

多波束测线技术的模型研究

摘要

多波束测线问题是在单波束测深的基础上发展起来的，多波束测深系统克服了单波束测深的缺点，在海底平坦的海域内，能够测量出以测量船测线为轴线且具有一定宽度的全覆盖水深条带。本文基于多波束基本要素，如海洋中心深度、覆盖宽度、重叠率等的研究，探究其彼此之间的关系，并完成寻找符合多目标要求的海域测线。

针对问题一，我们借助几何模型及正弦定理，通过模型强化、模型理想化，将多种要素建立数学联系，结合python软件，发现海水深度、覆盖宽度、与前一条覆盖宽度的重叠率，都会随着测线距中心点的距离的增大而下降。

针对问题二，在坡面法线及其投影、测线方向、坡面角度多种约束条件下，结合python软件，寻找他们在水平面及其切面上的夹角关系等，寻找覆盖宽度在不同测线方向夹角和不同测线中心距离下的值，并观察总结其变化趋势，为探究问题三和问题四做铺垫。

针对问题三，我们建立几何模型，我们将难点定位在测线方向和测线间距。首先探究相对适宜的测线方向，将测线方向定位在南北向，将理想初始情况设定为最西侧的测线恰好满足限制条件，并通过c++依次向东递推，发现测线间距呈下降趋势，取覆盖度为10%，求出测线数量及其总长度。

针对问题四，我们结合前三题的做题经验及其计算公式，根据题目的相应指标，运用贪婪算法，推算出满足多目标条件的测线。在实现上述目标后，我们运用python软件计算出该种情形下的测线指标。

关键词：多波束测线；覆盖宽度；重叠率；贪婪算法；正弦定理

1. 问题重述

1.1 问题背景

单波束测深是根据声波在水体中的传播特性来反映水体深度的技术。声波在均匀介质中作匀速直线传播，在不同水体面上产生反射，利用这一原理，通过声波在水体中的传播速率计算出海水的深度。多波束测深系统是在单波束测深的基础上发展起来的，多波束测深系统克服了单波束测深的缺点，在海底平坦的海域内，能够测量出以测量船测线为轴线且具有一定宽度的全覆盖水深条带。

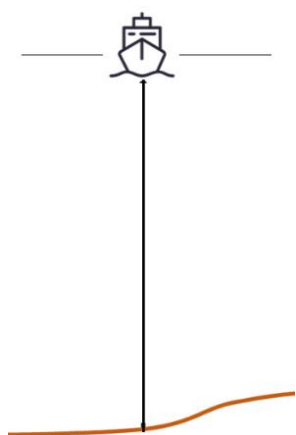


图1 单波束测深的工作原理

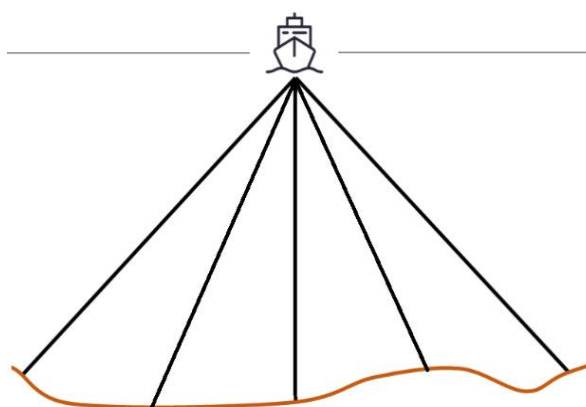


图2 多波束测深的工作原理

1.2 问题要求

基于上述背景我们需要建立数学模型解决以下问题：

- (1) 通过理论分析与实际经验相结合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。
- (2) 考虑一个矩形待测海域（如图3），测线方向与海底坡面的法向在水平面上投影的夹角为 β ，建立多波束测深覆盖宽度的数学模型。
- (3) 考虑一个南北长 2 海里、东西宽 4 海里的矩形海域内，海域中心点处的海水深度为 110 m，西深东浅，坡度为 1.5° ，多波束换能器的开角为 120° ，通过模型建立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需满足 10%~20% 的要求。
- (4) 利用多年前的一组数据为多波束测量船的测量布线提供帮助。在设计测线时，满足如下要求：a. 沿测线扫描形成的条带尽可能地覆盖整个待测海域；b. 相邻条带之间的重叠率尽量控制在

20% 以下；c. 测线的总长度尽可能短。在设计出具体的测线后，并计算如下指标：a. 测线的总长度；b. 漏测海区占总待测海域面积的百分比；c. 在重叠区域中，重叠率超过 20% 部分的总长度。

2. 问题分析

2.1 对于问题（1）的分析

问题一首先给出与测线垂直的平面和海底坡面的交线构成一个夹角 α ，考虑到坡度 α 会影响测线在海底的投射和反射，从而影响覆盖宽度和重叠率，需要分析坡度与声束角度之间的关系。其次根据给定测线距离中心点的距离，进行数学模型的建立，其中数学模型是根据测线、海底坡度、开角等之间的几何关系，利用正弦定理建立。最后在此模型的建立下，随着测线距离中心点的改变，可计算覆盖宽度以及覆盖率。最后通过实验或真实数据验证模型与参数拟合，

2.2 对于问题（2）的分析

问题二首先基于测线的夹角以及船距海域中心点处的距离这两个变量关系来计算覆盖宽度，在问题一的基础上新增一个矩形海域约束条件以及新的变量，即测线方向与海底坡面的法向在水平面上投影的夹角 β ，其次我们引入一个新的变量参数 β 来修正数学模型，最后通过真实或实验数据验证模型与参数拟合，我们可以收集不同坡度和测线方向下的实际覆盖宽度数据。

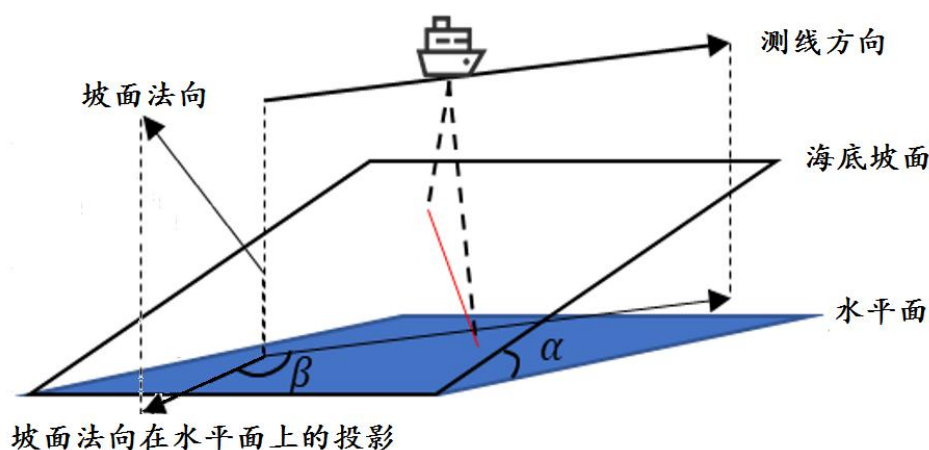


图3 问题二的示意图

2.3 对于问题（3）的分析

问题三首先给出一块特定的矩形区域，通过建立某种数学模型来求出如何设计最短的测线，以在最大程度上减少测线的总长度，同时确保能够覆盖整个待测海域。我们可以考虑使用东西方向或南北方向

的测线，这样可以简化设计。最后通过模型的建立以及真实或实验数据验证模型与参数拟合，从而给出符合要求的测线。

2.4 对于问题（4）的分析

首先利用给定的近几年单波束测深数据，希望建立各参量之间的数学模型，来为多波束测量船的测量布线作帮助，其中测线需满足：测线扫描形成的条带尽可能覆盖整个待测海域，且重叠率尽可能控制在20%以下，要求测线的总长度尽可能短，在利用数学模型求解以便设计出具体的测线，接下来求出测线的总长度、漏测海区占总待测区域面积的百分比和重叠区域超过20%的总长度等数据。

3. 模型假设

- 1. 假设没有其他海底地形和水下环境影响。
- 2. 假设声波在水中以恒定速度传播，不考虑声波在水中的折射。
- 3. 假设测量点与目标之间的距离足够远，以便近似为远场条件，使用波束形成技术，以从不同方向接收信号，而不必考虑波的相位变化。
- 4. 假设接收到的信号是窄带信号，即信号的带宽相对较小，可以在频域上近似为单一频率。该假设允许使用频域分析方法。
- 5. 假设信号源是均匀分布的，或者至少在某个时间段内保持不变。这个假设有助于信号处理和波束形成。

4. 变量说明

| 符号 | 含义 |
|----------|----------------------------|
| α | 指海底坡面与与测线方向垂直的平面的夹角。（单位：度） |
| W | 在一次测量中所能获取的海底宽度。 |
| η | 重叠率:相邻条带之间的交叠部分所占的比例 |

| | |
|----------|-------------------------------|
| θ | 多波束换能器的开角,表示每个波束的发射 |
| β | 测线方向与海底坡面的法向在水平面上投影的夹角。(单位:度) |
| D | 海水深度,(单位:米) |
| d | 相邻条带之间的距离(单位:米) |

5. 模型的建立与求解

5.1 对于问题一的模型建立与求解

5.1.1 问题一模型分析

问题一希望建立一个能表征垂直深度与覆盖宽度、海水深度以及相邻宽带之间的重叠率之间关系的函数模型,建模时我们需考虑换能器开角、海水深度、以及其他因素如海底坡度 α 对测深的覆盖宽度与相邻条带覆盖率的影响。本质上是使用方程思想对几何问题的求解。由于测线方向固定,首先建立测线及海水深度的模型,利用正弦定理的对距中心点不同距离的测线所测深度进行求解,然后根据所解出海水深度模型建立测量条带覆盖宽度模型,利用正弦定理与平面三角形的性质分别求出较深侧和较浅侧条带覆盖宽度,将其求和得到该位置测量条带覆盖宽度,最后根据公式算出条带重叠率。

5.1.2 问题一模型的建立

在平坦海面上重复率的计算公式:

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

如图1所示,在海底坡面下方的三角形中利用三角函数求解

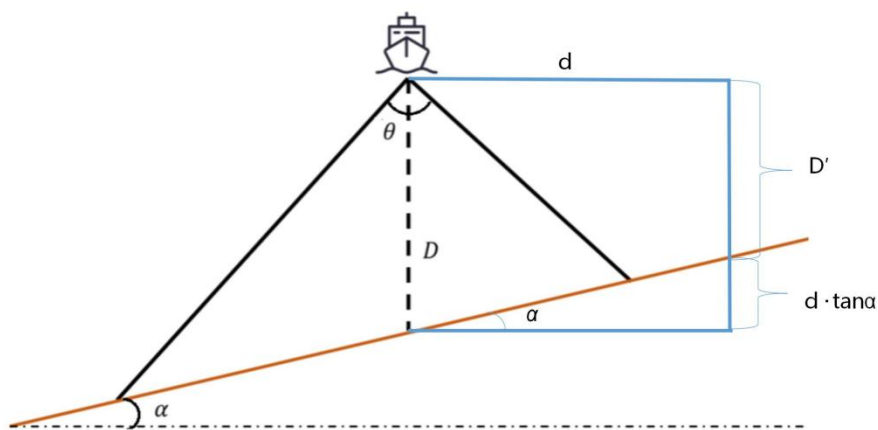


图1 海水深度示意图

可得出第二条测线测量海水深度的表达式，即：

$$D' = D - x \tan \alpha \quad (2)$$

问题中要求相邻条带之间应有10%~20%的重叠率。因此，我们需要调整测线的间距d以满足这一要求。通常，可以尝试不同的间距值并计算相应的重叠率，以找到合适的间距，使得重叠率在要求范围内。

使用以上步骤，通过改变距离中心点的位置来模拟不同的测线布局，可以在不同距离中心点的位置计算出覆盖宽度W和相邻波束之间的重叠率η。

其中，D表示测线距离中心点处的距离

如图2所示，

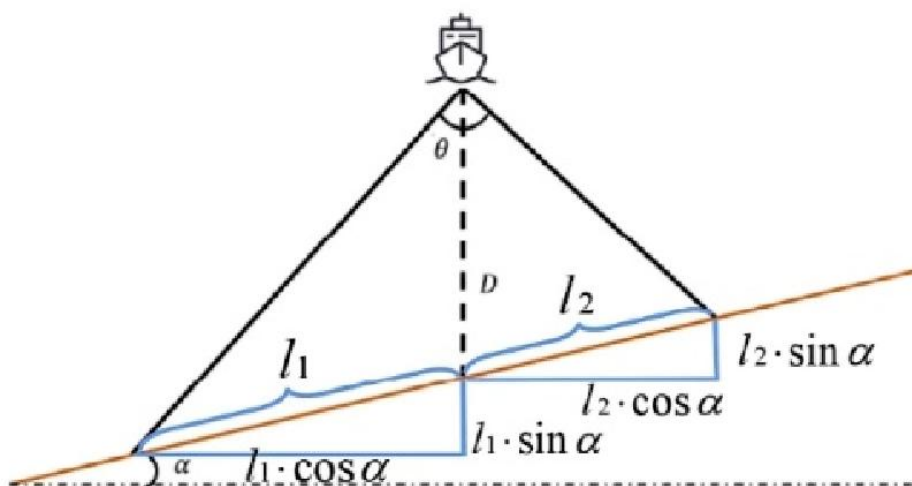


图2 覆盖宽度示意图

计算相邻波束之间的重叠率η： $1 - \frac{d}{W}$

其中 $D = d - x \tan \alpha$

计算 l_1 和 l_2 ,

$$l_1 = D \cdot \frac{\sin 60^\circ}{\sin(30-\alpha)} \tag{3}$$

$$l_2 = D \cdot \frac{\sin 60^\circ}{\sin(30+\alpha)} \tag{4}$$

$$W=(l_1 + l_2) \cdot \cos \alpha \tag{5}$$

计算出W。

5.1.3 问题一模型的求解

通过python编程求得结果如下表：

表1 问题一所求结果表

| 测线据中心点处的距离/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 | 315.71 | 297.53 | 279.35 | 261.17 | 242.99 | 224.81 | 206.63 | 188.45 | 170.27 |
| 与前一条测线的重叠率/% | —— | 32.78 | 28.40 | 23.42 | 17.69 | 11.03 | 3.21 | -6.13 | -17.46 |

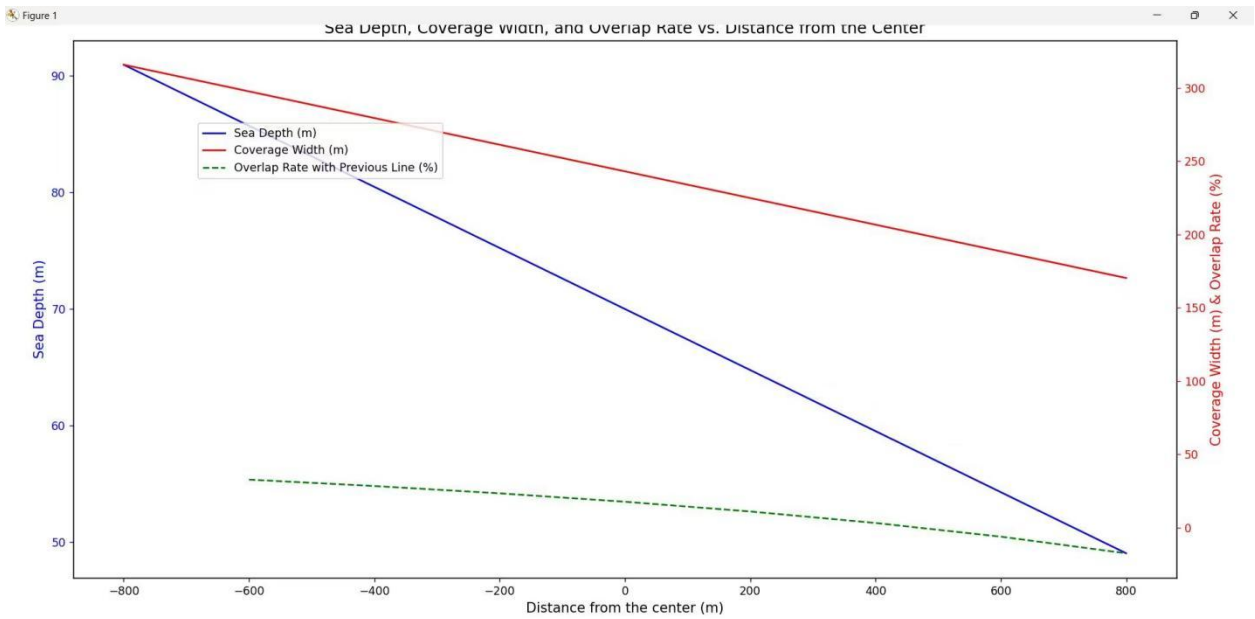


图3 海水深度、覆盖深度、重叠率与测线距中心点距离的关系图

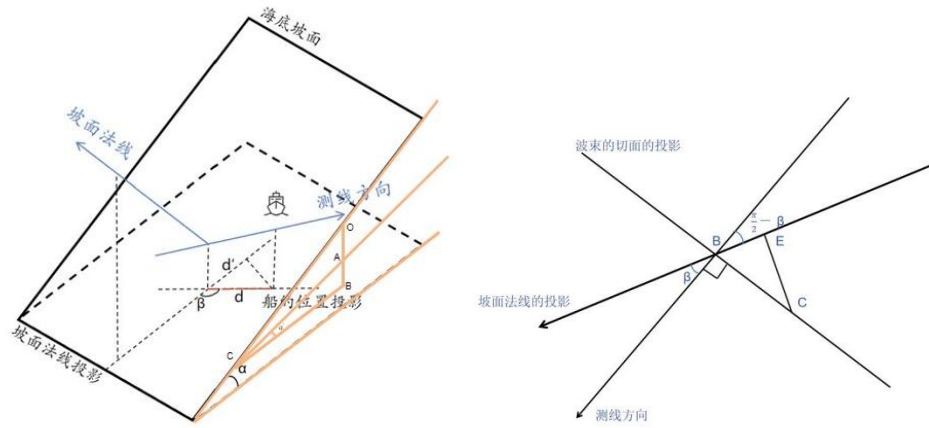
5.2 对于问题二的模型建立与求解

5.2.1 问题二模型的分析

问题二在测线方向可变、海底地形为坡面的矩形待测海域的情况下，建立多波束测深覆盖宽度的数学建模，解出各方向测线距海域中心点处不同距离的条带覆盖宽度。测线方向以及船距海域中心距离发生改变时，通过投影的夹角 β 和测量船距海域中心点处的距离 d 之间的函数关系，建立各方向测量船的海水深度（ D' ）模型，利用坡法线投影与船位置投影的函数关系，对距中心点不同距离的测线所测覆盖宽度进行求解，然后根据所解出覆盖宽度模型建立测线方向投影夹角模型，利用船距离中心点的距离和测线方向的数据求出在此情况下的覆盖宽度。

5.2.2 问题二模型的确立

得到的水深的模型，如图3所示：



$$AC=W_1, AB=W_1 \sin \alpha', BC=\cos \alpha', BE=BC \cos \frac{\pi}{2} - \beta$$

图4 图3俯视示意图

计算单个波束的覆盖宽度 W ，单个波束的覆盖宽度表示波束在海底上的宽度，我们使用以下公式来计算它：

$$W=(D+1852 \cdot x \cdot \tan 1.5^{\circ} \cdot |\sin \beta|) \cdot \left(\frac{\sin 60^{\circ}}{\sin (30^{\circ}-\alpha')} + \frac{\sin 60^{\circ}}{\sin (30^{\circ}+\alpha')} \right) \cdot \cos \alpha' \quad (6)$$

其中，当前的海水深度 $= (D+1852 \cdot x \cdot \tan 1.5^{\circ} \cdot |\sin \beta|)$ ，以及当前的坡度角 $\alpha' = \arctan (\tan 1.5^{\circ} \cdot |\sin \beta|)$ 均和 β 有关，根据不同的侧线方向夹角 β ，我们可以算出不同方向上的覆盖宽度，通常需要考虑360度中的各个方向，并计算每个方向上的覆盖宽度，而且实际的覆盖宽度是在坡面的投影宽度。

5.2.3 问题二模型的求解

通过python编程求得结果如下表：

表2 问题二所求结果表

| 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 | 416.12 | 451.79 | 487.47 | 523.14 | 558.82 | 594.49 | 630.17 | 665.84 |
| | 90 | 416.55 | 416.55 | 416.55 | 416.55 | 416.55 | 416.55 | 416.55 | 416.55 |
| | 135 | 416.12 | 380.45 | 344.78 | 309.10 | 273.42 | 237.75 | 202.08 | 166.40 |
| | 180 | 415.69 | 365.29 | 314.89 | 264.50 | 214.10 | 163.70 | 113.30 | 62.90 |
| | 225 | 416.12 | 380.45 | 344.77 | 309.10 | 273.42 | 237.75 | 202.08 | 166.40 |
| | 270 | 416.55 | 416.55 | 416.55 | 416.55 | 416.55 | 416.55 | 416.55 | 416.55 |
| | 315 | 416.12 | 451.79 | 487.47 | 523.14 | 558.82 | 594.49 | 630.17 | 665.84 |

覆盖宽度随着距离、角度的改变，其覆盖宽度变化如图：

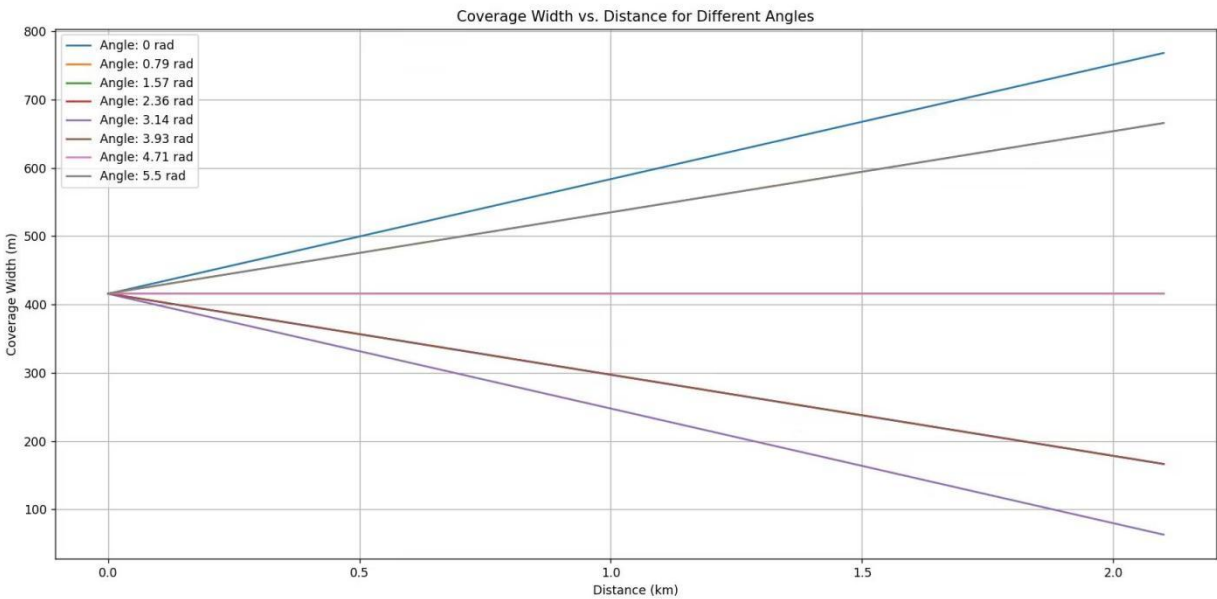


图5 不同角度的覆盖宽度与距离

5.3 对于问题三的模型建立与求解

5.3.1 问题三模型的分析

基于问题二的结论，我们的目标是设计最短的测线，以在最大程度上减少测线的总长度，同时确保能够覆盖整个待测海域。我们可以考虑使用东西方向或南北方向的测线，这样可以简化设计。观察问题二的结论表格，我们发现，当测线的角度 β 等于 90° 时，覆盖宽度趋于一个稳定值。因此，在南北方向上，我们需要确保波束的宽度("W")足够大，以便在海域的最浅处和最深处都能覆盖整个深度范围。同时，相邻测线之间的间距需要根据所需的重叠率（通常在10%到20%之间）来计算。接下来，我们可以使用模型来计算所有相邻测线之间的间距之和，再加上南北方向的宽度总和。最后，我们可以通过微调波束宽度和相邻测线的间距，以使总测线长度最小。

这种方法将帮助我们设计出满足覆盖要求的最短测线布局，以便在测量水深时既节省时间又保证数据的准确性。

5.3.2 问题三模型的确立

基于对建模前期的准备，我们对于根据海水深度和坡度求出覆盖密度，进而确立相邻测线间距的间距，并将测线方向确定在南北方向。我们给出建立的多目标模型：

如图6所示：

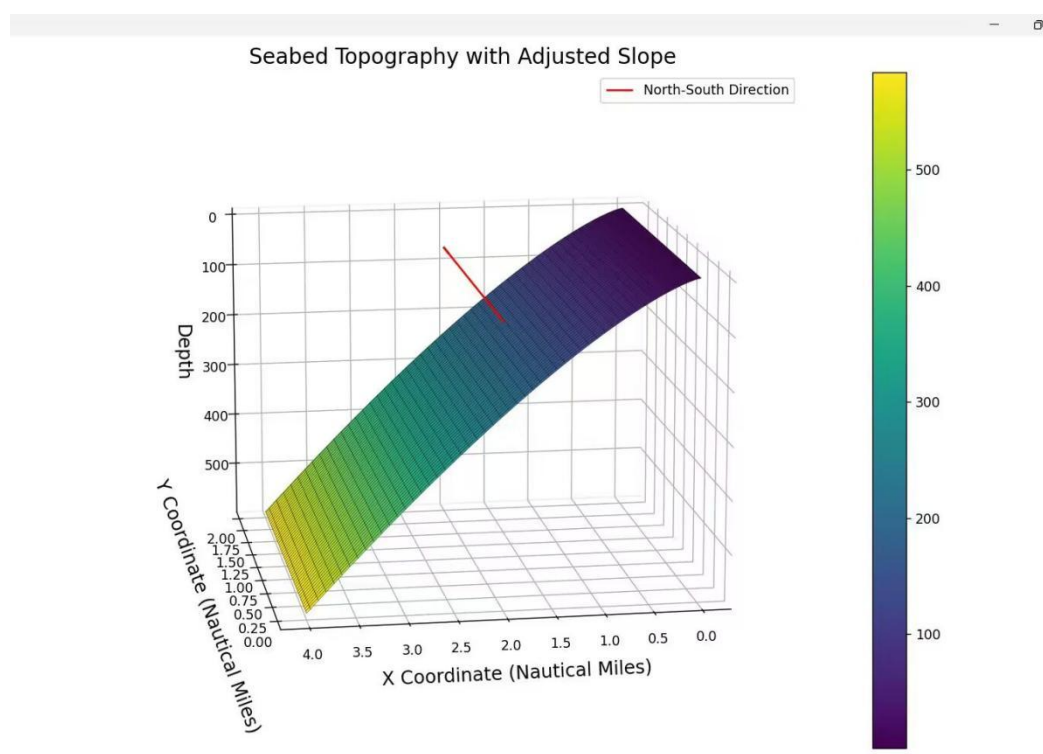


图6 覆盖宽度与海水深度和坡度关系示意图

(坡面西深东浅，测线方向沿南北向。)

5.3.3 问题三模型的求解

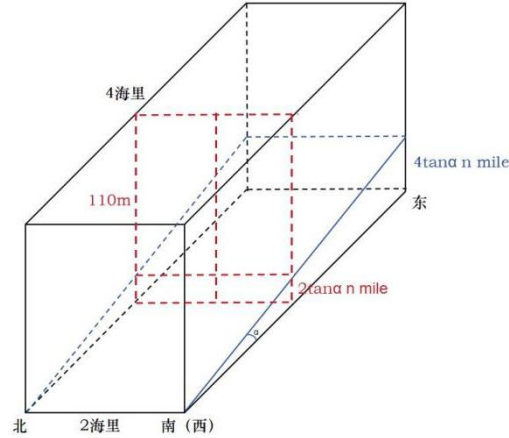


图7 矩形海域示意图

首先对将矩形海域扩充成为一个长方体，利用海洋中心点的深海距离及坡面角度，算得到该长方体的高度以及海洋中心处到水平面的距离。

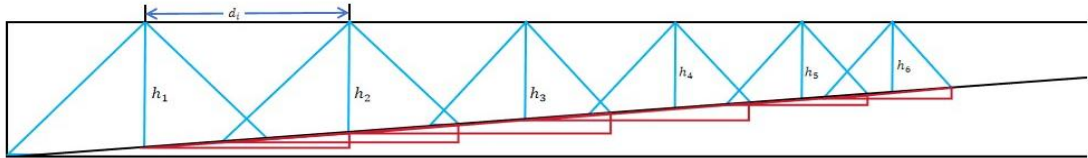


图8 作重叠覆盖宽度示意图

做一个与航线垂直的竖直平面，我们可以得到多个有重叠的覆盖宽度，利用三角知识与代换，得到：自西向东第一个点的覆盖宽度和水深，并推算出自西向东第*i*条测线对应的覆盖宽度和水深；

我们假设覆盖率稳定在10%，且西临界恰好在包含在范围内，

$$\text{联立方程组: } \begin{cases} \eta = 1 - \frac{d_i}{w_i} \\ h_i = h_{i-1} - d_i \cdot \tan \alpha \end{cases} \quad (7)$$

得

$$h_i = \frac{h(i-1)}{1+0.9 \left(\frac{\sin(\frac{\theta}{2})}{\sin(\frac{\pi}{2}-\frac{\theta}{2}+\alpha)} + \frac{\sin(\frac{\theta}{2})}{\sin(\frac{\pi}{2}-\frac{\theta}{2}-\alpha)} \right) \sin \alpha} \quad (8)$$

$$d_i = \frac{0.9h(i-1)\cos \alpha \left(\frac{\sin(\frac{\theta}{2})}{\sin(\frac{\pi}{2}-\frac{\theta}{2}+\alpha)} + \frac{\sin(\frac{\theta}{2})}{\sin(\frac{\pi}{2}-\frac{\theta}{2}-\alpha)} \right)}{1+0.9\sin \alpha \left(\frac{\sin(\frac{\theta}{2})}{\sin(\frac{\pi}{2}-\frac{\theta}{2}+\alpha)} + \frac{\sin(\frac{\theta}{2})}{\sin(\frac{\pi}{2}-\frac{\theta}{2}-\alpha)} \right)} \quad (9)$$

得出的代码运行后解得结果见附件。

5.4 对于问题四的模型建立与求解

5.4.1 问题四模型的分析

题目要求在一个固定大小的海域（南北长5海里、东西宽4海里）内设计多波束测量船的测量布线。在设计测线时，需要满足三个条件：

1. 沿测线扫描形成的条带尽可能地覆盖整个待测海域；
2. 相邻条带之间的重叠率尽量控制在20%以下；
3. 测线的总长度尽可能短。

其建模思路：根据决策变量建立目标函数，最小化测量带的总数，从而间接达到最小化测线总长度的目的。

其约束条件为整个区域需要被测量带完全覆盖，且任何两个相邻的测量带之间的重叠部分不得超过20%。

接下来进行模型求解：使用线性规划工具或优化库（如SciPy的minimize函数）来求解上述模型。通过优化，我们可以确定最佳的测量带位置，从而确定最佳的测线布局。

最后根据优化结果，我们可以确定在东西方向上应放置的测线位置。然后，考虑到南北方向的固定长度（5海里），我们可以计算总测线长度、漏测海区面积的百分比以及超过20%的重叠长度。

5.4.2 问题四模型的确立

通过决策变量与目标函数的关系，得到目标函数为：

$$\min \sum_{i=1}^n x_i$$

其覆盖约束：即为确保整个待测海域都被覆盖。

$$\sum_{i=1}^n \omega \geq 4 \text{海里}$$

重叠约束:相邻测线之间的重叠不超过20%,如果 x_i 和 x_j 都被选择($i \neq j$),那么 $|x_i - x_j| \geq \omega - 0$,

边界约束:测线的起始位置和结束位置都应该在给定的海域范围内。

$$0 \leq x_i \leq 4 - \omega, \forall i$$

5.4.3 问题四的模型求解:

使用贪婪算法求解模型,为了求解此模型,我们采用了贪婪算法。贪婪算法是一种基于当前最优选择的方法,它为每一步选择局部最优解,希望这样能得到全局的最优解。

对于本问题,我们从最西边的边界开始,放置第一条测线,然后根据给定的20%的重叠率移动到下一个位置放置下一条测线,如此继续,直到整个待测区域都被覆盖。

结果分析:

使用贪婪算法,我们得到了以下测线的起始位置:0海里、0.8海里、1.6海里、2.4海里和3.2海里。

根据这些位置,我们进一步计算了相关指标:

- 1.测线的总长度:由于每条测线的南北长度都是5海里,所以总长度为 $5 \times$ 测线数量;
- 2.漏测海区的百分比:由于测线覆盖了整个待测区域,所以漏测百分比为0%;
- 3.超过20%的重叠长度:由于我们设计的测线恰好有20%的重叠,所以超出的部分为0海里。

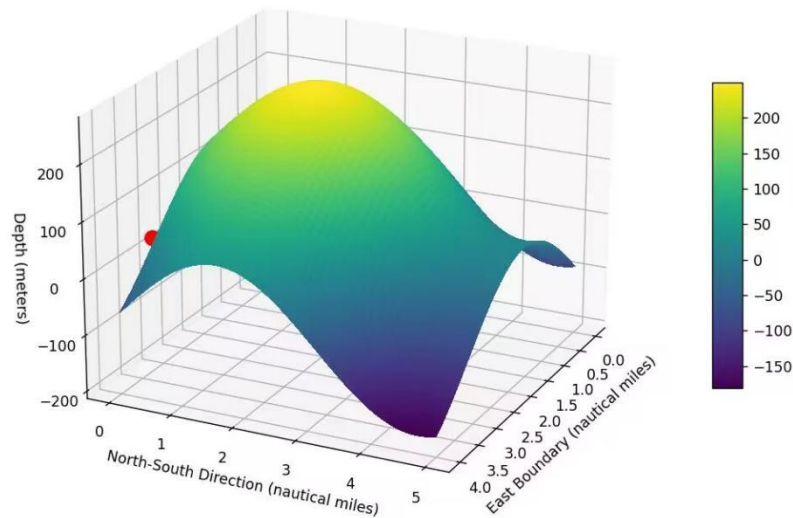


图9 地形数据图

6. 模型的检验

标准解与模型所求解的误差：由于本模型使用了搜索算法、贪婪算法对问题进行求解，且对多目标约束问题进行了分析，可能存在算法所带来的误差，而且理论值与实际测量值本来就会产生误差，两者综合就是模型会产生的第一种误差。

局部最优解造成的空间坐标误差，遗传群算法的本身原理会导致其收敛在局部的最优解，为了减少运算量，也为了避免模型陷入局部最优，在程序运行过程中，有时候需要人为干预来筛选一些外部环境因素的干扰，可能造成冗余解或缺解。

7. 模型的评价与推广

7.1 模型的优点

1. 高精度：多波束测深模型可以同时测量多个方向上的深度信息，从而提供更准确的深度测量结果。求解采用几何思想，
2. 高效性：相对于传统的单波束测深技术，多波束测深模型在同一时间内可以获得更多的深度数据，因此更高效。这可以节省时间和资源，并提高数据采集的速度。
3. 高分辨率：多波束测深模型可以提供更高分辨率的深度图像，使用户能够更清晰地看到水下地形的细节。这对于水下地形的精确建模和分析非常重要。

7.2 模型的缺点

1. **数据处理和分析复杂性**: 多波束测深系统生成大量数据, 需要复杂的数据处理和分析技术来提取有用的信息。由于时间问题没有对其情况进行检验。

2. **限制深度范围**: 多波束测深系统的性能通常受到深度范围的限制, 无法在极深或极浅的水域中提供准确的深度测量。

7.3 模型的改进方法

由于本模型覆盖宽度的长度较坡度更大, 同时显得长度量级较大, 此时会生成大量数据, 需要借助python等处理软件数据, 在生成大量数据之前可借助Atlas Fansweep20 测深系统, 校准包括船横摇、延迟、纵摇以及偏航校准, 具体的校准方法见附件。^{[1][2]}

多波束系统的校准是在美国 Coastal Ocean graphic 仪器公司推出 HYPACK 系统的 HYSWEEP 软件中进行^[3]。利用该软件对采集的校准数据进行调整来获取测量时所需的最佳校准参数, 校准界面见图6。

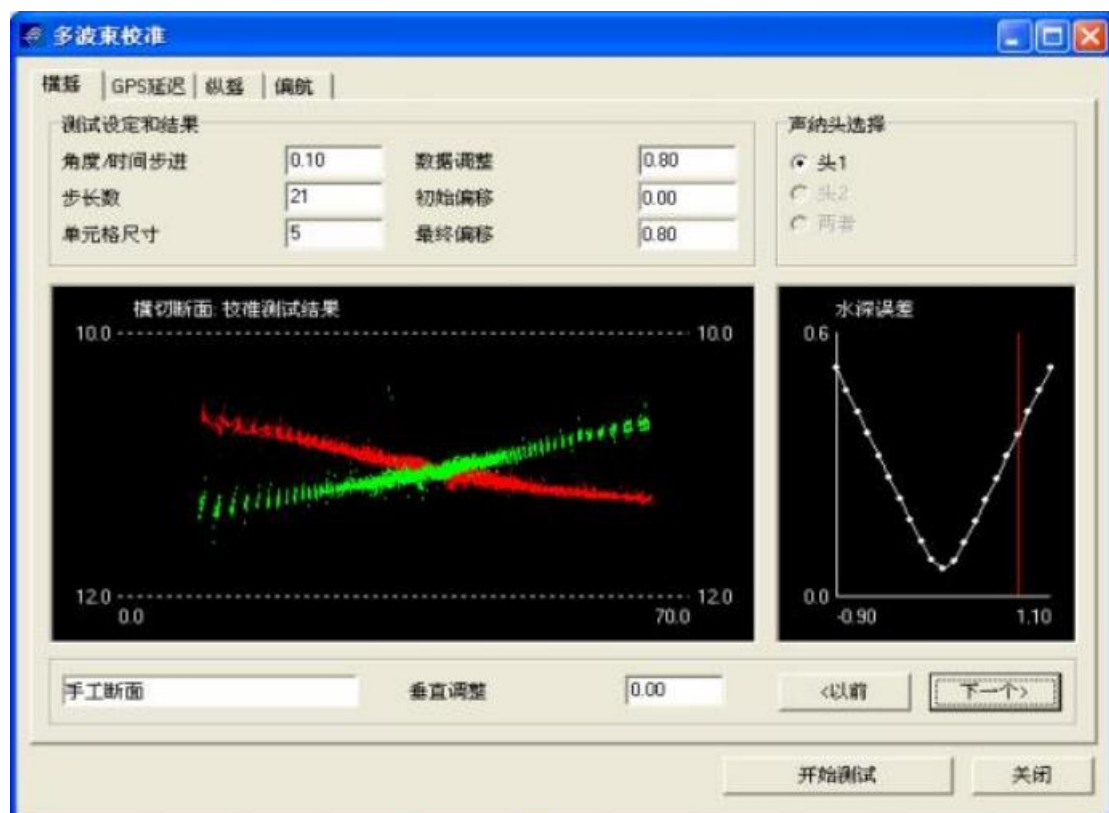


图6 HYSWEEP 软件中多波束测深系统校准界面

7.4 模型的推广

多波束测深覆盖宽度模型推广：

1. 基本假设：

海域是一个矩形区域，南北和东西方向有明确的长度。

海域中心点处的深度是已知的。

海底坡度是均匀的，并且在东西方向上有一定的角度。

多波束换能器的开角是已知的。

2. 覆盖宽度计算：

覆盖宽度的计算考虑了多波束换能器的开角和海底坡度。具体公式为

覆盖宽度：

$$W=(D+1852 \cdot x \cdot \tan 1.5^{\circ} \cdot |\sin \beta|) \cdot \left(\frac{\sin 60^{\circ}}{\sin (30^{\circ}-\alpha')} + \frac{\sin 60^{\circ}}{\sin (30^{\circ}+\alpha')} \right) \cdot \cos \alpha'$$

其中：

D=海域中心点处的深度

θ =多波束换能器的开角

α' =海底坡度

β =坡面法向量的投影与测线方向的夹角

3. 设计测线：

为了最大化覆盖并最小化测线的总长度，我们需要确定每条测线之间的最佳间隔。这个间隔要考虑到相邻条带之间的重叠率。具体公式为：

间隔=覆盖宽度* (1-r)

其中：r=期望的重叠率

注意事项：

1. 海域的坡度和多波束换能器的开角以及坡面法向量的投影与测线方向的夹角对覆盖宽度有直接影响，更大的开角或更小的坡度会增加覆盖宽度，而 β 角在0-90度使得覆盖宽度增大，90度到180度使得覆盖宽度减小。

2. 为了确保整个海域被完全覆盖，测线的设计必须考虑到海域的具体形状和大小。

3. 测线的总长度和所需测线数目是测量项目的关键指标，这两个指标需要在设计中进行权衡。

8. 参考文献

- [1] 宋志宏, 周刚炎, 周凤珍; 多波束测深系统及其比测试验[J]; 水文; 2002
(02): 41-44
- [2] 李建锋.Seabat8101多波束测深系统在水利工程中的应用. 中国水利国际合作与科技网
http://www.chinawater.net.cn/CWSNews_View.asp?CWSNewsID=12737
- [3] HYSWEEP User's Manual.Coastal Oceanographics. 1996.

9. 附录

附录清单:

附录一: 计算海水深度、覆盖宽度和重叠率随着测线随中心域的变化情况的python程序

附录二: 计算覆盖宽度随着船距中心点、测线方向夹角的变化情况的python程序

附录三: 找到一个最短的测量长度, 以便完全覆盖整个待测的海域。同时, 这个测量长度必须确保相邻的测量带之间的重叠率在10%到20%之间, 来得到测量的最优解的C++程序。

附录四: 基于简化模型根据题目指标设计得到的测线的python程序

附录五: Atlas Fansweep20 测深系统的具体校准方法表

附录六: 问题三模型求解的运行结果

附录一：

问题一程序：

| 程序编号 | T1—1 | 文件名称 | Result1 | 说明 | 海水深度、覆盖宽度、重叠率的计算结果 |
|--|------|------|---------|----|--------------------|
| <pre> #Python import math import pandas as pd def calculate_width(D, theta, alpha): #由正弦定理可知, 设W为覆盖宽度, alpha为坡度角, theta为开角, D为当前的海水深度 w1=(math.sin(math.radians(theta/2))/math.sin(math.radians(90+alpha-theta/2))) w2=(math.sin(math.radians(theta/2))/math.sin(math.radians(90-alpha-theta/2))) #抽象出覆盖宽度W和海水深度D的函数关系 W=D*(w1+w2)*math.cos(math.radians(alpha)) return W #防止分母为0的情况 def calculate_overlap(d, W): """计算重叠率""" return 0 if W == 0 else 1 - d / W # 数据初始化 distances = [-800, -600, -400, -200, 0, 200, 400, 600, 800] #depth是根据D-x*tan(alpha)计算的, 其中x为distances的距离 depths = [70 - math.tan(math.radians(1.5)) * d for d in distances] descriptions = [f'距离中心点{abs(d)}米' for d in distances] descriptions[4] = '中心点位置' data = { '测线距中心点处的距离/m': distances, '海水深度/m': depths, '描述': descriptions } result = pd.DataFrame(data) # 多波束测深系统参数 theta = 120 alpha = 1.5 # 使用Pandas的向量化计算来计算覆盖宽度 result['覆盖宽度/m'] = result['海水深度'] </pre> | | | | | |

```
/'m'].apply(calculate_width, args=(theta, alpha))

# 计算与前一条测线的重叠率
diff_values = result['测线距中心点处的距离/m'].diff()
result['与前一条测线的重叠率/%'] = ((1 - diff_values / result['覆盖宽度/m']) * 100).where(diff_values.notna(), other=None)

# 尝试保存到Excel，但如果失败则显示错误信息,成功则result1.xlsx与当前python文件在一个文件夹里
try:
    result.to_excel('result1.xlsx', index=False)
except Exception as e:
    print("保存到Excel失败:", e)

print(result)
```

| 程序编号 | T2-1 | 文件名称 | Result2 | 说明 覆盖宽度程序 |
|--|------|------|---------|-----------|
| <pre>import pandas as pd import math import numpy as np # 计算多波束测深的覆盖宽度 def calculate_width(D, theta, alpha, distance, angle): # 当前的坡度角alpha会随着angle 变化而变化 alpha_ = math.atan(math.tan(15 * math.pi / 1800) * abs(math.sin(angle))) # 当前的海水深度 D = D + K * distance * math.tan(15 * math.pi / 1800) * math.cos(angle) # 覆盖的坡面长度 W_D = D * (np.sin(math.pi / 3) / np.sin(math.pi / 6 + alpha_) + np.sin(math.pi / 3) / np.sin(math.pi / 6 - alpha_)) # 投影后的覆盖宽度 W = W_D * np.cos(alpha_) return W # 测量船距海域中心点处的距离和角度数据 distances = [0, 0.3, 0.6, 0.9, 1.2, 1.5, 1.8, 2.1] angles = [0, math.pi / 4, math.pi / 2, 3 * math.pi / 4, math.pi, math.pi * 5 / 4, math.pi * 3 / 2, math.pi * 7 / 4] # 多波束测深系统的开角和坡度和海里与米的倍率和海域中心点处的海水深度 theta = math.pi * 2 / 3 alpha = math.pi * 15 / 1800 K = 1852 D = 120 # 创建一个空的二维列表来存储覆盖宽度 widths = [] # 计算并填充宽度数据 for angle in angles: row = [calculate_width(D, theta, alpha, distance, angle) for distance in distances] widths.append(row)</pre> | | | | |

```
# 创建 DataFrame，不需要更改列 'angles' 的数据类型
result = pd.DataFrame(widths, columns=distances, index=angles)

# 将角度从弧度制转换为角度制
result.index = [math.degrees(angle) for angle in result.index]

# 设置 Pandas 显示选项，确保小数点均保留五位小数
pd.options.display.float_format = '{:.5f}'.format

# 打印计算结果
print(result)

# 尝试保存到Excel，但如果失败则显示错误信息，成功则result2.xlsx与当前python文件在一个文件夹里
try:
    result.to_excel('result2.xlsx', index=False)
except Exception as e:
    print("保存到Excel失败:", e)
```

| 程序编号 | T3—1 | 文件名称 | Result3 |
|---|------|------|---------|
| <pre> #include <iostream> #include <cmath> using namespace std; // 常量定义 const double theta = M_PI * 2 / 3; // θ 的定义，单位为弧度 const double alpha = M_PI * 15 / 1800; // α 的定义，单位为弧度 const double K = 1852; // K的常数值 const double D = 120; // D的常数值 const double C = sin(M_PI / 3) / (sin(M_PI / 6 + alpha)) + sin(M_PI / 3); const double x = 1 + 0.9 * C * sin(15.0 / 1800 * M_PI); // 临界值的计算 //206.9927是根据中心海域点海水深度计算得到的西边界上的海水深度 const double hmax = (206.9927 / sin(M_PI / 6)) * (sin(M_PI * 285 / 1800)); const double hmin = (2 * 1852 * tan(15.0 / 1800 * M_PI)) + 110 - (1852 * tan(M_PI / 6)); int main() { cout<<"南北走向为最佳的测线方向，其覆盖宽度最稳定"<<endl; double a = hmax; // 初始化a为hmax值 const double b = 206.9927 / (1.0 - sin(M_PI / 3) * sin(15.0 / 1800 * M_PI)); // 的计算公式 double length = 1; // 测量条数初始化为1（至少一条） // 当b与a之差大于1e-5时继续循环 while (b - a >= 1e-5) { // 根据给定条件调整length的值 while ((a / hmin - pow(x, length)) >= 1e-5) { length++; } // 输出当前海水深度值和测线length条数和测线总长度 length=length+1; cout << "h: " << a << " length: " << length << " Ls: " << length * D << endl; a += 0.1; // a增加0.1, 模拟海水深度在起点区间取到所有可能值 length = 1; // 重置length为1 } return 0; } </pre> | | | |

| 程序编号 | T4—1 | 文件名称 | | 说明 | |
|---|------|------|--|----|--|
| <pre> # 参数 w = 1 # 测量带的宽度，单位为海里 o = 0.2 * w # 重叠，即宽度的20% east_boundary = 4 # 待测海域的东边界，单位为海里 # 初始化起始位置和用于存储选定位置的列表 current_position = 0 selected_positions = [] # 贪婪算法来放置测量带 while current_position < east_boundary: selected_positions.append(current_position) current_position += w - o # 移动到下一个位置，考虑到重叠 # 计算相关指标 total_length_greedy = 5 * len(selected_positions) # 测量带的总长度 missed_percentage_greedy = 0 # 漏测海区的百分比 excess_overlap_length_greedy = 0 # 超过20%的重叠长度 # 打印结果 print("选定的测量带位置:", selected_positions) print("测量带的总长度:", total_length_greedy, "海里") print("漏测海区百分比:", missed_percentage_greedy, "%") print("超过20%的重叠长度:", excess_overlap_length_greedy, "海里") </pre> | | | | | |

附录五：多波束测深系统校准方法

| 序号 | 项目 | 地形要求 | 校准要求 | 航速要求 |
|----|----|--------|---------------------|--------------|
| 1 | 横摇 | 平坦地形 | 同一条线，反方向测 2 次 | 2 次同速、正常测量速度 |
| 2 | 延迟 | 斜坡或目标物 | 同一条线，同方向测 2 次 | 不同速、两次航速差一半 |
| 3 | 纵摇 | 斜坡或目标物 | 同一条线，反方向测 2 次 | 2 次同速、正常测量速度 |
| 4 | 偏航 | 目标物 | 2 条线、目标物两侧约 15m、同方向 | 2 次同速、正常测量速度 |

附录六：问题三模型求解的运行结果

```
Selection View Go Run ... ← → coder2022_12-2023(c-
... PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
ITORS
match_1_1.cpp h: 212.904 length: 37 Lsum(/N mile):74
match_1_2.cpp h: 213.004 length: 37 Lsum(/N mile):74
22_12-2023(C++) h: 213.104 length: 37 Lsum(/N mile):74
h: 213.204 length: 37 Lsum(/N mile):74
h: 213.304 length: 37 Lsum(/N mile):74
h: 213.404 length: 37 Lsum(/N mile):74
h: 213.504 length: 37 Lsum(/N mile):74
h: 213.604 length: 37 Lsum(/N mile):74
h: 213.704 length: 37 Lsum(/N mile):74
h: 213.804 length: 37 Lsum(/N mile):74
h: 213.904 length: 37 Lsum(/N mile):74
h: 214.004 length: 37 Lsum(/N mile):74
h: 214.104 length: 37 Lsum(/N mile):74
h: 214.204 length: 37 Lsum(/N mile):74
h: 214.304 length: 37 Lsum(/N mile):74
h: 214.404 length: 37 Lsum(/N mile):74
h: 214.504 length: 37 Lsum(/N mile):74
h: 214.604 length: 37 Lsum(/N mile):74
h: 214.704 length: 37 Lsum(/N mile):74
h: 214.804 length: 37 Lsum(/N mile):74
h: 214.904 length: 37 Lsum(/N mile):74
h: 215.004 length: 37 Lsum(/N mile):74
h: 215.104 length: 37 Lsum(/N mile):74
h: 215.204 length: 37 Lsum(/N mile):74
h: 215.304 length: 37 Lsum(/N mile):74
h: 215.404 length: 37 Lsum(/N mile):74
h: 215.504 length: 37 Lsum(/N mile):74
h: 215.604 length: 37 Lsum(/N mile):74
h: 215.704 length: 37 Lsum(/N mile):74
h: 215.804 length: 37 Lsum(/N mile):74
h: 215.904 length: 37 Lsum(/N mile):74
h: 216.004 length: 37 Lsum(/N mile):74
h: 216.104 length: 37 Lsum(/N mile):74
h: 216.204 length: 37 Lsum(/N mile):74
h: 216.304 length: 37 Lsum(/N mile):74
PS C:\Users\10251\Desktop\文件夹\coder2022_12-2023(c++)>
0 C/C++: cpp.exe 生成和调试活动文件 (coder2022_12-2023(c++))
```

```
View Go Run ... ← → coder2022_12-2023(c++)

... PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE

h: 216.304 length: 37 Lsum(/N mile):74
PS C:\Users\10251\Desktop\文件夹\coder2022_12-2023(c++)> cd "c:\Users\10251\Desktop\文件夹\coder2022_12-2023(c++)"
($?) { g++ match_1_2.cpp -o match_1_2 } ; if ($?) { .\match_1_2 }
南北走向为最佳的测线方向，其覆盖宽度最稳定
h: 197.604 length: 36 Lsum(/N mile):72
h: 197.704 length: 36 Lsum(/N mile):72
h: 197.804 length: 36 Lsum(/N mile):72
h: 197.904 length: 36 Lsum(/N mile):72
h: 198.004 length: 36 Lsum(/N mile):72
h: 198.104 length: 36 Lsum(/N mile):72
h: 198.204 length: 36 Lsum(/N mile):72
h: 198.304 length: 36 Lsum(/N mile):72
h: 198.404 length: 36 Lsum(/N mile):72
h: 198.504 length: 36 Lsum(/N mile):72
h: 198.604 length: 36 Lsum(/N mile):72
h: 198.704 length: 36 Lsum(/N mile):72
h: 198.804 length: 36 Lsum(/N mile):72
h: 198.904 length: 36 Lsum(/N mile):72
h: 199.004 length: 36 Lsum(/N mile):72
h: 199.104 length: 36 Lsum(/N mile):72
h: 199.204 length: 36 Lsum(/N mile):72
h: 199.304 length: 36 Lsum(/N mile):72
h: 199.404 length: 36 Lsum(/N mile):72
h: 199.504 length: 36 Lsum(/N mile):72
h: 199.604 length: 36 Lsum(/N mile):72
h: 199.704 length: 36 Lsum(/N mile):72
h: 199.804 length: 36 Lsum(/N mile):72
h: 199.904 length: 36 Lsum(/N mile):72
h: 200.004 length: 36 Lsum(/N mile):72
h: 200.104 length: 36 Lsum(/N mile):72
h: 200.204 length: 36 Lsum(/N mile):72
h: 200.304 length: 36 Lsum(/N mile):72
h: 200.404 length: 36 Lsum(/N mile):72
h: 200.504 length: 36 Lsum(/N mile):72
h: 200.604 length: 36 Lsum(/N mile):72
h: 200.704 length: 36 Lsum(/N mile):72
```