

多波束测深系统的条带覆盖宽度及重叠率的数值模拟与分析

摘 要

本文通过建立多波束测深的覆盖宽度及重叠率的数学模型，分析了海底坡度和测线方向等因素对测量结果的影响，并利用 Python 编程实现了模型的计算和仿真，为多波束测深系统提供了有效的测量方案。

针对问题一：在测线方向与海底坡面等深线相互平行的情况下，通过考虑海底坡度对多波束测深的覆盖宽度及相邻条带之间重叠率的影响，同时应用几何关系和三角函数推导出覆盖宽度和新的重叠率的表达式。最后，根据所列表达式变成计算出表 5-1 中所列位置的指标值并保存到 result1.xlsx 文件中。

针对问题二：我们发现问题一的情况为问题二特殊情况，故求出多波束探测器覆盖线与水平面的夹角 γ 以将问题二简化为问题一的情况进行计算。首先，我们利用向量和空间几何求解 γ ，得到了只与 α 、 β 有关的 γ 表达式。其次，求解测量船只相对于上一时刻中心点的距离 x 和距离测量水域中心点 B 的关系，由此可以得到 γ 和 β 、 x 和 β 的关系。最后，将这两个关系表达式代入问题一中，得到了覆盖宽度的数学模型，并使用 Python 计算不同 R 和 β 情况下的覆盖宽度并保存到 result2.xlsx 文件中。

针对问题三：通过查阅文献，我们考虑了沿南-北疏密分布、沿东-西平均分布、沿西南-东北方向疏密分布三种测线分布情况，分别计算了测线总长度及相邻条带之间的重叠率并结合查阅的文献进行排除，最终选择了沿南-北疏密平行分布的测线设计。之后我们使用穷举法，每次选择最接近 10%的情况作为局部最优解进行迭代，直到当前测线距离最左端的距离大于或等于之前计算出的末位置测线的距离时停止计算。最后，得到了测线分布最优解，以及相应的总长度 137048m。

针对问题四：我们首先对给出的数据使用 MATLAB 进行可视化处理，得到三维海底深度图，并进一步得到了海底地形图。其次，由上述问题的公式得到了覆盖宽度，为了求解题目进一步写了相应的目标函数。最后，我们利用粒子群算法进行求解，得到最佳函数。

关键词：覆盖宽度 穷举 几何求解 粒子群算法

一、问题重述

1.1 引言

本文的主要问题是探讨单波束和多波束测深的数学模型和方法，以及如何利用已有的单波束测深数据为多波束测量船的测量布线提供帮助。单波束测深是一种利用声波在水中的传播特性来测量水体深度的技术，它通过记录声波从发射到接收的传播时间，结合声波在海水中的传播速度，计算出海水的深度。多波束测深系统是在单波束测深的基础上发展起来的，它可以在与航迹垂直的平面内一次发射出数十个乃至上百个波束，从而提高了测量的效率和质量。本文需要解决以下几个具体的问题：

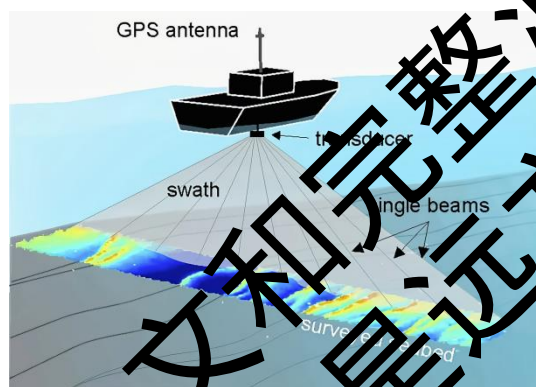


图 1-1 多波束测线示意图

1.2 问题的提出

问题一：建立多波束测深的覆盖宽度及相邻条带之间重叠率的数学模型，并根据给定的参数计算出不同海水深度、坡度下的覆盖宽度和重叠率。

问题二：考虑测线方向与海底坡面的法向在水平面上投影的夹角对覆盖宽度的影响，建立覆盖宽度的数学模型，并根据给定的参数计算出不同位置的覆盖宽度。

问题三：设计一组测量长度最短、可完全覆盖整个待测海域的测线，且相邻条带之间的重叠率满足 10%~20% 的要求。

问题四：利用已有的单波束测深数据为多波束测量船的测量布线提供帮助，设计出具体的测线，并计算出测线的总长度、漏测海区占总待测海域面积的百分比、以及重叠率超过 20% 部分的总长度。

二、问题分析

2.1 整体建模流程



图 2-1 建模流程图

2.2 问题一的分析

本题中给出了单波束测深和多波束测深的工作原理，以及多波束测深条带的覆盖宽度、测线间距和重叠率之间的关系。要求建立多波束测深的覆盖宽度及相邻条带之间重叠率的数学模型，并根据给定的参数计算出不同海水深度和坡度下的覆盖宽度和重叠率。

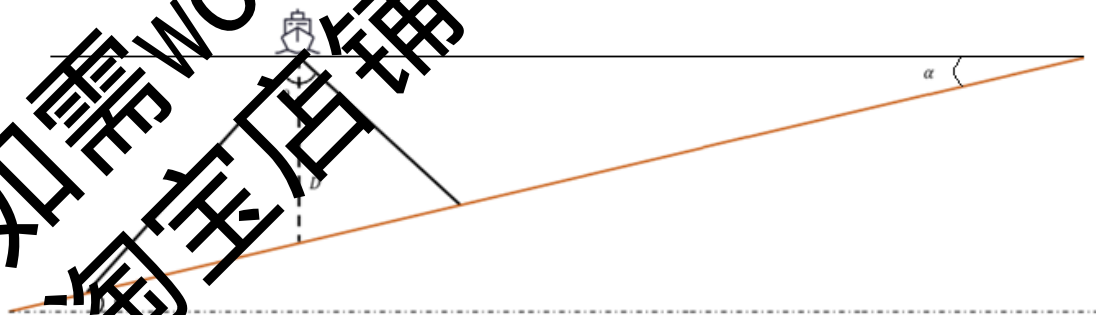


图 2-2 问题 1 示意图

利用题目中给出的信息和条件，我们对问题一进行如下分析：

首先根据问题中图 7 得出如图 2-2 所示示意图，对多波束测深系统的模型充分理解。然后针对海水深度，延长坡面和海平面，使二者交于一点，二者延长线的夹角等

于坡面的坡度 α 。再根据正切值的定义对海水宽度进行求解。

其次针对覆盖宽度，题目中并没有给出清晰的定义，我们定义多波束测线在坡面上的探测距离为覆盖宽度。然后根据角度的关系和正弦定理可以对覆盖宽度进行求解。

最后对于重叠率，题目中给出以下定义公式：

$$\eta = 1 - \frac{d}{W} \quad (1)$$

相邻两测线间的间距 d 为 200m，带入刚求解出的覆盖宽度 W 即可对重叠率进行求解。

根据以上分析，我们可以建立多波束测深的覆盖宽度及相邻条带之间重叠率的数学模型，并根据给定的参数计算出不同海水深度和坡度下的覆盖宽度和重叠率。

2.3 问题二的分析

问题二可以看作是问题一的一个拓展，将原本在平面上的二维问题升级到了立体空间中。实际上，问题一可以视作问题二中特殊情况的一部分，即当问题二中的角度 $\beta=90^\circ$ 和 $\beta=270^\circ$ 时。在解决问题二时，可截取与测线方向垂直的平面，进而将问题二转化为问题一的情况。而我们的挑战在于确定 β 和波束探测系统与水平平面之间的夹角关系，以便准确测量多波束覆盖线在水平平面上的角度。

为了探究多波束探测器覆盖线和水平平面的真实夹角，我们首先考虑了四种特殊情况，即 $\beta=0^\circ$ ， $\beta=90^\circ$ ， $\beta=180^\circ$ 和 $\beta=270^\circ$ 。当 $\beta=90^\circ$ 或 $\beta=270^\circ$ ，多波束探测器的覆盖线与水平平面的夹角为 α ，并且覆盖范围与测量船距海域中心点的距离无关，保持不变。因此，我们可以将测量船距海域中心点的距离代入问题一的公式中，从而得到覆盖范围。

当 $\beta=0^\circ$ 或当 $\beta=180^\circ$ 时，测量船只的多波探测器覆盖线与水平面夹角为0，因此在定点情况下可将海底地形视为平坦进行计算。首先，我们计算了由于测量船距海域中心点处的距离的改变而产生变化的海水深度。其次，基于问题一计算出当 $\alpha=0^\circ$ 时相应的多波束测深的覆盖宽度。

以上四种特殊情况可以用来验证普通情况下求解出的数值是否正确。首先，根据问题一以及问题背景，我们易得多波束测深的覆盖宽度的变化具有一定的规律性和连续性。其次，我们利用 β 、 γ 以及向量和几何关系相关知识探究覆盖宽度的变化的规律

性和连续性。

当 β 不变时,坡面与水平面的夹角 γ 也不变化,此时需要对测量船只相对于上一时刻中心点的距离 x 和距离测量水域中心点 R 的关系。

当求得 β 和 α 的关系, x 和 R 的关系,带入第一问所求得公式,便可得到 β 和 R 相对于覆盖宽度 W 的关系。

根据以上分析,我们可以建立多波探测器覆盖线与水平面夹角的数学模型,并用以建立多波束测深的覆盖宽度的数学模型。最后,根据给定的参数计算出不同海水深度和坡度下的覆盖宽度。

2.4 问题三的分析

本问题让我们在一个矩形海域内设计一组测线,要求测线总长度最短,并且能够完全覆盖整个海域,且满足相邻测带之间的重叠率在10%~20%之间,是一个单目标规划问题。为了满足这些要求,我们首先需要分析不同测线方向对测线间距和重叠率的影响,进而选出最优的测线方向。

我们通过查阅文献和相关知识得知,测线要求平行布置,并且在区域内所有的测线应尽量保持平行关系,且它们的走向也应与航道底部的等深线走向尽量一致,以最大程度地扩大测线的覆盖范围,提高测量的有效性。因此测线的方向不能随意设定,我们大致将测线方向分为三种情况:沿南-北疏密分布、沿东-西疏密分布、沿西南-东北方向疏密分布,并对各个方向进行具体分析和讨论。

对于沿南-北疏密分布的情况,我们首先计算出距离海域西边界和东边界的最优测线距离,方便后续求解。由于该分布情况下测量船在每组测线上覆盖范围固定不变,且覆盖率越低测量间距越大,总测线长度越小,因此我们选择穷举法进行求解。通过穷举距离当前测线位置的所有位置的重叠率,选择重叠率向下最接近10%的最为局部最优解,进而找出全局最优解。

对于沿东-西平均分布的情况,在该分布情况下,测量船在同一测线的不同位置上覆盖宽度不同。假如航向是由西向东进行,在该航线上,探测器覆盖宽度随船只前行而逐渐变窄。但在不同航线相同位置,测量船测量宽度相同。由于测量宽度由西向东逐渐变窄,为了避免在浅水处发生漏测,我们需要限制测量船在东边界探测器恰好覆盖整个海底坡面。

对于沿西南-东北方向疏密分布的情况，该情况属于上述两种特殊情况的一般情况，我们使用单目标规划进行求解。通过限制测线倾斜角度、测线之间间距、重叠率在 10%~20%之间等条件，不断最小化测线总长度进行求解。

2.5 问题四的分析

首先，我们需要对附件数据进行可视化分析，以便于对若干年前该海域的地形情况有一个大致的了解。根据多波束测深的原理和给定的多波束换能器的开角，我们可以计算出在不同深度下的覆盖宽度。覆盖宽度的计算公式可以参考问题二中给出的定义。在这个过程中，我们需要进行以下步骤：

1. 设计测线在海平面上的分布，以确保能够覆盖整个海域。
2. 根据设计的测线，计算漏测面积和重叠面积。
3. 调整测线设计，以最小化总测线长度和漏测面积，同时控制重叠率。

对于初步测线来说，我们可以计算整个海域的平均覆盖宽度，并根据平均覆盖宽度和允许的最大重叠率（20%）来确定相邻测线之间的间距。

为了优化测线设计，我们可以采用多目标规划的方法，确定以下目标函数：

- (1) 测线总长度 L 最小
- (2) 重叠率不超过 20%
- (3) 漏测海区面积 M 最小
- (4) T 应为 0 或最接近 0

然后，我们可以通过 PSO 算法，即粒子群算法，来求解这个多目标规划问题。但是，由于经典粒子群算法局限于它的粒子群的变动范围只能落在一维坐标上，且经典粒子群算法中没有对自变量的限制性条件。因此，我们引用二维粒子群优化算法，并且新增惩罚函数，使其能够解决在二维坐标系里的非线性规划问题。

三、模型假设

1. 假设多波束测深系统的参数（如发射角度、发射频率、接收阵列等）在测量过程中保持不变。
2. 假设海底坡度和海水深度在测量区域内是连续变化的，且没有突变或异常点。
3. 假设测量船只的航速和航向在每一条测线上是恒定的，且每一条测线之间的转向时间可以忽略不计。
4. 假设测量船只和多波束探测器之间的相对位置和姿态是固定的，且不受海流、风浪等因素的影响。
5. 假设测线间的间距只能为整数，为了求解方便，我们将测量区域划分为等宽的矩形，并且每一条测线与邻近的测线平行。

如需word论文和完整源代码
淘宝店铺：亦是远方

四、符号说明

符号	说明	单位
x	测线距中心点处的距离	m
D	海水深度	m
W	覆盖宽度	m
η	与前一条测线的重叠率	%
α	坡度角	°
θ	换能器开角	°
d	相邻两测线间的间距	m
β	测线方向与海底坡面的法向在水平面上投影的夹角	°
γ	多波探测器覆盖线与水平平面夹角	°
δ	测线与多波探测器覆盖线的夹角	°
h_{max}	海域最深处深度	m
A	延长坡面与海平面交点	——
l_1	海域宽度	m
x_0	南-北方向测线始发处距离海域最西侧距离	m
x_1	延长坡面与海平交点距离海域中心点处距离	m
x_i	南-北方向测线结束点距离海域最东侧距离	m
W_i	东-西方向测在海域最深处的覆盖宽度	m
Q	重叠区域宽度	m
T_{sum}	重叠率超过 20%的总长度	m
$p_{dtext,gbest}^k$	群体在第 k 次迭代中第 d 维的历史最优位置	——

五、模型的建立与求解

5.1 问题一模型的建立与求解

5.1.1 模型的建立

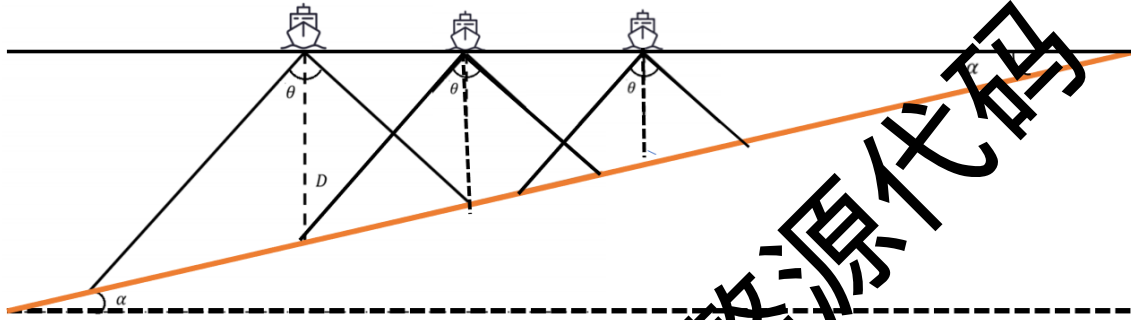


图 5-1 问题 1 优化示意图

由题可得，每隔 200m 会进行一次多波束测线，当延长坡面和海平面，使二者交于一点，二者延长线的夹角等于坡面的坡度 α 。首先计算测线距离中心点处为零的船只，该船只的 $D_0 = 70m$ ， $\theta = 120^\circ$ ， $\alpha = 11.5^\circ$ ，根据三角形的正切函数，可例如下公式：

$$\tan \alpha = \frac{D}{x_0} \quad (2)$$

我们定义坡面延长线和海平面的交点由以上可求得在测线距离中心点处为零的船只距离理想岸边的距离为 x_0 ，测海水深度可由以下式子求出：

$$D = (x_0 - x) \tan \alpha \quad (3)$$

由正弦定理可抽象出一个覆盖宽度 W 和垂直深度 D 的函数关系：

$$W = D \left\{ \frac{\sin \frac{\theta}{2}}{\sin(\frac{\pi}{2} + \alpha - \frac{\theta}{2})} + \frac{\sin \frac{\theta}{2}}{\sin(\frac{\pi}{2} - \alpha - \frac{\theta}{2})} \right\} \quad (4)$$

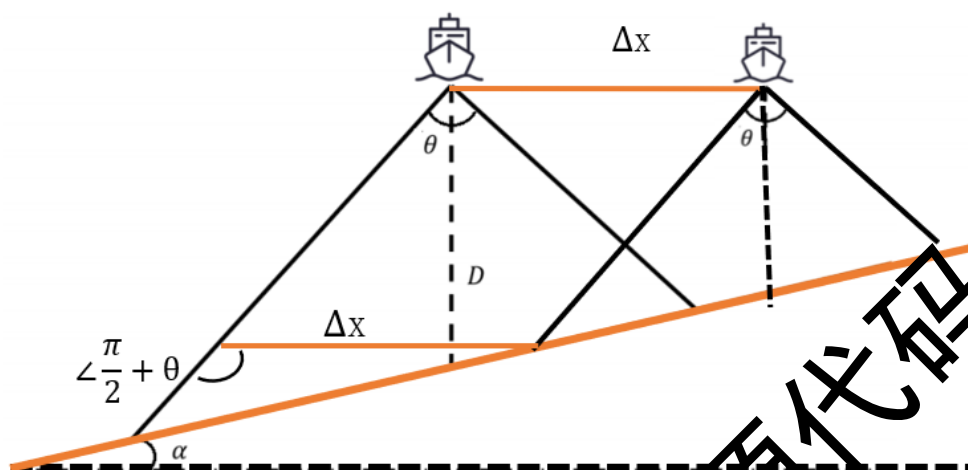


图 5-2 斜平面时覆盖宽度示意图

由公式(1)中在海底平面平行时的覆盖宽度可以得出如图 5-2 所示在海底平面有一定角度下的公式，由正弦定理得：

$$\frac{\Delta x}{\sin(\frac{\pi}{2} - \alpha - \theta)} = \frac{d}{\sin(\frac{\pi}{2} - \theta)} \quad (5)$$

结合公式(1)给出的覆盖宽度下的定义，得到在当前海底平面为斜面时的定义，代入公式(5)得到在海底平面为坡面时的重叠率计算公式。

$$\eta = 1 - \frac{\Delta x \sin(\frac{\pi}{2} + \frac{\theta}{2})}{W \sin(\frac{\pi}{2} - \alpha - \frac{\theta}{2})} \quad (6)$$

5.1.2 模型的求解

通过使用 Python 进行求解，我们引用了 numpy 和 pandas 库编写代码，具体代码见附录一，解得：

表 5-1 result1 结果表

测线距中心点 处的距离/m	-800	-600	-400	-200	0	200	400	600	800
海水深度/m	90.95	85.71	80.47	75.24	70.00	64.76	59.53	54.29	49.05
覆盖宽度/m	315.81	297.63	279.44	261.26	243.07	224.88	206.70	188.51	170.33
与前一条测线 的重叠率/%	——	29.59	25.00	19.78	13.78	6.37	-1.39	-11.17	-23.04

为了方便展示，这里进保留最后两位数字，具体内容在附件 result1.xlsx 中。

通过 Python 求解出以上数据，可得当测线距中心距离为 600m 或 800m 时，其与前一条测线的重叠率为负，即在水浅处靠近理想岸边的地方出现了漏测的情况，影响了测量质量。

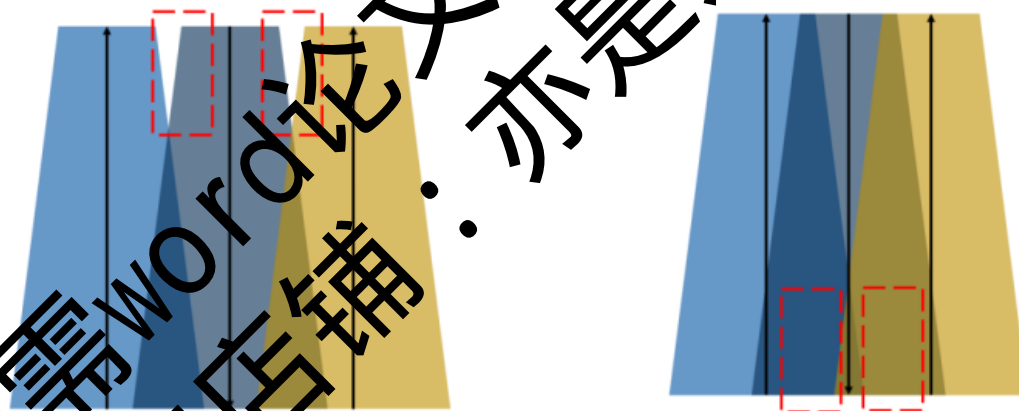


图 5-3 漏测和重叠冗余的情况

5.2 问题1模型的建立与求解

5.2.1 模型的建立

因为 result2.xlsx 中的距离单位为海里，而 1 海里=1852 米，由于与计算过程中单位不统一，需要进行单位转换。

当 $\beta=90^\circ$ 或当 $\beta=270^\circ$ ，测量船只多波探测器与水平面夹角为 α ，且覆盖范围与测量

船在测线上距海域中心点处的距离无关，保持不变。可以带入问题一中的公式进行求解。即对于公式（4）来说，其中 D 恒等于一常数，可列：

$$D = x_0 * \tan\alpha \quad (7)$$

此时 $\alpha=1.5^\circ$ ，与 D_0 同时代入公式（4）可解。

当 $\beta=0^\circ$ 或当 $\beta=180^\circ$ 时，测量船只多波探测器覆盖宽度所成的直线与水平面平行，且覆盖范围 W 只与距离水域中心的距离有关。

当 $\beta=0^\circ$ 时，测量船只远离岸边，此时船只所在位置处的海域深度 D_i 逐渐增加，可由以下公式得出：

$$D_i = D_0 + x * \tan\alpha \quad (8)$$

由几何关系可得此时在如下公式中的 α_1 和 α_2 都为 30° ，且由几何关系可得其现在的覆盖宽度 W 为：

$$W = 2(D_0 + x * \tan\alpha) \tan \frac{\theta}{2} \quad (9)$$

当 $\beta=180^\circ$ 时，测量船只远离岸边，此时船只所在位置处的海域深度 D_i 逐渐减小，可由以下公式得出：

$$D_i = D_0 - x * \tan\alpha \quad (10)$$

由几何关系可得此时在如下公式中的 α_1 和 α_2 都为 30° ，且由几何关系可得其现在的覆盖宽度 W 为：

$$W = 2(D_0 - x * \tan\alpha) \tan \frac{\theta}{2} \quad (11)$$

之后，我们考虑 β 角不是特殊角度的一般情况。我们建立出如下图所示的几何图形：

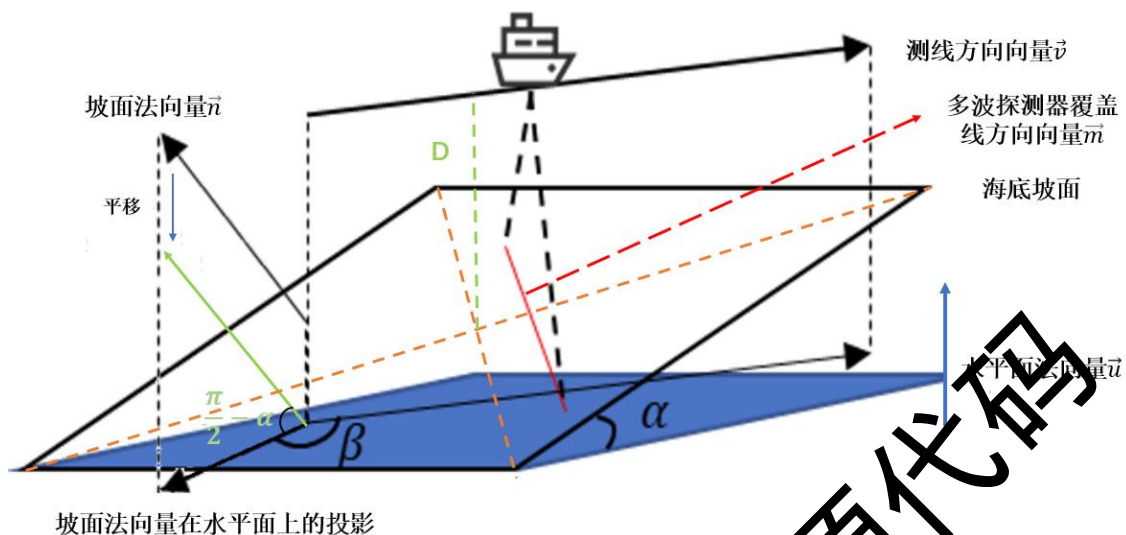


图 5-4 平面法向量与 β 和 α 的关系

其中，各个向量所表示含义均在图 5-4 中标注。由平面几何图形之间的关系可得如下公式：

$$\cos \alpha = \frac{\vec{n} \cdot \vec{v}}{|\vec{n}| |\vec{v}|} \quad (12)$$

$$\cos \beta = \frac{\vec{p} \cdot \vec{v}}{|\vec{p}| |\vec{v}|} \quad (13)$$

其中， \vec{p} 为坡面法向量在水平面上的投影。由投影的计算公式可得：

$$\vec{p} = \frac{\vec{n}}{|\vec{n}|} \cos \left(\frac{\pi}{2} - \alpha \right) \quad (14)$$

将其带入到公式(13)中可得

$$\cos \beta = \frac{\vec{n} \cdot \vec{v}}{|\vec{n}|^2 |\vec{v}|} \quad (15)$$

我们记多波探测器覆盖线与水平平面夹角为 γ ，则有：

$$\cos \gamma = \frac{\vec{u} \cdot \vec{m}}{|\vec{u}| |\vec{m}|} \quad (16)$$

由于多波探测器覆盖线与测线方向垂直，可得：

$$\cos \delta = \frac{\vec{v} \cdot \vec{m}}{|\vec{v}| |\vec{m}|} = \cos 90^\circ = 0 \quad (17)$$

用三角恒等式，我们可以得到：

$$\sin \beta = \sqrt{1 - \cos^2 \beta} = \sqrt{1 - \left(\frac{\vec{p} \cdot \vec{v}}{|\vec{p}||\vec{v}|}\right)^2} \quad (18)$$

将上述公式代入 $\cos \gamma$ 的表达式中，最终可以得到：

$$\gamma = \sin \beta \times \alpha \quad (19)$$

至此求完 γ 与 β 的关系，下面将求解测量船只相对于上一时刻中心点的距离 x 和距离测量水域中心点 R 的关系。

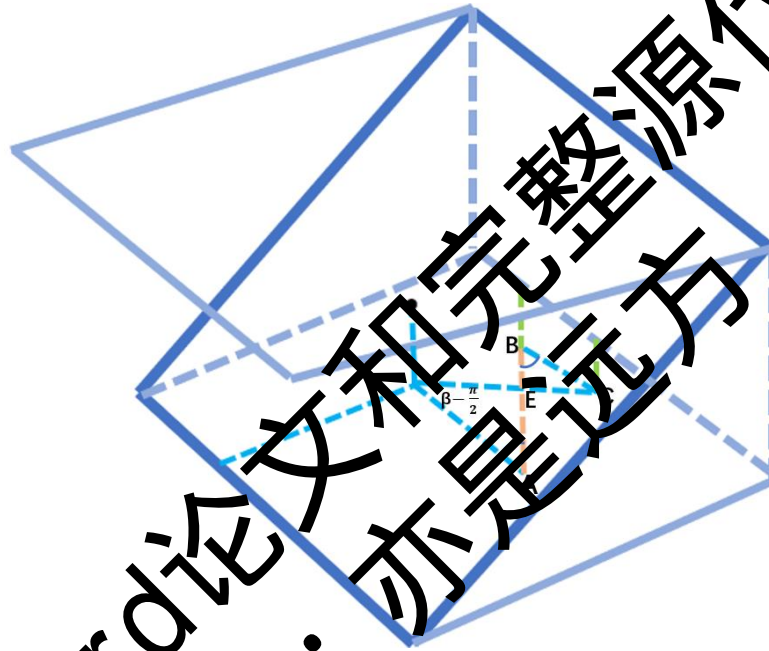


图 2-5 水平面上投影的角度关系

如图 2-5 所示，此时假设 β 不变，将海平面上因距离海域中心点 R 距离不同引起的测量船只相对靠近岸边的距离为 x ，即线段 BE。远离海域中心点的过程中，其相对移动的距离为 Δx ，即线段 EC，由角度关系可得：

$$\angle EBC = \pi - \beta$$

故在直角三角形 BEC 中， x 满足以下公式：

$$x = \frac{\Delta R}{\tan(\pi - \beta)} \quad (20)$$

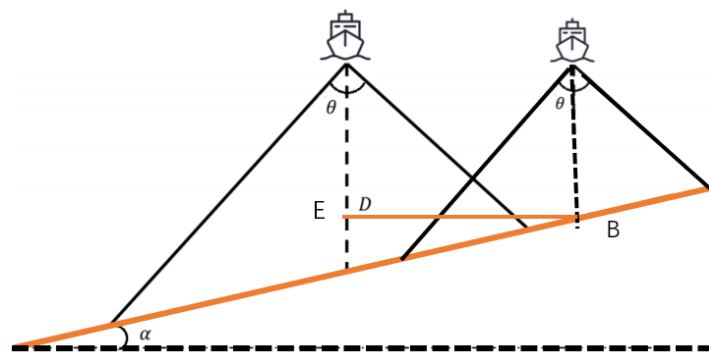


图 5-6 沿测线方向的相对平移的测量船只示意图

此时由公式(19)与公式(20)可以求得 γ 和 β 、 x 和 β 的关系，代入公式(3)和公式(4)可得：

$$W = (D - \frac{R \tan(\alpha_0 \sin \beta)}{\tan(\pi - \beta)}) \left\{ \frac{\sin \frac{\theta}{2}}{\sin(\frac{\pi}{2} + \alpha_0 \sin \beta - \frac{\theta}{2})} + \frac{\sin \frac{\theta}{2}}{\sin(\frac{\pi}{2} - \alpha_0 \sin \beta - \frac{\theta}{2})} \right\} \quad (21)$$

此式便是覆盖宽度 W 与测线方向夹角 θ 与测量船距海域中心点处的距离 R 的关系。

5.2.2 模型的求解

通过使用 Python 进行求解，我们引用了 numpy 和 pandas 库编写相关代码，具体代码见附录三，解得如下表格：

表 5-2 result2 结果表

覆盖宽度/m		测量船距海域中心点处的距离/海里							
		0	0.3	0.6	0.9	1.2	1.5	1.8	2.1
测线方向夹角	0	415.69	466.09	516.49	566.89	617.29	667.69	718.09	768.48
	45	416.19	451.87	487.54	523.22	558.90	594.57	630.25	665.92
	90	416.69	416.69	416.69	416.69	416.69	416.69	416.69	416.69
	135	416.19	380.52	344.84	309.16	273.49	237.81	202.13	166.46

/°	180	415.69	365.30	314.92	264.53	214.14	163.76	113.37	62.98
	225	416.19	380.52	344.84	309.16	273.49	237.81	202.13	166.46
	270	416.69	416.69	416.69	416.69	416.69	416.69	416.69	416.69
	315	416.19	451.87	487.54	523.22	558.90	594.57	630.25	665.92

为了方便展示，这里进行保留两位小数，具体文件在附件 result2.xlsx 中。

5.3 问题三模型的建立与求解

5.3.1 模型的建立

为了简化测量模型，将测线方向规定为沿南-北疏密分布、沿东-西平均分布、沿西南-东北方向疏密分布三种情况。

首先对于测线方向沿南-北分布时，有以下情况进行分析：

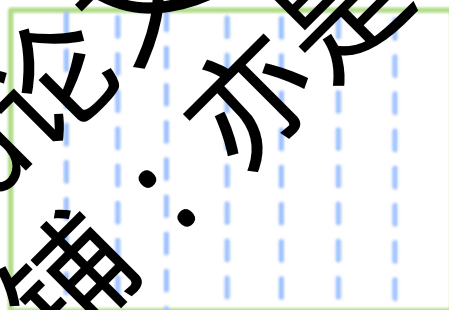


图 5-7 南北方向测线设计示意图

为了计算在沿南北方向测量时的测线间隔，我们需要对在最深处的西边方向和最东处的东边方向进行模型的建立。从最西侧方向覆盖宽度 W 正好与待测海域东侧边缘相接情况下开始移动，如果在沿测线移动的过程中，其覆盖宽度 W 正好与待测海域西侧边缘相接，此时在满足重叠率的要求之下，可完全覆盖整个待测海域。

首先计算在最深处的始发点：

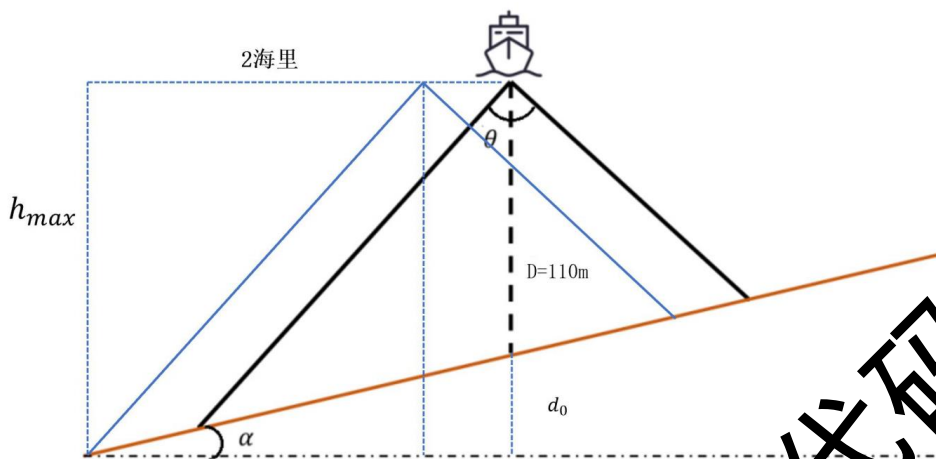


图 5-8 海域最深处示意图

已知海域中心点处的海水深度为 110m，坡度为 $1/54$ ，南北长 2 海里、东西宽 4 海里，设海域宽为 l_1 由

$$\frac{D}{l_0} = \tan \alpha = \frac{h_{max}}{\frac{l_1}{2} + l_0} \quad (22)$$

得到 $h_{max} = \tan \alpha \left(\frac{l_1}{2} + \frac{D}{\tan \alpha} \right) = 205.993\text{m}$ 。

又因为换能器开角 $\theta = 120^\circ$ 故可得始发处距离海域最西侧距离为

$$x_0 = h_{max} \tan \frac{\theta}{2} \quad (23)$$

得到 $x_0 \approx 355.522\text{m}$

同理可得在海域最浅处的测量位置，如图 5-9 所示，延长坡面与海平面相交与点 A。

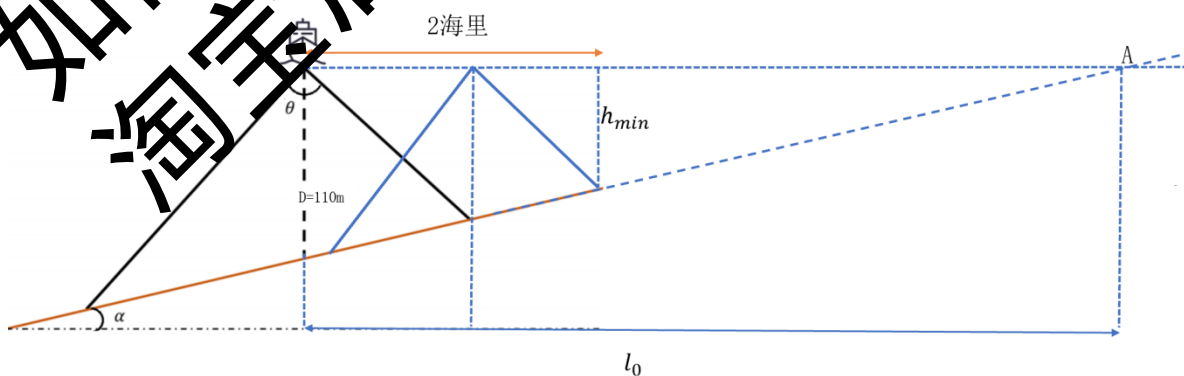


图 5-9 海域最浅处示意图

$$\frac{D}{l_0} = \tan\alpha = \frac{h_{min}}{l_0 - \frac{l_1}{2}} \quad (24)$$

得到 $h_{min} = \tan\alpha(\frac{D}{\tan\alpha} - \frac{l_1}{2}) \approx 13.007\text{m}$ 。

可得结束点距离海域最东侧距离为

$$x_i = h_{min} \tan \frac{\theta}{2} \quad (25)$$

得到 $x_i \approx 22.529\text{m}$

以上计算的两条测线为海域最西侧和最东侧的测线，且其覆盖宽度 W 都正好与待测海域边缘相接。

由此可以得到最西侧测线的覆盖宽度。

为了得到局部最优解，我们使用贪心思想进行由西侧测线到东侧测线疏密的设计，要求每次设计都可以取得局部最优解，易得每次的重叠率都要取 10%，容易发现其测线应如图 5-10 变化：

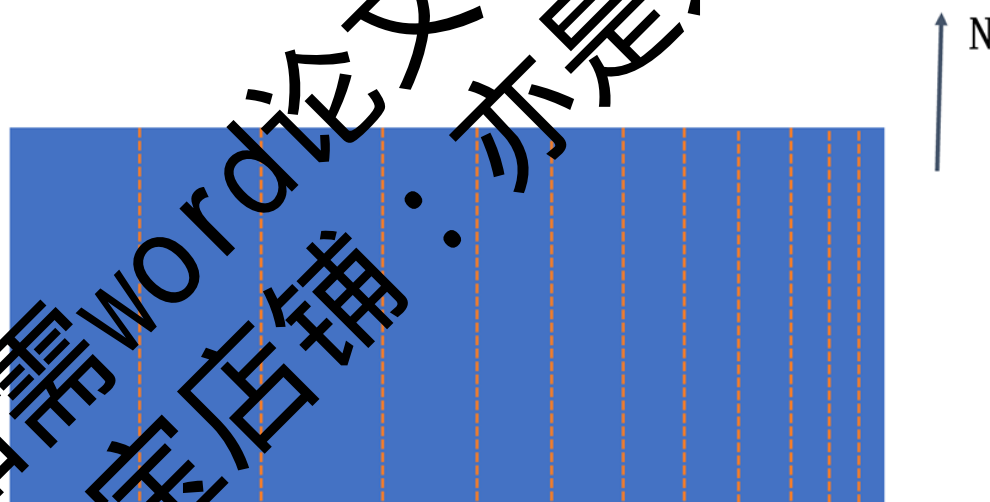


图 5-10 测线沿南北方向设计示意图

然后考虑测线方向沿东-西方向分布时，测线与重叠宽度的分布情况如图 5-11 所示：

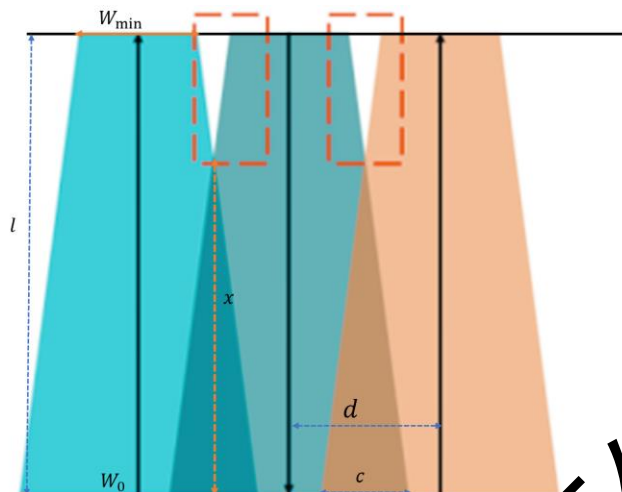


图 5-11 东-西方向俯视图

这种情况即为问题二中 $\beta = 180^\circ$ 的情况，它的计算公式可表示为

$$W = 2(D_0 - x * \tan \alpha) \tan \frac{\theta}{2} \quad (26)$$

由第一种情况可以知道海域最深处的深度为到 $h_{max} = \tan \alpha (\frac{l_1}{2} + \frac{D}{\tan \alpha}) \approx 206.993\text{m}$ ，此时的

$$W_0 = 2(h_{max} - 0 * \tan \alpha) \tan \frac{\theta}{2} = 2h_{max} \tan \frac{\theta}{2} \quad (27)$$

故当东西方向的东边测线第一条距离岸边 $\frac{W_0}{2}$ ，由于东西方向所有测线覆盖宽度都是一样的，所以西边测线第一条距离岸边也为 $\frac{W_0}{2}$ 。而且测线的坡角都为 0° ，且他们互相平行，故当测量长度最短时，他们的间隔应尽量大，且重叠率尽量低，故测线间的间隔 d 都一样。

$$d = \frac{l - W_0}{n} \quad (28)$$

其中 l 为东西方向长度， n 为测线总条数。

测线总长度为

$$l_2 = nl \quad (29)$$

故若确定了 n 就能确定测线总长度，而 n 需要由 d 来确定。下面我们来确定测线间的间隔 d 。

首先,我们由重叠率要求去求 d ,由于多波束测深的覆盖宽度的条带为等腰梯形,故一个条带的面积为

$$S = \left(2h_{\min}\tan\frac{\theta}{2} + 2h_{\max}\tan\frac{\theta}{2}\right)l \div 2 = \tan\frac{\theta}{2}(h_{\min} + h_{\max}) \quad (30)$$

由于重叠率的要求为 10%~20%,故重叠面积为 $(10\%S, 20\%S)$,由图示可得,重叠面积

$$S_1 = c * x/2 \quad (31)$$

x 东西方向侧线间距漏测初点西船行驶的距离,由此可得

$$l = (n - 1)c \quad (32)$$

故

$$d = \frac{(n - 1)c - W_0}{2} \quad (33)$$

其次,我们计算漏测面积,设东西方向侧线间距漏测初点为 B,则在 B 点处:

$$W_1 = d$$

联合公式(9),可以得到船行驶的距离 x ,从而得到船距离海岸的距离 $l_1 = l - x$,且,所有侧线均为这一距离

故由图 5-11 可得,两侧线间的漏测面积为

$$S_2 = \frac{(d - W_{\min})l_1}{2} \quad (34)$$

漏测面积为

$$S_3 = (n - 1)S_2 + (W_0 - W_{\min})l/2 \quad (35)$$

由图分析可知当前情况必然存在漏测面积,故将所有测线整体向北移动 $W_0 - W_{\min}$ 个单位长度,并在海域最南处加一条距离海域边缘 $W_0 - W_{\min}$ 个单位长度,重复上述计算,可得出移动后测线间隔公式为:

$$d = \frac{l - 2(W_0 - W_{\min})}{n} \quad (36)$$

同理,我们使用穷举法进行求解,不断枚举测线间隔 n 的值,最后选择覆盖率在

10% ~ 20%之间且间隔最大的情况。

最后考虑测线方向沿西南-东北方向，此时属于以上两种特殊情况的一般情况，如图 5-12 所示。



图 5-12 测线沿西南-东北方向设计示意图

对于此种情况，我们可以使用单目标规划来求解，该问题需要满足以下条件：

设计一组测量长度最短的测线，以覆盖整个待测海域，并且满足相邻条带之间的重叠率在 10%~20%之间。

对于此种情况，我们可以使用多目标规划来求解。首先，我们需要将问题分解成一系列的单目标规划问题。具体来说，我们可以将问题分解成以下两子问题：

1. 设计一组测量长度最短的测线，以覆盖整个待测海域。
2. 设计一组测量长度最短的测线，以满足相邻条带之间的重叠率在 10%~20%之间。

由上题可知，海底平面方程为 $f_{\text{海底}} = f(\alpha, \beta, h)$ ，测线的覆盖方程 $W_i = f(\alpha, \beta, h)$ ，且令西南到东北正方向测线分别为 l_i ，其中 $i = 1, 2, 3, 4, \dots$ ， l_i 与 l_{i+1} 的距离为 $d_{i,i+1} = D(l_i, l_{i+1})$ ，由此可得重叠率的定义 $\eta_i = 1 - \frac{d_{i,i+1}}{W_i}$ ，其中 $i = 1, 2, 3, \dots, n-1$ ，且 $10\% \leq \eta_i \leq 20\%$ ，测线长度为 S_i ，且 $S_{\text{总}} = \sum_{i=1}^n S_i$ ，综上可得：

目标规划 $\min S_{\text{总}} = \sum_{i=1}^n S_i$

$$\begin{cases} f_{\text{海底}} = f(\alpha, \beta, h) \\ W_i = f(\alpha, \beta, h) \\ 10\% \leq \eta_i \leq 20\% \end{cases} \quad (37)$$

若要进行测线设计，首先要确定测线的方向，其次要确定相邻两条测线的间距 d 。

测线方向的确定:

由以上情况,我们可以大概考虑三种测线方向,分别是西南-东北方向的倾斜测线,东-西方向的水平测线,南-北方向的水平测线,如下图所示(图 5-13)。



图 5-13 测线方向设计示意图

因要选择一种测线方向设计进行建模求解,首先要确定测线方向相关要求,故我们搜集相关文献,得到:

测线要求平行布设。在区域内,所有的测线应尽量保持平行关系,并且它们的走向也应与航道底部的等深线走向尽量一致,这样的布置可以最大程度地扩大测线的覆盖范围,提高测量的有效性。

总结得:

要求一:测线要求平行布设。

要求二:尽量与航道底部的等深线走向一致。由于我们需要考虑西深东浅的海域的测线,而东-西方向的水平测线与航道底部的等深线走向相互垂直,因此可大致先排除东-西方向的水平测线。

最后我们拟定南-北方向的水平测线为最优分布情况。

相邻两条测线间距的预确定:

要确定测线间距,首先要知道测线间距的要求,由题意可得:

要求一:各个测线测量范围应有重叠。由于位于测深系统中心下部的某一范围,信号表现清晰且探测效果卓越;相反,位于边缘的反射波由于强烈的散射,许多反射波无法有效地被接收器接收,故有效波较少,信号质量受到较大干扰。因此,我们需要确保相邻测线的探测范围存在一定的重叠区域,以确保每个区域的信号清晰和可靠,从而保障测量的方便性和数据的完整性。具体来说,我们需要确保相邻条带之间的重叠率在 10%到 20%之间。

要求二：漏测应尽量少。由于我们上述测线方向确定为南-北方向的水平测线，即我们为平行于等深线方向，当每条测线在起始点没有出现漏测现象，就不会出现由于坡度变化而产生的漏测现象。

由于多波束测深的覆盖宽度在相同条件下：

条件一：坡度都为 1.5° ；

条件二：多波束换能器的开角为 120° ；

会随着深度的增加而增加，故在海域深处测线间的间距应该较大，在海域浅处测线间的间距应该较小。并且测线间距要保证相邻条带之间的重叠率满足 $10\%\sim 20\%$ 。

最终，得到两条测线间距应为：由西向东逐渐减小。

即，测线设计方案为由西向东测线间距逐渐减小的南-北向水平测线。

5.3.2 模型的求解

不失一般性，我们先对东-西方向的水平测线分布的情况进行求解。我们根据公式（36），通过 Python 枚举出测线条数 n 在 2~300 之间的所有情况，并计算出各情况最大重叠率和最小重叠率，同时将结果保存在 result3_1.xlsx 表格，其中部分数据如下表：

表 5- result3_1 结果表

测量带条数	间隔距离	最小覆盖率	最大覆盖率
8	609.8235383	-11.53395882	0.149530725
9	522.70589	-10.60053613	0.271026335
10	457.8676337	-9.150469116	0.362148044
...
91	41.11169921	0.087598282	0.942664993
92	40.65490256	0.097736079	0.943302048
93	40.20814538	0.107651067	0.943925103
94	39.77110033	0.117350512	0.944534612
95	39.34345409	0.126841366	0.945131014
96	38.9249067	0.136130288	0.945714727

从上表可以看出，所有情况均不能同时满足最大覆盖率与最小覆盖率在 $10\%\sim 20\%$ 之间，故该测线分布情况不符合要求。

我们最终选择沿南-北方向进行密集测线的布置，这些测线是互相平行的，并且测量船在每次测量过程中测量宽度保持不变，因此在满足题目重叠率的要求下，不会出现漏测的情况。根据题目要求，相邻测线条带之间的重叠率应在 10%到 20%之间，相邻测线条带之间重叠率越小，测线间隔越大，所需总测量次数最少，测量距离最短。因此，局部最优布线方案即全局最优的方案，即每次选择重叠率最接近 10%的情况。

本问题中，只需满足两条测带重叠率最近接 10%即为最短测量距离，故我们选择使用穷举法。穷举法的基本思想是列举出所有可能的情况，逐个判断有哪些是符合问题所要求的条件，从而得到问题的全部解答，进而选择出最优解的情况，其流程图如下图 5-14 所示。

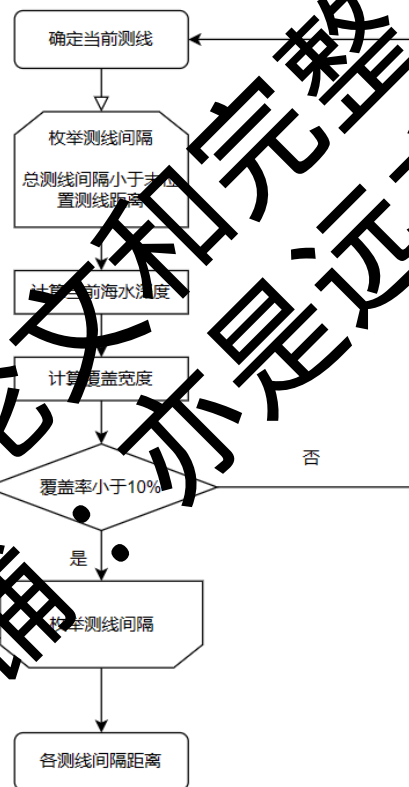


图 5-14 问题三的求解流程图

我们根据已建立的模型确定了起始点的位置和方向，从起始测线开始进行穷举。由于我们假设测线之间间隔只能为整数，故选择 1 米为步长，每次计算出距离当前测线各个位置的覆盖率，选择最接近 10%的情况作为局部最优解。然后，进行迭代，以新计算出的位置作为当前测线的起点，继续计算下一个测线的位置。

最后，达到终止条件，即当当前测线距离最左端的距离大于或等于之前计算出的

末位置测线的距离时，停止计算，得出所有测线之间的距离和位置。这样的穷举方法可以帮助我们找到满足条件的所有最优布线方案，确保了覆盖率的要求和测量的准确性。

通过使用 Python 编写相关代码求解，我们得到各测线处海水深度、距离上一条测线距离以及相邻两条测带之间的覆盖率表格（表 5-4），由于数据较多，此处只选用部分数据展示。

表 5-4 result3_2 结果表

序号	海水深度	距上一条测线距离	覆盖率
1	206.9926535	359	
2	183.3069168	546	0.101150229
3	170.0306545	507	0.100183289
4	157.7232714	470	0.100730442
5	146.3062096	436	0.100715616
6	135.7270973	404	0.101768822
7	125.9073707	356	0.10121997
8	116.794676	348	0.100855443
9	108.3366233	323	0.10029414
10	100.5070328	299	0.102265204

由完整的表格可知，测线共计 17 条，每条长度为 2 海里，即 3704 米，测线总长度为 62968 米。通过使用 Python 绘制测线分布如下图：

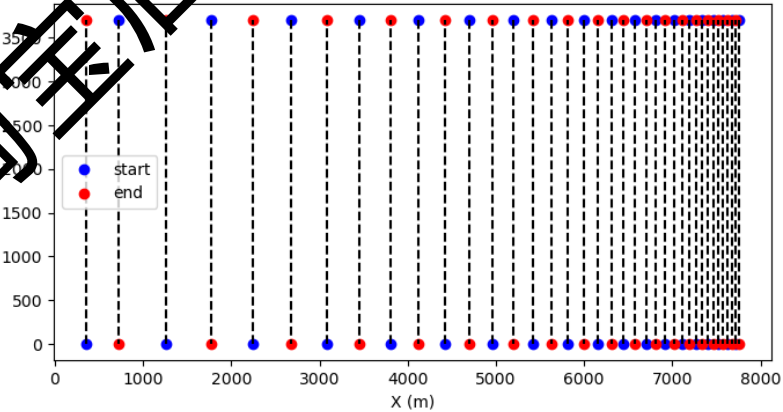


图 5-15 测线最优解分布图

从图 5-15 中可以看出，测线在水深较深处分布稀疏，在水深较浅处分布较为密集，符合我们的假设。

5.4 问题三模型的建立与求解

5.4.1 模型的可视化处理

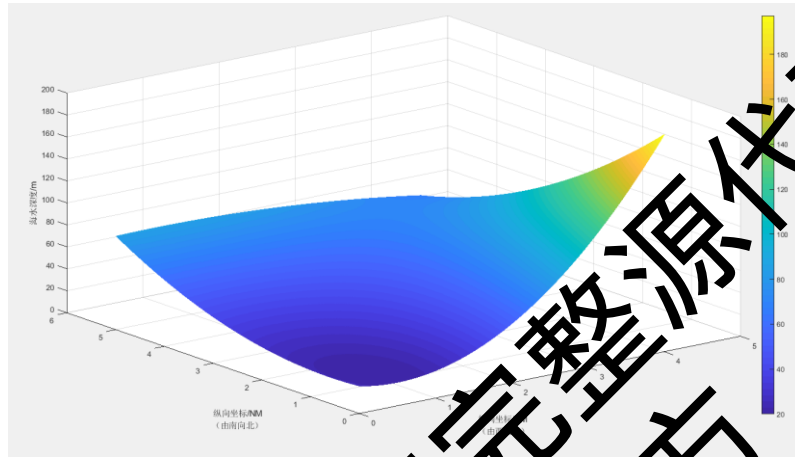


图 5-16 海水深度可视化示意图

由题目中的附件可以对海水深度在此海域内进行可视化分析，由于原来的数据是一个 251×201 的矩阵，在 MATLAB 中对其进行自动插值，可以得到如上图 5-16 较为平滑的可视化示意图。

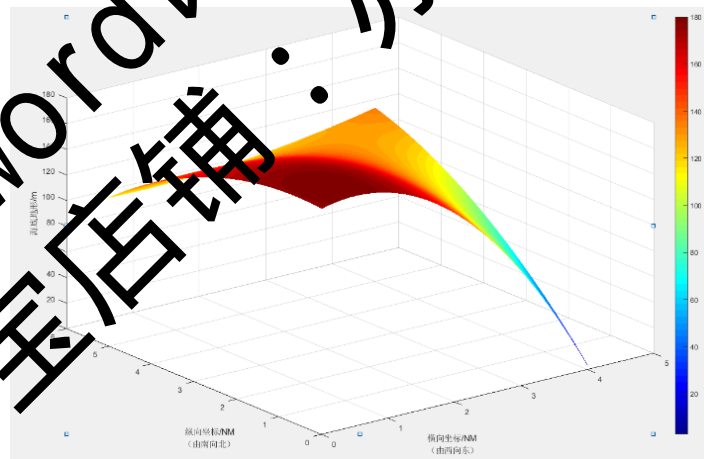


图 5-17 海水地形可视化示意图

由于附件中给出的是海水深度，求解出附件中最大的海水深度，将此海水深度减去附件中的所有值得到一个新的 251×201 的矩阵，将此矩阵通过 MATLAB 中对其进行自动插值，可以得到如上图较为平滑的可视化示意图 5-17。且对于图 5-17 作出如

下图所示的等深线地形图（图 5-18），可以较为明显的显示出的地形走势图。

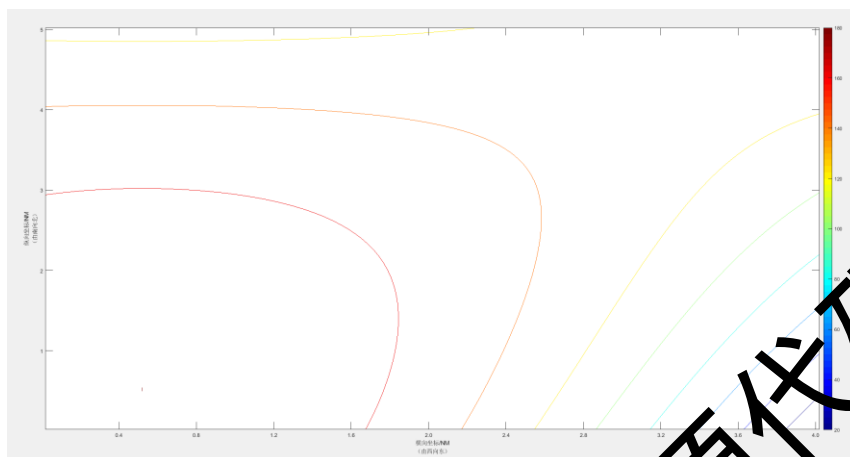


图 5-18 海水地形等深线示意图

5.4.2 模型的建模

根据多波束测深的原理和给定的多波束换能器角度，可计算出在不同深度下的覆盖宽度 W 。由问题二可得如公式(21)覆盖宽度关于倾斜时坡面角和多波束换能器角度的关系。

在设计测线时，需确保整个海域得到尽可能的覆盖，同时满足其他给定条件。首先，设计测绘船只的测线走向，以确保整个海域的覆盖，然后根据设计的测线，计算漏测面积和重叠面积，最后调整测线设计，以最小化总测线长度和漏测面积，同时控制重叠率。

对于初步测线来说，我们可以计算整个海域的平均覆盖宽度，并根据平均覆盖宽度和允许的最大重叠率（20%），来确定相邻测线之间的间距。

下面对于重叠区域宽度 Q 和重叠率进行定义，如果测线之间有重叠部分，可以对重叠区域宽度 Q 作如下定义：

$$Q = W - L \quad (39)$$

且漏测海区面积 M 满足以下公式：

$$M = S - MWL \quad (40)$$

重叠率超过 20% 的总长度 T_{sum} 为

$$T_{sum} = \sum_{i=1}^n (\lambda_i - 20\%) * W \quad (41)$$

其中仅考虑 $\lambda_i \geq 20\%$ 的情况。

在优化部分，定义测量路径为 P，在坐标(x,y)的深度为 D(x,y)，在深度为 D 处的覆盖宽度为 W(D)。

测量船只在一个海面上使用多波束探测的点位选择是一个典型的优化问题。它需要在满足一定条件下，找到一个最佳的航线坐标，使得船只最小化测线总长度 L、控制重叠率在 20%以下、最小化漏测海区面积 M、T 应为 0 或最接近 0。

为了解决这个问题，我们采用了一种启发式的优化方法，即粒子群优化算法。粒子群优化算法是一种模仿自然界中鸟群觅食行为的智能算法，它通过模拟鸟群中个体的位置、速度和适应度，以及个体之间的信息交流和协同作用，来寻找最优解。粒子群优化算法具有简单、易实现、收敛速度快等优点，已经被广泛应用于各种优化问题中。然而，经典的粒子群优化算法只适用于一维坐标轴上的变量，而我们的问题涉及到一个 2500×500 的二维坐标系，因此需要对算法进行改进。我们引入了二维粒子群优化算法，它在原有算法的基础上，增加了约束惩罚函数，使得算法能够处理二维坐标系中的非线性规划问题，并且能够考虑自变量的限制条件。通过这种方法，我们可以有效地求得测量位置的最优位置坐标，并且保证侧廊位置的测线总长度和覆盖宽度达到最佳水平。

经典粒子群算法是一种基于群体智能的优化算法，它模仿了鸟群或鱼群等自然现象中的协作和信息共享的行为，通过简单的规则来寻找最优解。经典粒子群算法的基本思想是：在一个多维搜索空间中，有一群粒子，每个粒子代表一个潜在的解，它有自己的位置和速度，同时也能记住自己和群体中最好的位置。粒子根据自己和群体的历史信息来调整自己的速度和位置，从而向更优的区域移动。经典粒子群算法的流程如下：

1. 初始化一定数量的粒子，随机生成它们的位置和速度。
2. 计算每个粒子的适应度值，即目标函数的值，并更新每个粒子和整个群体的最优位置。
3. 根据惯性权重、个体学习因子、群体学习因子和随机数来更新每个粒子的速度和位置。

4. 重复步骤 2 和步骤 3，直到达到最大迭代次数或满足终止条件。

经典粒子群算法的核心是速度和位置的更新公式，它们分别为：

$$v_{id}^{k+1} = \omega v_{id}^k + c_1 r_1 \text{left}(p_{id, \text{text} p_{best}}^k - x_{id}^k \text{right}) + c_2 r_2 \text{left}(p_{d, \text{text} g_{best}}^k - x_{id}^k \text{right})$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \quad (43)$$

其中， $i = 1, 2, \dots, N$ 表示粒子序号， $d = 1, 2, \dots, D$ 表示维度序号， k 表示迭代次数， ω 表示惯性权重， c_1 和 c_2 表示个体学习因子和群体学习因子， r_1 和 r_2 表示区间 $[0, 1]$ 内的随机数， v_{id}^k 表示第 i 个粒子在第 k 次迭代中第 d 维的速度向量， x_{id}^k 表示第 i 个粒子在第 k 次迭代中第 d 维的位置向量， $p_{id, \text{text} p_{best}}^k$ 表示第 i 个粒子在第 k 次迭代中第 d 维的历史最优位置，即在第 k 次迭代后，第 i 个粒子（个体）搜索得到的最优解； $p_{d, \text{text} g_{best}}^k$ 表示群体在第 k 次迭代中第 d 维的历史最优位置，即在第 k 次迭代后，整个粒子群体中的最优解。

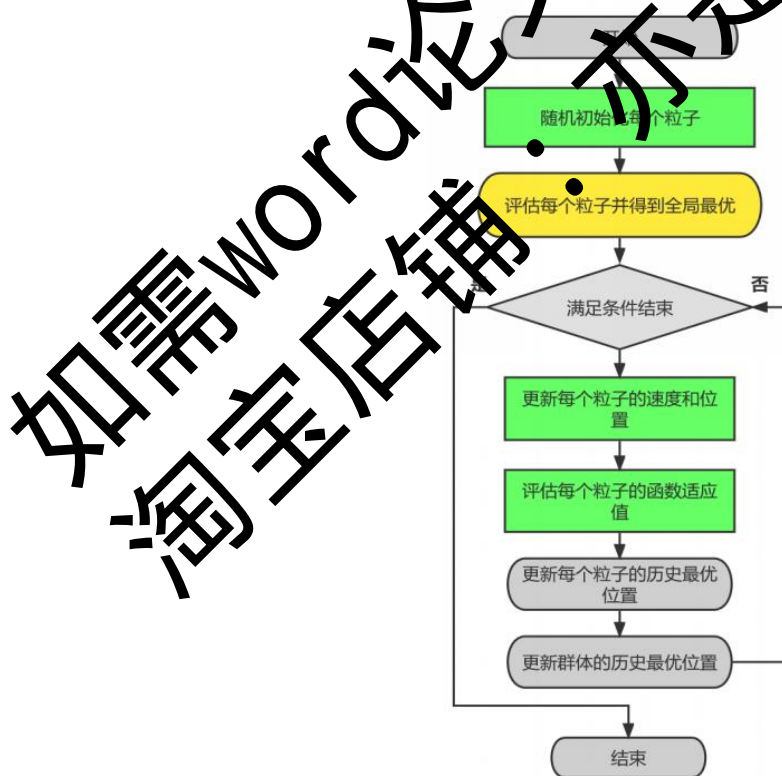


图 5-19 粒子群算法的流程图

为了利用二维粒子群算法高效地求解问题一中的规划函数，我们对经典粒子群算法进行了优化，通过引入惩罚函数来限制粒子的移动范围。同时，我们采用拉格朗日乘法，将约束条件转化为目标函数的惩罚项，把目标函数变为：

$$F(x) = f(x) + h(k)H(x) \tag{44}$$

其中，

$$h(x) = \sqrt{k}ork\sqrt{k}, H(x) = \sum_{i=1}^m \theta(q_i(x)) q_i(x_i)^{\gamma(q_i(x_i))} \tag{45}$$

式中， $f(x)$ ：原来的目标函数； $h(x)$ ：动态更新的惩罚系数，与迭代次数相关； $H(x)$ ：约束惩罚项； $q_i(x)$ 是相对约束惩罚函数； $\theta(q(x))$ 是分段赋值函数； $\gamma(q_i(x))$ 是惩罚指数。

将目标规划函数，以及 XY 取值 0-2500 代入粒子群，初始化输入相关系数。且在粒子群算法中取以下相关系数：

表 5-4 相关系数说明表

输入系数	含义
$n = 100$	粒子数量
$narvs = 2$	变量个数函数中有几个自变量
$c_1 = 2$	每个粒子的个体学习因子
$c_2 = 2$	每个粒子的社会学习因子
$w = 0.9$	惯性权重
$iter = 100$	迭代的次数
$V_{max} = 1.2$	粒子的最大速度查
$x_{-1} = 0$	下界
$x_{-bb} = -2500$	上届

5. 4. 3 模型的求解

下面对粒子群算法进行迭代分析，迭代效果如下图所示：

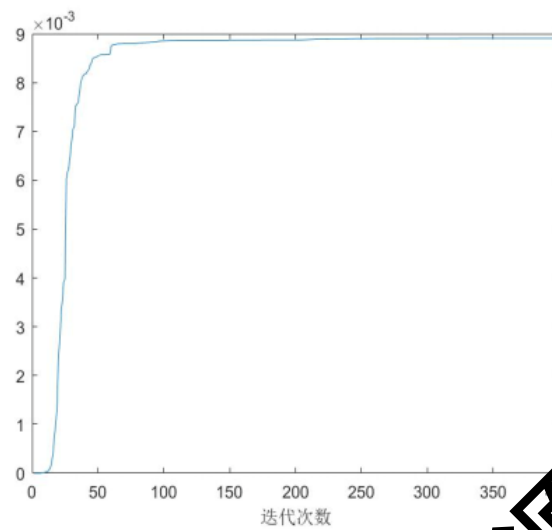


图 5-20 迭代效果示意图

从图 5-20 中可以看出，当我们用 100 个粒子进行粒子群算法迭代时，目标函数值在 150 次迭代后就基本稳定了，这说明我们的计算量已经足够发挥粒子群算法的优势。同时，图 5-20 中也显示了 PSO 的快速收敛性，没有出现阶梯式的变化，这表明我们的算法具有较高的精度和稳定性，粒子之间有较强的认知和交流能力。这些结果都证明了我们的算法是有效的，高效的，准确的。

得到的最优解效果如下

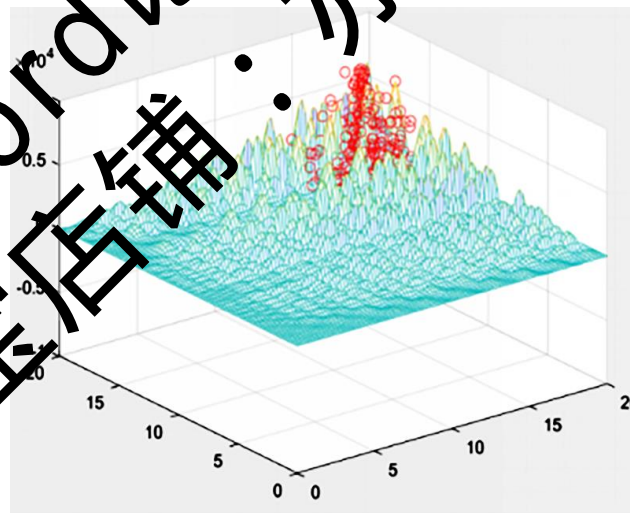


图 5-21 状态位置变换图

可知，粒子群在经过 200 多次的迭代以后，稳定保持在一个 (X_i, Y_i) 顶峰位置，则可以证明这组坐标为 (X_i, Y_i) 为目标函数的最优解。

测得最佳测量方式图下：

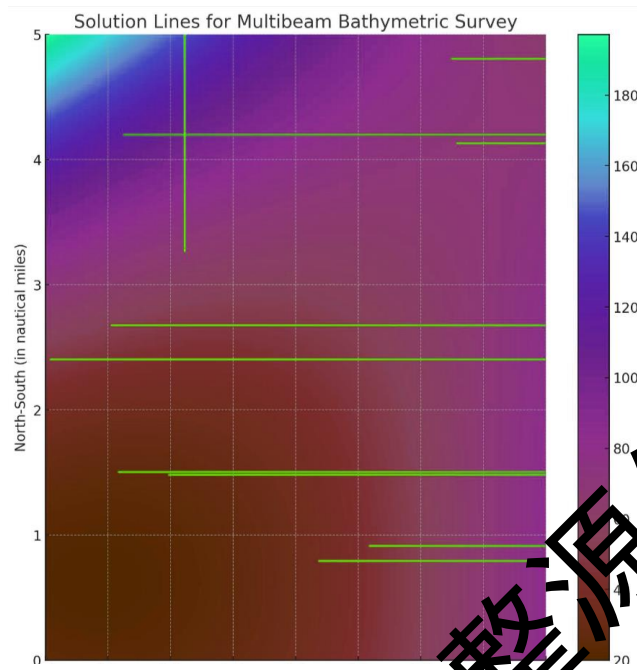


图 5-22 最佳测量方式示意图

使用 Python 对粒子群算法进行对应编码，对于代码存于附录 4。求得相对应的测线总长度为 44395.6m，由于粒子群算法求得为最优情况，故漏测海域占总计算海域的 0%，且在重叠区域中，重叠率超过 50%部分的总长度为 15120.7m。

六、模型的分析与检验

6.1 灵敏度分析

对多波束测模型进行灵敏度分析是为了评估系统对不同参数或环境条件的响应以及系统性能的稳定性的。

在问题一中，不难发现，当海水深度 D 和随换能器开角 θ 发生变化时多波束测深条带的覆盖宽度 W 也会随着变，下面更改这两个参数：

$$W = D \left\{ \frac{\sin \frac{\theta}{2}}{\sin(\frac{\pi}{2} + \alpha - \frac{\theta}{2})} + \frac{\sin \frac{\theta}{2}}{\sin(\frac{\pi}{2} - \alpha - \frac{\theta}{2})} \right\}$$

首先，我们可以固定一个参数（例如 θ ），然后改变另一个参数（例如 D ）。我们可以观察到，当 D 增加时， W 也会相应地增加。这表明，海水深度对多波束测深条带的覆盖宽度有显著影响。

同样，我们也可以固定 D ，然后改变 θ 。我们可能会发现，当 θ 增加时， W 也会增加。这表明，换能器开角对多波束测深条带的覆盖宽度也有显著影响。

通过这种方式，我们可以得到一个关于 D 和 θ 对 W 影响的灵敏度矩阵。这个矩阵可以帮助我们理解在实际操作中应如何调整这两个参数以优化多波束测深条带的覆盖宽度。

6.2 模型的检验

我们通过修改问题一的代码，将测线距中心线距离为0时海水深度 D 修改为120米，计算出当前深度下覆盖宽度为 $W = 416.6918699354501$ ，与问题2中测量船距海域中心点处的距离为0时误差结果如下表。

表 6-1 模型检验说明表

	问题 1 距中心处 距离为 0 时	问题 2 距中心处距 离为 0 时	绝对误差	相对误差
覆盖宽 度/m	416.69186993545	415.69219381653	-0.99967611892	-0.002399078
	416.69186993545	416.19140913868	-0.50046079677	-0.001201033
	416.69186993545	416.69186993545	0.00000000000	0
	416.69186993545	416.19140913868	-0.50046079677	-0.001201033
	416.69186993545	415.69219381653	-0.99967611892	-0.002399078
	416.69186993545	416.19140913868	-0.50046079677	-0.001201033
	416.69186993545	416.69186993545	0.00000000000	0
	416.69186993545	416.19140913868	-0.50046079677	-0.001201033

由上表可知，问题二模型计算的特殊结果与单问题一模型计算数据相比较，发现误差较小，结果几乎保持一致，检验通过。

七、模型的评价、改进与推广

7.1 模型的优点

模型的灵活性:本文中多波束测深的覆盖宽度的模型通过 z 逐步的加入各个变量,再进行简化,最终都能得到初始模型。即模型可以根据问题的不同需求进行调整和扩展,适应不同类型的问题和数据,灵活性高。

模型的准确性:本文模型基于几何关系证明,向量,三角函数等数学原理、迭代及数据分析,能够提供高度准确的预测和分析结果。

简单而易于实现:本文基于简单的数学原理,容易理解和实现。

可重复性强:本文的模型在逐渐加入条件后,基本符合各种情况,可以在不同的情境下反复使用,而不需要重新创建,从而提高工作效率。

7.2 模型的缺点

本文的模型在计算相邻两条测线的间距时,时间成本较高。

本文模型在计算实际海域的模型中局部搜索能力较差,搜索精度不够高。

7.3 模型的改进

自适应参数:可以采用自适应的迭代策略,例如逐步减小学习率,以提高在计算相邻两条测线的间距时收敛速度。

混合算法:将粒子群算法与其他优化算法结合,以克服其早熟收敛的问题。

7.4 模型的推广

本文建立了三维空间中多波束测深系统的条带覆盖宽度及重叠率的模型。而在不同的应用场景中,可能需要根据具体需求调整条带覆盖宽度和重叠率。例如,在测绘航道时,可能需要更高的覆盖宽度和较低的重叠率,以提高测量速度。而地质调查中,可能需要更高的地形分辨率,因此可以采用较小的覆盖宽度和较高的重叠率。同时,还可以考虑系统的技术参数,如声波频率、发射功率、接收灵敏度等,以进一步优化性能。

八、参考文献

- [1] 舒升元, 程龙. 多波束系统在长江航道测量中的应用分析[J]. 中国水运, 2017
- [2] 朱庆, 李德仁. 多波束测深数据的误差分析与处理[D]., 1998.
- [3] 张国英, 吴艺娟. 基于多级惩罚函数的粒子群约束优化算法[J]. 北京石油化工学院学报, 2008, 16(4): 30-32.
- [4] Shi Y, Eberhart R C. Fuzzy adaptive particle swarm optimization[C]//Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546). IEEE, 2001, 1: 101-106.1

如需word论文和完整源代码
淘宝店铺：亦是远方

附录

附录 1

介绍：支撑材料的文件列表

result1.xlsx
result2.xlsx

附录 2

介绍：该代码是 Python 语言编写的，用于求解问题一

```
1. import numpy as np
2. import pandas as pd
3. data = pd.read_csv('result2.csv', encoding='gbk')
4. #初始化
5. Sea_mile = 1852 #单位转化
6. a0 = np.deg2rad(1.5) #海底坡面与水平面夹角
7. D0 = 120 #测线距中心线距离为 0 时海水深度
8. ct = np.deg2rad(120) #多波束换能器的开角
9. k = 0
10.
11. for b0 in range(0, 315 + 1, 45):
12.     Covering_width = [] #覆盖宽度
13.     print('当  $\beta = %d$  时' % b0, end=' ')
14.     Beta_Angle = np.deg2rad(b0) #贝塔角
15.     a = np.sin(Beta_Angle) * 0 #真实投影角
16.     print('a = ', np.rad2deg(a))
17.     a1 = (np.pi - ct) / 2 + a #多波束换能器与坡面左侧夹角
18.     a2 = (np.pi - ct) / 2 + a #多波束换能器与坡面右侧夹角
19.     for i in range(0, 2.2, 0.3):
20.         x = i * Sea_mile #测量点与海域中心点处的距离，单位米
21.         #覆盖宽度计算
22.         if Beta_Angle == 0: #特殊情况
23.             W = 2 * (D0 + x * np.tan(a0)) * np.tan(ct / 2)
24.             Covering_width.append(W)
25.         print(W)
26.         else:
27.             W = (D0 + x * np.tan(a) / np.tan(Beta_Angle)) * np.sin(ct / 2) * (1 / np.sin(a1) + 1 / np.sin(a2))
28.             Covering_width.append(W)
29.         print(W)
30.     for i in range(len(Covering_width)):
31.         data.iloc[k, i + 1] = Covering_width[i]
32.         k += 1
33. data.to_csv('result2.csv', index=None, encoding='gbk')
```

附录 3

介绍：该代码是 Python 语言编写的，用于求解问题二中的覆盖宽度 W

```
1. import numpy as np
2. import pandas as pd
3. data = pd.DataFrame(columns=['当前测线位置序号', '海水深度', '距上一条测线距离', '覆盖率'])
4. #初始化
5. Sea_mile = 1852 #单位转化
6. D0 = 110 #测线距中心线距离为 0 时海水深度
7. Length = 2 * Sea_mile #矩形海域长
8. Width = 4 * Sea_mile #矩形海域宽
9. a0 = np.deg2rad(1.5) #海底坡面
10. ct = np.deg2rad(120) #多波束换能器的开角
11. a1 = (np.pi - ct) / 2 - a0 #多波束换能器与坡面左侧夹角
12. a2 = (np.pi - ct) / 2 + a0 #多波束换能器与坡面右侧夹角
13. target_rate = 0.1 #最优覆盖率
14.
15. #计算起始测线位置
16. dt = Length * np.tan(a0) #中心海域与最深处高度差
17. H = D0 + dt #最深处深度
18. s0 = H / np.tan(a0+a1) #起始测线距西边界距离
19.
20. #计算末位置测线
21. l0 = D0 / np.tan(a0) #理想岸边距离
22. h = (l0 - Length)*np.tan(a0) #最末处海水深度
23. sn = Width - h * np.tan(ct/2) #末位置测线距西边界距离
24.
25. s = [] #各个测线间距
26. s.append(0)
27. # 2
28. data.append(['当前测线位置序号': 1, '海水深度': np.tan(a0+a1)*s0,
29.             '距上一条测线距离': s0, '覆盖率': '—'])
30. # ignore_index=True)
31. while sum(s) < sn:
32.     rate_temp = 0
33.     sx_temp = 0
34.     y_temp = 0
35.     cnt = 0
36.     for sx in range(20,600):
37.         #print('第%d个测线, 迭代%d次' % (i,cnt))
38.         cnt += 1
39.         #print('距离上一条航线 %d 米' % sx)
40.         D = H - (sx + sum(s)) * np.tan(a0) # 当前海水深度
```

```

40.         #print('\t 海水深度:', D)
41.         W = D * np.sin(ct / 2) * (1 / np.sin(a1) + 1 / np.sin(a2))
42.         #print('\t 测量宽度:', W)
43.         # ans = ((1 - rate) * W * np.sin(a1)) / np.cos(ct / 2)
44.         rate = 1 - (sx * np.cos(ct / 2) / (W * np.sin(a1)))
45.         #print('\t 覆盖率: ', rate)
46.         #print('\t 当前测量到: ', sum(s))
47.         if rate < 0.12:
48.             if rate < target_rate:
49.                 data = data.append({'当前测线位置序号': i, '海水深度': D_temp,
50.                                     '距上一条测线距离': sx_temp, '覆盖率': rate_temp}, ignore_index=True)
51.                 s.append(sx_temp) #最优
52.                 break
53.             else:
54.                 D_temp = D
55.                 sx_temp = sx
56.                 rate_temp = rate
57.                 ...
58.                 ...
59.                 ...
60.             i += 1
61.             ...
62.         data.to_excel('result-go.xlsx', index=None)
63.         print('测线总距离为: ', length*len(s))

```

附录 4

介绍：该代码是用 Python 语言编写的，针对于问题四的粒子群优化算法求解

```

1.     import numpy as np
2.
3.     class Particle:
4.     def __init__(self, dim, minx, maxx):
5.         self.pos_dor = np.random.uniform(low=minx, high=maxx, size=dim)
6.         self.vel_city = np.random.uniform(low=-0.1, high=0.1, size=dim)
7.         self.best_position = np.copy(self.position)
8.         self.best_score = -np.inf
9.
10.    class PSO:
11.    def __init__(self, dim, n_particles, bounds):
12.        self.dim = dim
13.        self.n_particles = n_particles
14.        self.bounds = bounds
15.        self.particles = [Particle(dim, *bounds) for _ in range(n_particles)]

```

```

16.
17.     def optimize(self, func, n_iter):
18.         global_best_score = -np.inf
19.         global_best_position = None
20.
21.         for i in range(n_iter):
22.             for particle in self.particles:
23.                 score = func(particle.position)
24.
25.                 if score > particle.best_score:
26.                     particle.best_score = score
27.                     particle.best_position = particle.position
28.
29.                 if score > global_best_score:
30.                     global_best_score = score
31.                     global_best_position = particle.position
32.
33.             for particle in self.particles:
34.                 particle.velocity += 2 * np.random.rand(self.dim) * (particle.best_position - particle.position) + 2
35.                 * np.random.rand(self.dim) * (global_best_position - particle.position)
36.                 particle.position += particle.velocity
37.
38.             for i in range(self.dim):
39.                 if particle.position[i] < self.bounds[0]:
40.                     particle.position[i] = self.bounds[0]
41.                 elif particle.position[i] > self.bounds[1]:
42.                     particle.position[i] = self.bounds[1]
43.
44.             return global_best_position
45.
46.     def fitness(position):
47.         total_length = np.sum(position)
48.         missed_area_percentage = np.sum(data[position == 0]) / np.sum(data)
49.         overlap_length = np.sum(position[position > 1] - 1)
50.
51.         # 希望最小化这三个指标
52.         return total_length + missed_area_percentage + overlap_length
53.
54. pso = PSO(dim=20, n_particles=30, bounds=(0, 1))
55. best_position = pso.optimize(fitness, n_iter=1000)

```