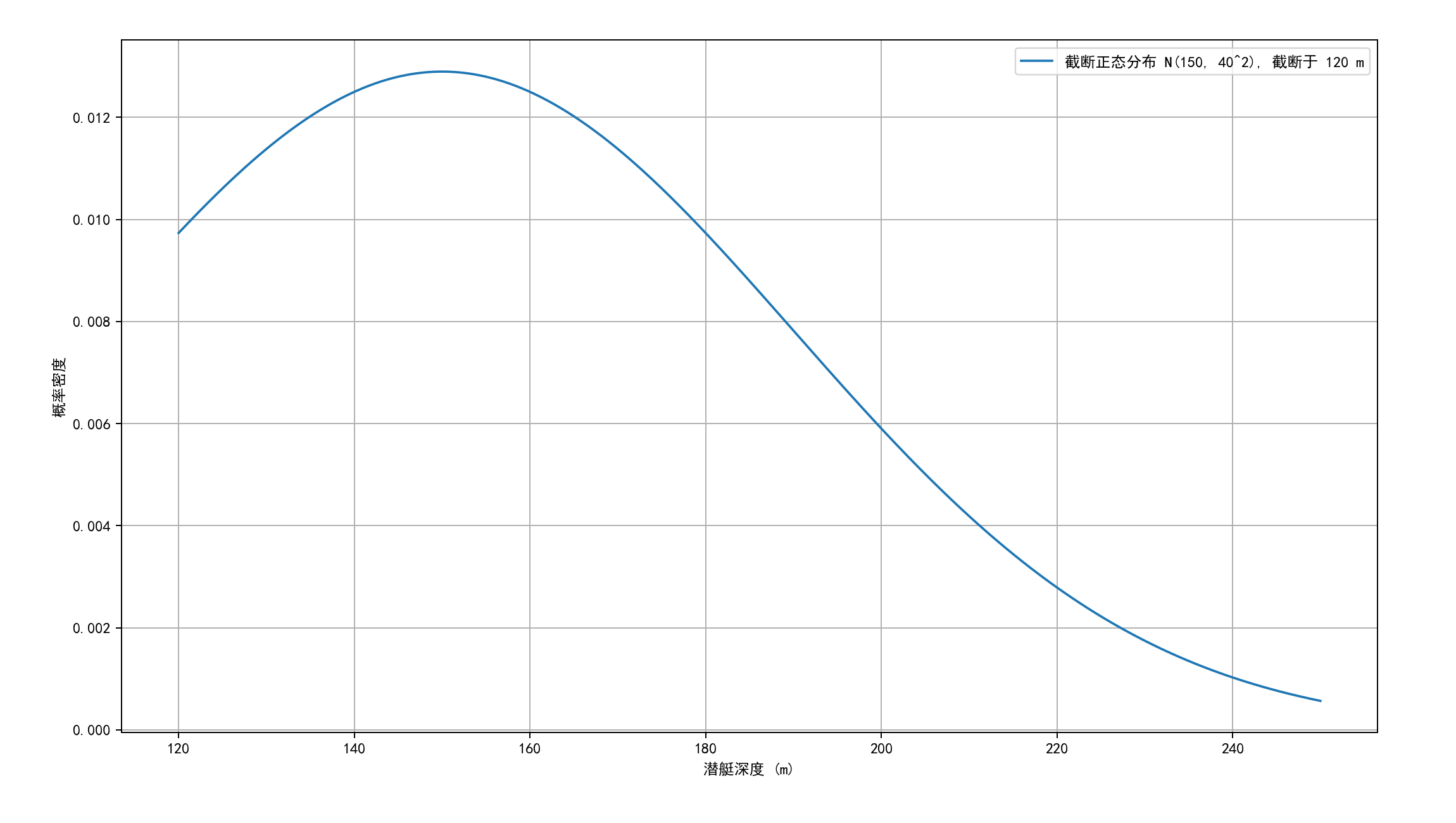
根据模型五可知，潜艇被命中最大概率为。计算结果为：0.08536，与图四完全吻合，可以得出，引爆深度（单位：米）时，潜艇被命中概率最大。

## 问题二模型的建立与求解

5.2.1潜艇被命中概率在垂直方向限制条件

结合问题一，潜艇在垂直方向上的定位存在误差，也就是说单纯使用问题一模型计算问题二中深弹命中概率是不准确的，需要考虑潜艇深度误差对潜艇被命中概率的影响，需对问题一模型进行改进。

由于潜艇在深度方向上服从单边截尾正态分布（如图五），因此，在问题一模型基础上，再考虑潜艇深度误差，从而得出潜艇不同深度下被命中概率模型。



图五 潜艇单边截尾正态分布密度函数

5.2.2模型建立

（1）根据潜艇在垂直方向上满足单边正态分布密度函数，

所以，只考虑潜艇深度影响，潜艇被命中概率为：

（16）

（2）在问题一研究的基础上，结合(10)式，可得出潜艇被命中概率模型。

模型六：

 （17）

（3）模型六最大概率计算

要计算最大概率，由问题一可知，最大值为0.08536，当最大时，潜艇被命中

概率最大。而

（18）

其中，。

将问题已知条件参数代入表达式，应用python网格搜索法[3]，详见附录2。计算出

最大值为0.317，最优引爆深度为149.5米，此时，问题二中投弹命中概率最大值为0.08536\*0.317=0.0271352。

## 问题三模型建立与求解

5.3.1概率计算

在问题三中，为了增大命中概率，同时投射9枚深弹在平面上，投弹点呈3×3的阵列形状，行间距为 a，列间距为b，为了最大化命中潜艇的概率，这里需要考虑两点。

（1）计算不同引爆深度下，一枚深弹命中概率

根据问题一、问题二模型，每一枚深弹不同深度下的命中概率，根据模型六可得：

 （19）

（2）发射9枚深弹时，至少一枚命中概率

当多枚深弹投放时，每枚深弹的命中都是独立的，因此，一枚深弹不中的概率为

，得出至少一枚投中概率为：

 （20）

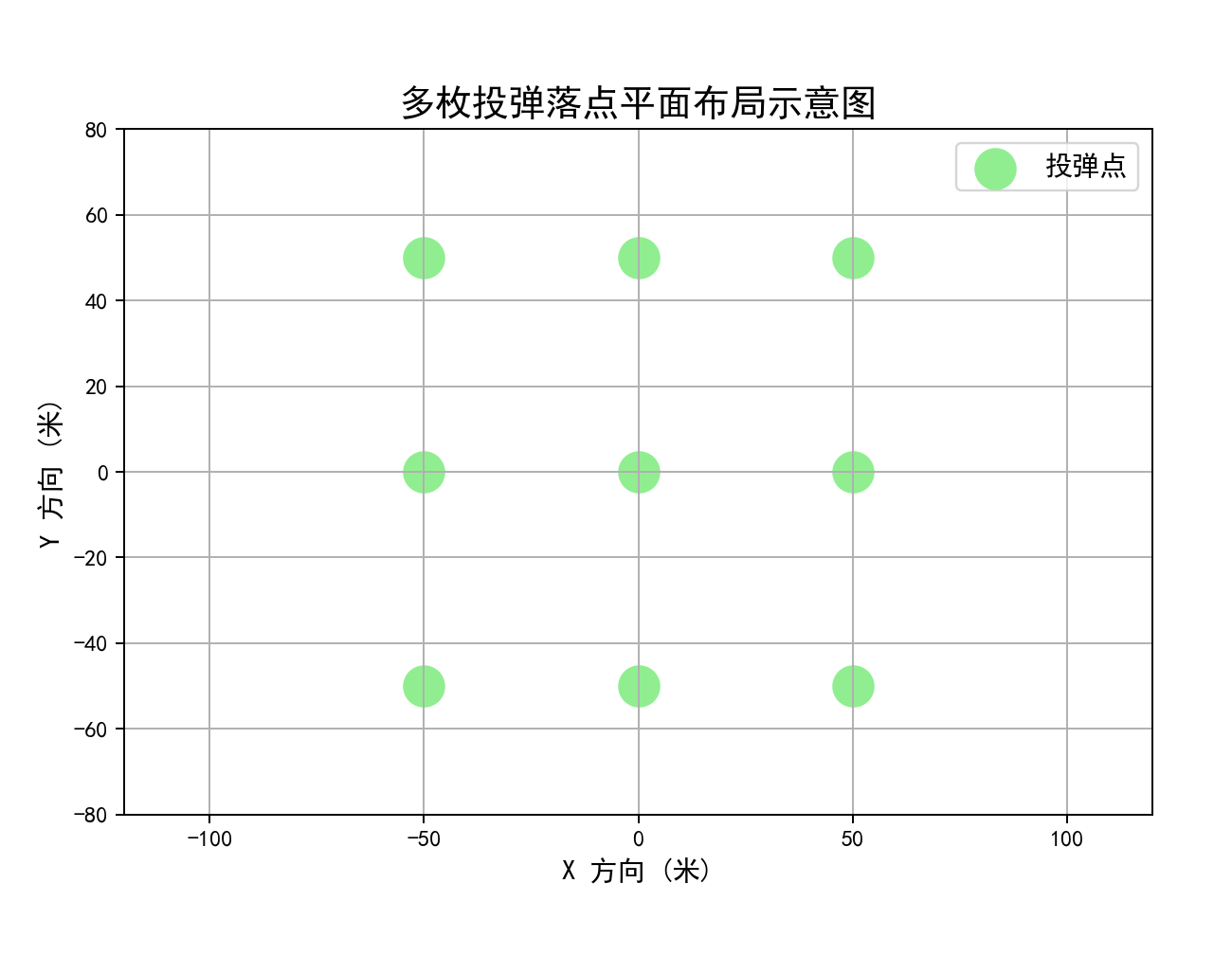
5.3.2布局优化

为了最大化命中概率，我们可以对引爆深度和投弹点的行间距、列间距进行优化。优化的目标是找到使得最大的参数组合。具体步骤如下：

设置引爆深度 的搜索范围为 [120m, 200m]，步长为 1m。设置行间距a和列间距b的搜索范围为 [30m, 100m]，步长为 1m。对每一组、a、b组合，计算至少一枚深弹命中的总概率。记录最大命中概率及其对应的参数组合。

通过这种优化方法，本文应用python网格搜索法[4][5]，详见附录3，通过搜索找到最佳的投弹布局和引爆深度，从而在给定条件下最大化至少一枚深弹命中潜艇的概率。

最终本文得出的结论为：最佳引爆深度: 137.78 m, 最佳行间距为30 m, 最佳列间距30 m（见图六）, 最大命中概率: 0.55203。



图六 深弹落点平面布局

# 模型评价与推广

## 模型的评价

6.6.1模型优点

（1）本文引入全新的搜索潜艇在深弹杀伤范围内距离潜艇最近点的欧式距离，容易找出潜艇被命中的所有可能区域，能准确计算深弹命中概率，没有误差。

（2）本文给出的概率统计模型，对于连续性随机变量，模型计算结果精准。

（3）本文应用 Python程序进行网格搜索，给出最优发射方法，精准深弹发射位置大大提高了深弹命中率，大大节省了时间，且计算便捷，节省了计算成本。

6.6.2模型缺点

文章模型建立过程中，好多因素理想化，比如外部客观条件对深弹发射命中率影响，结果可能不够理想，对于问题的精确性计算不一定高，需要进一步提升模型精确度。

## 模型的推广

文中建立模型过程中充分考虑了水平误差、定位误差对深弹发射概率影响，不考虑自然因素、人为因素的情况下，如果深弹、潜艇数据参数都准确的情况下，本文建立模型计算结果合理准确，可以推广到研究深弹攻击一般几何体的情形中。

# 参考文献

[1] 龚兆龙等，《概率论与数理统计》，北京大学出版社，2009.

[2] 林信良等，《Python程序设计教程》，清华大学出版社，2017

[3] 余强等，《Python科学计算》, 电子工业出版社, 2016.

[4] 李航等，《统计学习方法》，清华大学出版社，2012.  
[5] 王万森, 杨元元等，《机器学习与优化算法》，电子工业出版社，2019.

# 附录

|  |
| --- |
| 附录1（水平面上的深弹引爆深度和命中概率关系图）  附录1中的库配置包括了附录2和附录3，运行后面代码请先放上附录1的库配置文件 |
| import numpy as np import matplotlib.pyplot as plt from scipy.integrate import dblquad import matplotlib.pyplot as plt from scipy.stats import truncnorm from scipy.integrate import dblquad  *# 中文显示*  plt.rcParams['font.sans-serif'] = ['SimHei']plt.rcParams['axes.unicode\_minus'] = False  *# 参数* class 配置:  def \_\_init\_\_(self, 长度=100, 宽度=20, 高度=25, 水平误差=120, 垂直误差=40, 中心深度=150, 最小深度=120, 杀伤半径=20):  self.长度 = 长度 *# 长度 (m)* self.宽度 = 宽度 *# 宽度 (m)* self.高度 = 高度 *# 高度 (m)* self.水平误差 = 水平误差 *# 水平误差的标准差 (m)* self.垂直误差 = 垂直误差 *# 深度误差的标准差 (m)* self.中心深度 = 中心深度 *# 中心深度 (m)* self.最小深度 = 最小深度 *#中心深度最小值 (m)* self.杀伤半径 = 杀伤半径 *# 杀伤半径 (m)* 配置参数 = 配置() *# 算动态计水平命中概率*  def 计算水平命中概率(引爆深度, 配置):  深度差 = abs(引爆深度 - 配置.中心深度)  最大概率 = 0.08536 *# max水平命中概率* 最小概率 = 0.0241 *# min水平命中概率* 衰减参数 = 10 *# 高斯衰减参数，控制衰减的速率* if 深度差 <= 12.5:  水平概率 = 最大概率 *# 在上下12.5米范围内，水平概率最大* elif 深度差 <= 32.5:  *# 使用高斯衰减模型来代替指数衰减* 水平概率 = 最小概率 + (最大概率 - 最小概率) \* np.exp(-((深度差 - 12.5) \*\* 2) / (2 \* 衰减参数 \*\* 2))  else:  水平概率 = 最小概率 *# 超过32.5米，水平命中概率为最小值* return 水平概率 *# 使用距离公式计算垂直命中概率* def 计算垂直命中概率(引爆深度, 配置):  深度差 = abs(引爆深度 - 配置.中心深度) *# 计算投弹点与潜艇中心深度的差值  # 当与潜艇中心深度的距离在上下32.5米时，垂直命中概率为1，否则为0* if 深度差 <= 32.5:  return 1.0  else:  return 0.0 *# 计算总命中概率* def 计算总命中概率(引爆深度, 配置):  水平命中概率 = 计算水平命中概率(引爆深度, 配置) *# 使用高斯衰减计算水平命中概率* 垂直命中概率 = 计算垂直命中概率(引爆深度, 配置) *# 垂直概率* return 水平命中概率 \* 垂直命中概率 *# 绘制引爆深度与命中概率的对称关系* def 绘制对称命中概率(配置):  引爆深度值 = np.linspace(100, 200, 100)  总命中概率 = []   *# 150米之前，正常计算* for 引爆深度 in 引爆深度值:  if 引爆深度 <= 150:  概率 = 计算总命中概率(引爆深度, 配置)  else:  *# 对称部分* 概率 = 计算总命中概率(300 - 引爆深度, 配置)  总命中概率.append(概率)  *# 绘制命中概率与引爆深度的关系图* plt.figure(figsize=(10, 6))  plt.plot(引爆深度值, 总命中概率, label='命中概率', color='b', marker='o')  plt.xlabel('引爆深度 (m)')  plt.ylabel('命中概率')  plt.grid(True)  plt.legend()  plt.show() *# 绘图函数* 绘制命中概率(配置参数) |
| 附录2（求解题目二中深弹最大命中概率） |
| *# 定义PDF φ(t)* def phi(z, h0, sigma\_z):  t = (z - h0) / sigma\_z  return (1 / np.sqrt(2 \* np.pi)) \* np.exp(-0.5 \* t \*\* 2) *# 计算Φ(x) 累积分布函数* def Phi(x):  return norm.cdf(x) *# 数值积分计算命中概率* def compute\_Phd(h1, h2, h0, sigma\_z, l):  *# 对PDF进行积分* integral, \_ = quad(lambda z: phi(z, h0, sigma\_z), h1, h2)  *# 计算P\_hd并返回* Phd = integral / (sigma\_z \* (1 - Phi((l - h0) / sigma\_z)))  return Phd *# 网格搜索算法* def grid\_search(objective\_function, h1, h2, num\_points):  grid\_points = np.linspace(h1, h2, num\_points)  best\_z = None  best\_value = -np.inf  for z in grid\_points:  value = objective\_function(z)  if value > best\_value:  best\_value = value  best\_z = z   return best\_z, best\_value *# 定义函数* def objective\_function(z, h1, h2, h0, sigma\_z, l):  return compute\_Phd(h1, h2, h0, sigma\_z, l) *# 参数* def find\_optimal\_depth(h1, h2, h0, sigma\_z, l, num\_points=100):  *# 具体的目标函数* def obj\_func(z):  return objective\_function(z, h1, h2, h0, sigma\_z, l)  *# 网格搜索执行* optimal\_z, max\_value = grid\_search(obj\_func, h1, h2, num\_points)  return optimal\_z, max\_value *# 输入参数* h0 = 150 *# 中心深度 (m)* sigma\_z = 40 *# 深度的标准差 (m)* l = 120 *# 最小深度 (m)* h1 = 137.5*# 上表面* h2 = 162.5 *# 下表面* num\_points = 1000 *# 网格点的数量，可调节 # 寻找最优引爆深度* optimal\_z, max\_value = find\_optimal\_depth(h1, h2, h0, sigma\_z, l, num\_points) *# 输出结果* print(f"垂直最大命中概率 = {max\_value}") |
| 附录3（九枚深弹最优投放方案） |
| *# 定义潜艇的参数* class 配置:  def \_\_init\_\_(self, 长度=100, 宽度=20, 高度=25, 水平误差=120, 垂直误差=40, 中心深度=150, 最小深度=120, 杀伤半径=20):  self.长度 = 长度 *# 潜艇长度 (m)* self.宽度 = 宽度 *# 潜艇宽度 (m)* self.高度 = 高度 *# 潜艇高度 (m)* self.水平误差 = 水平误差 *# 水平的标准差 (m)* self.垂直误差 = 垂直误差 *# 深度标准差 (m)* self.中心深度 = 中心深度 *# 深度 (m)* self.最小深度 = 最小深度 *# 深度最小值 (m)* self.杀伤半径 = 杀伤半径 *# 深弹半径 (m)* config = 配置() *# 动态计算水平命中概率，使用高斯衰减模型* def 计算水平命中概率(爆炸深度, 配置):  深度差 = abs(爆炸深度 - 配置.中心深度)  最大概率 = 0.08536 *# 最大水平命中概率* 最小概率 = 0.0241 *# 最小水平命中概率* 衰减参数 = 10 *# 高斯衰减参数，控制衰减的速率* if 深度差 <= 12.5:  水平概率 = 最大概率 *# 在上下12.5米范围内，水平概率最大* elif 深度差 <= 32.5:  *# 使用高斯衰减模型来代替线性插值或指数衰减* 水平概率 = 最小概率 + (最大概率 - 最小概率) \* np.exp(-((深度差 - 12.5) \*\* 2) / (2 \* 衰减参数 \*\* 2))  else:  水平概率 = 最小概率 *# 超过32.5米，水平命中概率为最小值* return 水平概率  *# 使用距离公式计算垂直命中概率* def 计算垂直命中概率(爆炸深度, 配置):  深度差 = abs(爆炸深度 - 配置.中心深度) *# 计算投弹点与潜艇中心深度的差值  # 当与潜艇中心深度的距离在上下32.5米时，垂直命中概率为1，否则为0* if 深度差 <= 32.5:  return 1.0  else:  return 0.0  *# 计算平面内命中概率（基于二维正态分布积分）* def 计算平面内命中概率(配置):  def f\_xy(x, y, 水平误差):  return (1 / (2 \* np.pi \* 水平误差 \*\* 2)) \* np.exp(-(x \*\* 2 + y \*\* 2) / (2 \* 水平误差 \*\* 2))   命中概率, \_ = dblquad(f\_xy, -配置.宽度 / 2, 配置.宽度 / 2, lambda x: -配置.长度 / 2, lambda x: 配置.长度 / 2,  args=(配置.水平误差,))  return 命中概率 *# 截尾函数，深度误差分析* def 截尾正态分布(深度, 配置):  a, b = (配置.最小深度 - 配置.中心深度) / 配置.垂直误差, np.inf  return truncnorm.pdf(深度, a, b, loc=配置.中心深度, scale=配置.垂直误差)  *# 计算总命中概率* def 计算总命中概率(爆炸深度, 配置):  水平命中概率 = 计算水平命中概率(爆炸深度, 配置) *# 使用高斯衰减计算水平命中概率* 垂直命中概率 = 计算垂直命中概率(爆炸深度, 配置) *# 垂直命中概率* return 水平命中概率 \* 垂直命中概率  *# 多枚深弹命中概率模块* def 计算至少一次命中概率(爆炸深度, 行距, 列距, 配置):  偏移量 = [(-行距, -列距), (-行距, 0), (-行距, 列距),  (0, -列距), (0, 0), (0, 列距),  (行距, -列距), (行距, 0), (行距, 列距)]   命中概率列表 = [计算总命中概率(爆炸深度, 配置) for x\_offset, y\_offset in 偏移量]   未命中概率 = np.prod([1 - p for p in 命中概率列表])  return 1 - 未命中概率 *# 引爆深度和投弹阵列间距的改进和升级* def 优化命中概率(配置):  深度范围 = np.linspace(120, 200, 100) *# 引爆深度范围* 行列距范围 = np.linspace(30, 100, 10) *# 行间距和列间距的可能值* 最佳深度, 最佳行距, 最佳列距, 最佳概率 = None, None, None, -1   for 深度 in 深度范围:  for 行距 in 行列距范围:  for 列距 in 行列距范围:  命中概率 = 计算至少一次命中概率(深度, 行距, 列距, 配置)  if 命中概率 > 最佳概率:  最佳概率 = 命中概率  最佳深度 = 深度  最佳行距 = 行距  最佳列距 = 列距  return 最佳深度, 最佳行距, 最佳列距, 最佳概率 *# 布局投弹点可视化* def 可视化投弹点(行距, 列距, 配置):  x偏移 = [-行距, -行距, -行距, 0, 0, 0, 行距, 行距, 行距]  y偏移 = [-列距, 0, 列距, -列距, 0, 列距, -列距, 0, 列距]  fig, ax = plt.subplots()  ax.scatter(x偏移, y偏移, c='blue', marker='\*', s=200, label='投弹点')  ax.set\_xlim([-行距 - 配置.长度 / 2 - 20, 行距 + 配置.长度 / 2 + 20])  ax.set\_ylim([-列距 - 配置.宽度 / 2 - 20, 列距 + 配置.宽度 / 2 + 20])  ax.legend()  plt.title("多枚投弹落点平面布局示意图", fontsize=14)  plt.grid(True)  plt.show()  *# 执行优化* 最佳深度, 最佳行距, 最佳列距, 最佳概率 = 优化命中概率(config) print(  f"最佳引爆深度: {最佳深度:.2f} m, 最佳行距: {最佳行距:.2f} m, 最佳列距: {最佳列距:.2f} m, 最大命中概率: {最佳概率:.5f}") *# 最佳投弹点可视化布局* 可视化投弹点(最佳行距, 最佳列距, config) |