

多波束测深系统针对非平坦海域地形测量的模型研究

摘要

本文通过分层求解法、多波束测深系统设计、空间几何分析、多目标优化等方法，建立了非平坦海域地形测量的数学模型，解决了与多波束测线内容相关的问题，并应用于非平坦海域地形测深的优化设计。

针对问题一，对于覆盖宽度与重叠率模型的求解，为更精准地对海底坡面进行相关计算，首先分析水平面到坡面的变化，描述条带的覆盖宽度，进行了一个重叠率中两条相邻条带的覆盖宽度合理假设，并对其合理性进行一个简单的验证。随后通过对图像进行几何分析来确定数学模型相关参量 W 与 W' 的表达式，最终建立了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，最后本文基于给定的数据和建立的模型，并求解出了对应的海水深度、覆盖宽度和与前一条测线的重叠率。具体结果详见附录。

针对问题二，对于多测线方向的覆盖宽度的数学模型求解，首先建立以坡面法向在水平面上投影为 y 轴，竖直方向为 z 轴，水平面所在平面为 xy 平面，建立标准的坐标系。其次通过建立的坐标系使用向量的方法来求解测线在海底坡面的投影和条带在海底坡面所在直线与水平面的夹角，为后续的水深与侧面的高度变化提供一个随 α 与 β 的数学模型公式，由此问题转变为斜坡上的多波束测深模型，并参考问题一的模型，通过依次带入所需值的方式，求解出了相关的覆盖宽度。具体结果详见附录。

针对问题三，对于设计最短测量长度的优化设计问题，首先基于当测线方向走向符合等深线走向时，会得到尽量大的覆盖宽度[1]。同时，以每条相邻测线间的覆盖率固定为最小值10%布线，以尽可能少的测线覆盖整个待测海域，从而使得总测线长度的测量长度最短，得出沿着等深线走向行进最优的结论。经分析可得首条测线对应条带边界应恰好与海水边界重合。最后通过空间几何、迭代思想等手段，创建多个等式并联立求解，最终获得每条测线的位置，并计算出总测线长度为125,936m。

针对问题四，分析各指标间的关系，易知过大的重叠说明我们设计的测线可能有较多的冗余，因而使总测线长度变长。根据多目标优化算法的原理，决定尽最大可能先满足重叠率尽可能较小的要求，同时在大的覆盖面积占比的前提下尽可能的减少测线总长度。当覆盖区域覆盖了整个海域，或当测线的角度或位置达到了一定的上限时，迭代过程会结束。最终，在最优方案下，共需54条测线，总长度为442,650米。未测量海区占总待测海域面积的比例为17.72%，漏测面积占比为3.71%，而重叠率超过20%的部分的总长度为0。

本模型建立中运用 Python（图像绘制，方程求解，最优规划）、Matlab（方程求解，最优规划）、R（图像绘制）、Excel（图像绘制）等工具，从而使建模过程顺利进行，使所建模型更加精简。

关键词：计算求解模型、分层求解法、多波束测深系统设计、空间几何分析、多目标优化

一、 问题重述

多波束测深技术是通过在与航线垂直的平面内发出大量波束，利用声波在界面反射的特点，接收返回声波从而测得一定宽度的全覆盖水深条带。在测量中，两条测线的间距太小会导致其条带具有一定程度的重叠率。当重叠率在10% ~ 20%时，可以保证测量具有便利性和数据的完整性。

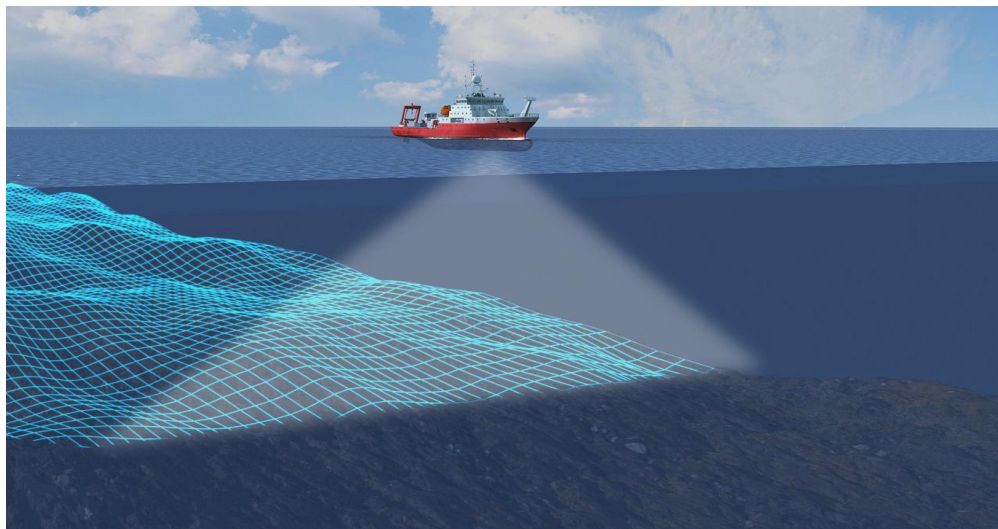


图 2-1: 非平坦海域地形多波束测深示意图

由于在真实的海底地形中起伏相对较大，使得重叠率在个别区域出现不同劣势情况，从而影响测量的质量和效率，因此提出以下问题：

问题一：构建一个在倾斜海平面上多波束测深系统的覆盖宽度和相邻测量带的重叠率的数学模型。考虑多波束换能器的开角为 120° ，坡度为 1.5° ，并且在海域的中心位置的海水深度为 $70m$ ，使用前述模型计算表 1 中列出的位置指标，并将结果以表 1 的形式展示在正文中，同时保存到 `result1.xlsx` 文件中。

问题二：构建一个多波束测深覆盖宽度的数学模型，针对与一个矩形待测海域，计算在测线方向与海底坡面的法向在水平面上投影的夹角为不同 β 下各测线覆盖宽度。考虑多波束换能器的开角为 120° ，坡度为 1.5° ，并且在海域的中心位置的海水深度为 $120m$ ，使用前述模型计算表 2 中列出的覆盖宽度指标，并将结果以表 2 的形式展示在正文中，同时保存到 `result1.xlsx` 文件中。

问题三：在一个南北延伸2海里、东西跨度4海里的矩形海域中，海域的中心深度为 $110m$ ，且海底从西到东逐渐变浅，坡度达到 1.5° 。考虑使用开角为 120° 的多波束换能器。要规划一条测量路径，使其长度最短但能够全面覆盖这片海域，并确保相邻测量带间的重叠率在10%到20%之间。

问题四：利用附件中多年前的单波束测线水深数据，构建一个多波束测线模型。在设计布线时，要尽可能按以下要求进行路线优化：① 测量线要覆盖整个待测海域的条带，以尽可能完整地测量该区域；② 将相邻条带之间的重叠率尽可能控制在 20% 以下，以确

保数据的准确性；③ 总的测量线长度要尽可能短。得出最优线路后，给出下面的指标：

① 总测量线的长度；② 未测量到的海域所占总待测海域面积的百分比；③ 在重叠区域中，重叠率超过 20% 部分的总长度。

二、 问题分析

2.1 问题一的分析

问题一是在题目所给予的水平覆盖宽度、侧线间距与重叠区域的基础上，将测深环境替换到了一个坡面上。给出的坡面角度较小可近似为地形平坦。但考虑到由于坡面的变化导致水深 D 发生改变，从而导致条带的覆盖宽度随着测线的位置不同而不同。这对其本身的覆盖宽度计算无较大影响，但是相邻条带间计算重叠率时，条带的覆盖宽度 W 的选取变得模糊。对此，假设一个合理的新相邻条带间计算时的覆盖宽度 W' ，并对这个 W' 进行简单的验证，说明其本身的一定合理性。最终通过 W' 所形成的重叠率公式与几何分析，得到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，并计算相关位置的指标值。

2.2 问题二的分析

问题二是将问题一的特殊方向的测线为变成了未知方向的测线，并要求提供一定的计算结果。在中心点处海水深度的情况下，通过获得海底坡面高度随行驶距离的变化率 K_1 （即测线在海底坡面的投影 L_1 与水平面的夹角 $\alpha_{船}$ ），得到行进过程中所有的中心点处海水深度，并利用海水坡度 $\alpha_{测}$ （与测线方向垂直的平面和海底坡面的交线构成一条与水平面夹角）转化为问题一的数学模型并带入海水深度等参数求解。最终，我们通过建立坐标系使用向量方法求得的 $\alpha_{船}$ 与 $\alpha_{测}$ 和问题一的数学模型，并计算相关的覆盖宽度。

2.3 问题三的分析

问题三是设计一组满足要求的合理测线，其海水深度、剖面、海域形状已经给出，仅需确定测线方向与测线的起始位置。首先，通过查阅资料，得到当测线方向走向符合等深线走向时，会得到尽量大的测线范围^[1]。由于更大的测线范围，可使总测线的测量长度最短。由此，确定了最优的测线行进方向。然后，通过测线起始位置在内部、最西侧、最东侧（最东侧此处是指测线所造成的条带东侧边界与矩形海域边界重合，最西侧同理），这三种情况分析。经过简单分析可知，其一定存在一个使测量长度最短的初始测线在最西或最东侧。通过多等式联立求解和迭代计算，求得每一条侧线的具体位置，得到测线初始位置在最西侧和最东侧的总测线的测量长度。最终经过两者比较得到最小的总测线的测量长度。

2.4 问题四的分析

获取附件数据后，绘制海底地形图。观察海底地形，可以发现海底地形以对角线方向为最低处，向两侧逐渐上升，其中(0,0)点处半侧坡脚更大，(0,0)点也是海底地形最高处。可以看到两侧坡度相对均匀，且多年前的单波束测线数据可能会有一定偏差，因此假设两侧斜坡坡度大致恒定。考率到题目要考量三个指标：① 测线的总长度；② 漏测海区占总

待测海域面积的百分比；③ 在重叠区域中，重叠率超过 20% 部分的总长度。显示是个多目标优化问题。分析各指标间的关系，易知过大的重叠说明我们设计的测线可能有较多的冗余，因而使总测线长度变长。根据多目标优化算法的原理，决定尽最大可能先满足重叠率尽可能较小（但不要低于10%）的要求，具体的：以10% ~ 20%的重复率为必须参数代入程序，同时在保证极大的覆盖面积占比的前提下尽可能的减少测线总长度。以此为基本思想，设计 Matlab 程序代码，进行迭代计算，得出所需结果。

三、 模型假设

1. 假设声波在海水中以恒定速度传播，不考虑声波在水中的折射；
2. 假设数学模型中海底坡面坡度角特别小可以视作海底地形平坦；
3. 假设在其从平面到为坡面的转换过程中重叠率计算公式中 W 变为 W' ，由左右条带的部分覆盖宽度共同组成；
4. 假设多年前的单波束测试数据足够准确，也不考虑若干年后地形存在的微小变化；
5. 假设在两侧坡度相对均匀的情况下，斜坡坡度大致恒定。

四、 符号说明

序号	符号	符号说明
1	W	多波束测深条带的覆盖宽度，即条带水平方向的长度
2	W'	两条测线之间条带的部分的总覆盖宽度，由左右条带的部分宽度共同组成
3	w_1	条带靠测线左边的覆盖宽度。 w_1' 指上一个条带靠测线左边的覆盖宽度
4	w_2	条带靠测线右边的覆盖宽度
5	d	两条测线间的水平距离
6	D	当前测线点位的海水深度
7	D_0	海域中心点处的海水深度
8	α	海域坡度。 $\alpha_{船}$ 为测线方向的海域坡度。 $\alpha_{测}$ 为与测线方向垂直的海域坡度
9	β	坡面法向在水平面上的投影与测线方向的夹角
10	θ	多波束换能器的开角

五、 模型建立与求解

5.1 问题一：探测的覆盖宽度与相邻条带间重叠率的数学模型和相关位置指标值的计算

5.1.1 背景知识

多波束测深条带的覆盖宽度 W 随换能器开角 θ 和水深 D 的变化而变化。若测线相互平行且海底地形平坦，则相邻条带之间的重叠率定义为

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

式中 d 为相邻两条测线的间距， W 为条带的覆盖宽度。

5.1.2 平面到微斜面过程中重叠率公式内条带的覆盖宽度的改变

由于相邻条带之间的重叠率公式以海水地形平坦为条件，模型将假设所有处理的海底坡面坡度特别小，可近似为海水地形平坦。在平面到微斜面的转化，本文将考虑重叠率公式(1)中条带的覆盖宽度 W 的改变使其变为 W' ，并对其作假设，证明其一定程度的合理性。

以测线为分隔线，将两边的条带分为左右不同的部分，其中当前左条带覆盖宽度为 w_1 ，当前右条带覆盖宽度为 w_2 ，而其前一条测线对应分别表示为 w_1' 与 w_2' （具体表示如图 6-1 所示）。

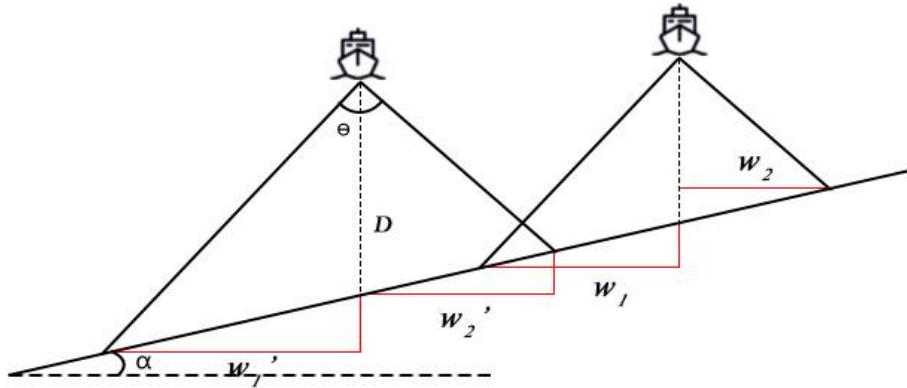


图 6-1：两相邻测线的重叠条带示意图

本文考虑将在重叠侧的覆盖宽度的 w_2' 与 w_1 相加得到的一个新 W' 假设作为重叠率公式中的覆盖宽度 W 。根据重叠率的常识性理解，结合两条相邻测线各自的情况来计算它们之间的覆盖率更为合理，可以发现这一点与问题一所要得到的结果之一（“与前一条测线的重叠率”）更为契合。 W' 的数学公式如下：

$$W' = w_2' + w_1 \quad (2)$$

其中， w_2' 为前一条测线右条带覆盖宽度， w_1 为当前测线左条带覆盖宽度。

下面我们将对假设的 W' 的合理性进行一个讨论分析。首先我们利用几何公式求解出 W' （求解过程详见模型的建立部分）

$$W'(x) = (D - x' \cdot \tan\alpha) \left(\frac{\sin\left(\frac{\theta}{2}\right)}{\sin\left(\frac{\pi}{2} - \frac{\theta}{2} + \alpha\right)} \right) \cdot \cos\alpha + (D - x \cdot \tan\alpha) \left(\frac{\sin\left(\frac{\theta}{2}\right)}{\sin\left(\frac{\pi}{2} - \frac{\theta}{2} - \alpha\right)} \right) \cdot \cos\alpha \quad (3)$$

其中， D 为中心点处的海水深度， x 为测线距测量原点的水平距离， α 为海底坡面的坡度， θ 为多波束换能器的开角。

在变化趋势上，当 α 、 θ 、 D 保持不变时，随着 x 的增加， W' 会不断减小，使得重叠率 η 也随之减小。其与实际情况相吻合，有一致方向的变化趋势。在与前一测线 $W_{前}$ 和当前测线 W 所测得的重叠率 η 的比较中，我们可以得到下图 6-2：

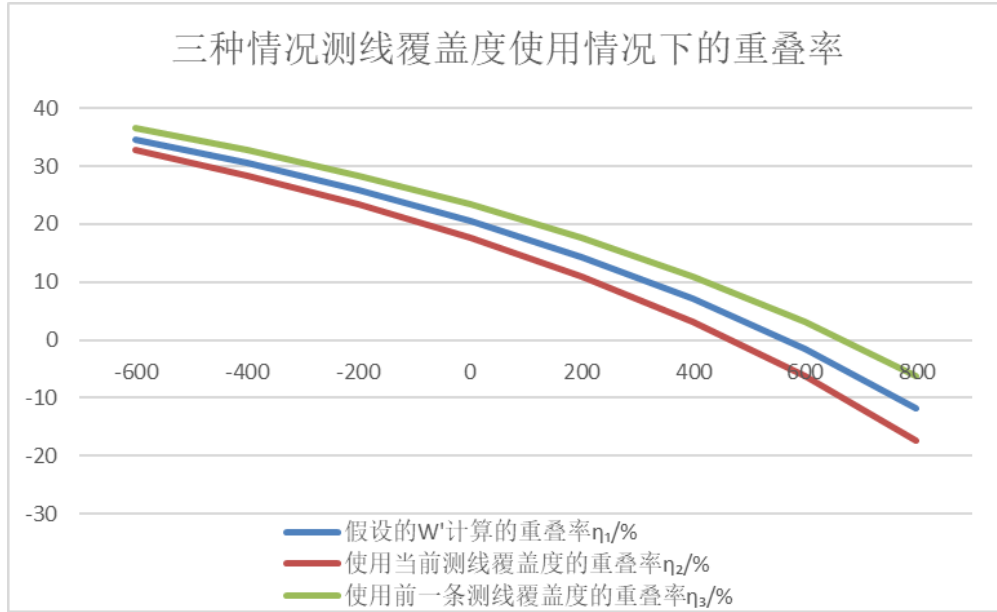


图 6-2：三种情况测线覆盖度使用情况下的重叠率

在上图 6-2 中，我们所假设的 W' 所计算出的重叠率处于前一测线 $W_{前}$ 和当前测线 W 所计算出的重叠率中间，能够无太大偏向性的描述两条带之间的重叠率。而且从逻辑上来讲，描述两条带间的重叠率肯定需要两条带重叠部分的参与，而不是任意单独的覆盖宽度 W 。由此，本文认为假设的 W' 在计算重叠率时具有较好的合理性。

5.1.3 模型的建立

如下图 6-3 所示，通过三角形的正弦定理，可将模型切割为两个三角形：

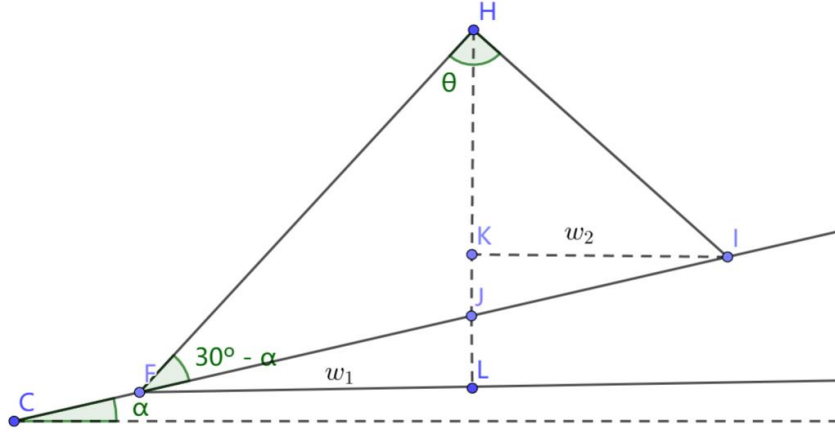


图 6-3：针对单测线条带宽度的几何分析图

由此可计算出当前水深 D 计算 测线覆盖左条带 w_1 和右条带 w_2 :

$$\begin{cases} w_1 = D \times \frac{\sin\left(\frac{\theta}{2}\right)}{\sin\left(90^\circ - \frac{\theta}{2} - \alpha\right)} \times \cos\alpha \\ w_2 = D \times \frac{\sin\left(\frac{\theta}{2}\right)}{\sin\left(90^\circ - \frac{\theta}{2} + \alpha\right)} \times \cos\alpha \end{cases} \quad (4)$$

而当前测线覆盖宽度即为这前后两端相加：

$$w = w_1 + w_2 \quad (5)$$

随后根据梯形的几何关系，结合其他测线距中心测线的距离 x 计算出各测线处的水深 $D(x)$:

$$D(x) = D - x \tan\alpha = 70 - x \cdot \tan 1.5^\circ \quad (6)$$

依次按照公式（4）带入公式（2）得到新的关于 w_1 与 w_2 的公式将其带入公式（3）可得到每处测线的覆盖宽度 $W(x)$ 的计算公式：

$$W(x) = (D - x \cdot \tan\alpha) \left(\frac{\sin\left(\frac{\theta}{2}\right)}{\sin\left(90^\circ - \frac{\theta}{2} - \alpha\right)} + \frac{\sin\left(\frac{\theta}{2}\right)}{\sin\left(90^\circ - \frac{\theta}{2} + \alpha\right)} \right) \cdot \cos\alpha \quad (7)$$

在计算覆盖率时，出于更常理化的推断，我们使用上面假设的 W' 来代替原覆盖宽度 W 进行覆盖率 η 的计算， W' 由 w_2' (前一条测线的右条带)和 w_1 (当前测线的左条带)两部分组成：

$$W'(x) = (D - x' \cdot \tan\alpha) \left(\frac{\sin\left(\frac{\theta}{2}\right)}{\sin\left(\frac{\pi}{2} - \frac{\theta}{2} + \alpha\right)} \right) \cdot \cos\alpha + (D - x \cdot \tan\alpha) \left(\frac{\sin\left(\frac{\theta}{2}\right)}{\sin\left(\frac{\pi}{2} - \frac{\theta}{2} - \alpha\right)} \right) \cdot \cos\alpha \quad (8)$$

公式中的 x' 代表上一条测线距离中心测线的距离。

最后将公式（5）带入公式（1）可得 W' 的计算公式，以下将公式简化展示为：

$$\eta = 1 - \frac{d}{W'} \quad (9)$$

5.1.4 相关位置指标值的求解

模型求解主要借助 Python 进行运算，将题目给定条件录入 Python 程序（具体程序详见附录程序一）：多波束换能器的开角为 120° ，坡度为 1.5° ，海域中心点处的海水深度为 $70m$ ，计算结果如下图表 6-1 所示。

表 6-1：问题一的计算结果

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	90.94874	85.71155	80.47437	75.23718	70	64.76282	59.52563	54.28845	49.05126
覆盖宽度/m	315.7051	297.5256	279.346	261.1665	242.987	224.8074	206.6279	188.4484	170.2688
与前一条测线的重叠率/%	—	34.68387	30.56123	25.8831	20.52911	14.34137	7.108689	-1.45803	-11.7654

不同海水深度、测线位置与条带宽度、重叠区域的表示见下图 6-4：

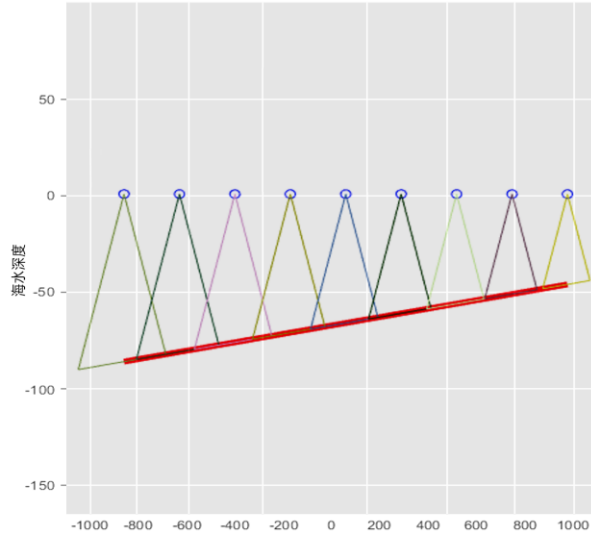


图 6-4: 测线位置与条带宽度、重叠区域示意图

5.2 问题二的模型建立与求解

5.2.1 模型的建立

首先我们需要获得在指定测线方向夹角、指定测量船距海域中心点处的距离的条件下的位置，其海水深度值。

如图下 6-5 所示，假定测线方向夹角为某个固定值 β ，沿对应测线方向移动一段距离。设沿海底坡面上升的水平距离为 x ，则易得垂直方向上升高度为 $x \cdot \tan\alpha$ ，移动轨迹的水平投影为 $\frac{x}{\sin(\beta - \frac{\pi}{2})}$ 。此时三角关系确立，可得到移动方向海底上升的相关参数：

$$\begin{aligned} \tan\alpha_{\text{船}} &= \frac{x \cdot \tan\alpha}{\frac{x}{\cos(\beta - \frac{\pi}{2})}} \\ &= \tan\alpha \cdot \cos(\beta - \frac{\pi}{2}) \end{aligned} \quad (10)$$

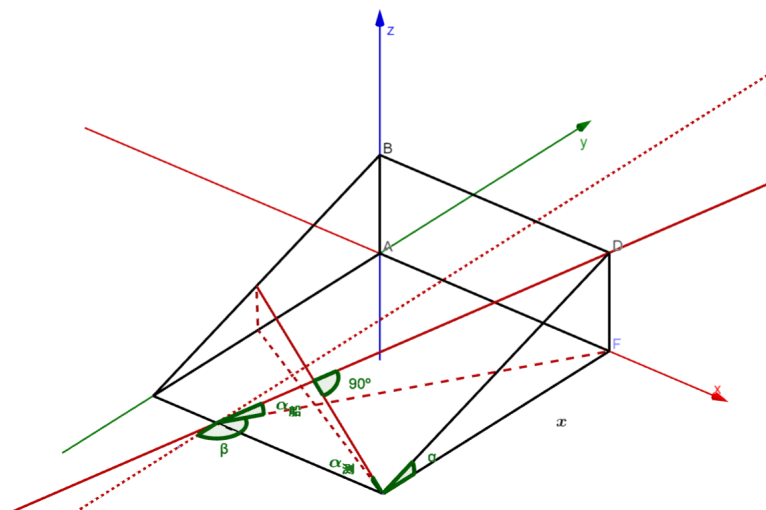


图 6-5: 移动方向海底上升计算示意图

而指定测线方向夹角、指定测量船距海域中心点处的距离的条件下的位置，与原海域中心点处的位置形成了如下图 6-6 的梯形。

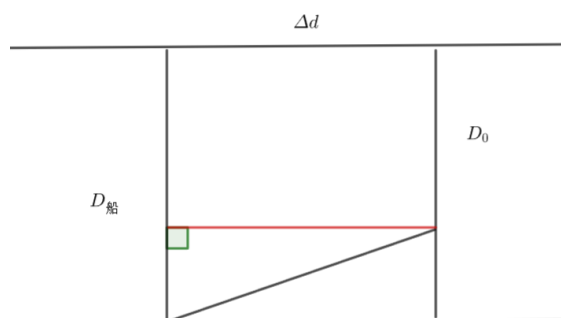


图 6-6: 指定位置与原海域中心点处关系示意图

由已知信息易知不同的指定测量船点位距海域中心点处的距离为已知值，则数值方向上，海底的高度变化为 $\Delta d \cdot \tan \alpha_{\text{船}}$ 。由此得到当前测线任意位置的海水深度为：

$$D_{\text{船}} = D_0 - \Delta d \cdot \tan \alpha_{\text{船}} \quad (11)$$

由此我们具备了求取覆盖宽度所需的基本数据。

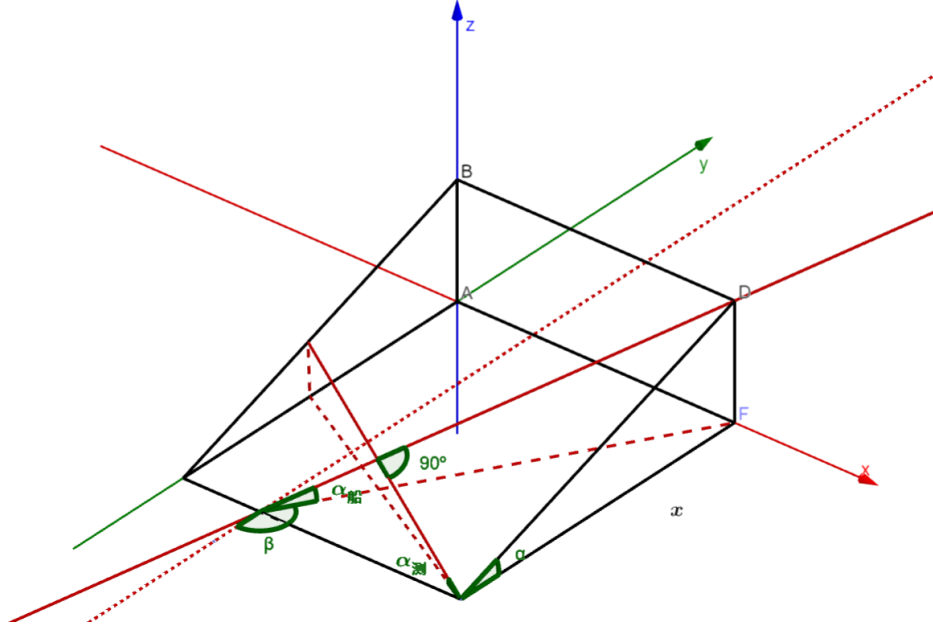


图 6-7: 覆盖宽度求取示意图

如上图 6-7 所示，我们针对船位于海海底附近的一块区域建立了笛卡尔坐标系并进行分析，其中 x 轴为海底坡面的法向量的水平向量的分向量， z 轴为竖直方向向量。测线向量在海底坡面的投影所形成的向量为：

$$\vec{a}_{\text{船}} = \left(-1, \frac{1}{\tan\left(\beta - \frac{\pi}{2}\right)}, \tan\alpha \right) \quad (12)$$

由题目信息所述，单个条带在海底坡面所在直线必定与测线或测线向量在海底坡面的投影所形成的向量垂直。假设这条直线的代表的向量由海平面的其他两条不平行的子向量组成：

$$\begin{cases} \vec{m} = (0, 1, 0) \\ \vec{n} = (-1, 0, \tan\alpha) \end{cases} \quad (13)$$

即使用这两条不平行的子向量代表：

$$\begin{aligned} \vec{a}_{\text{测}} &= a \cdot \vec{m} + b \cdot \vec{n} \\ &= (-b, a, b \cdot \tan\alpha) \end{aligned} \quad (14)$$

由垂直可得两向量点乘值为 0，即得到 a 与 b 之间的关系：

$$\begin{aligned} (-b, a, b \tan\alpha) \cdot \left(-1, \frac{1}{\tan\left(\beta - \frac{\pi}{2}\right)}, \tan\alpha \right) &= 0 \\ a &= -b \times (1 + \tan^2\alpha) \cdot \tan\left(\beta - \frac{\pi}{2}\right) \end{aligned}$$

单个条带在海底坡面所在直线，也即是多波线深测覆盖向量为：

$$\begin{aligned}\vec{a}_{测} &= (-b, a, b \cdot \tan\alpha) \\ &= \left(-b, -b \times (1 + \tan^2\alpha) \cdot \tan\left(\beta - \frac{\pi}{2}\right), b \cdot \tan\alpha\right) \\ &\rightarrow \left(1, (1 + \tan^2\alpha) \cdot \tan\left(\beta - \frac{\pi}{2}\right), b \cdot \tan\alpha\right)\end{aligned}\quad (16)$$

最终得到单个条带在海底坡面所在直线与水平面形成的夹角 $\sin\alpha_{测}$ 的值：

$$\begin{aligned}\sin\alpha_{测} &= \cos < \vec{a}_{测}, \vec{z} > \\ &= \frac{\vec{a}_{测} \cdot \vec{z}}{|\vec{a}_{测}| \times |\vec{z}|} \\ &= \frac{-\tan\alpha}{\sqrt{1 + (1 + \tan^2\alpha) \cdot \tan\left(\beta - \frac{\pi}{2}\right) + \tan^2\alpha}}\end{aligned}\quad (17)$$

由此得到 $\alpha_{测} = \arcsin(\sin\alpha_{测})$ ，并将所有得到的结果带入问题一的 W 求解公式可得覆盖宽度的值。

5.2.2 多波束测深的覆盖宽度的求解

通过空间向量分析的方法，利用问题二的数据得出问题一中模型所需关键数据（测线中心水深 D 和相对坡脚 α ），最终利用问题一模型，通过 Python 软件编程（源程序见附录 5）求解。结果如下表 6-2 所示：

表 6-2：问题 2 的计算结果

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 / $^{\circ}$	0	415.6922	466.0911	516.4899	566.8888	617.2876	667.6865	718.0854	768.4842
	45	415.6923	451.3297	486.9671	522.6045	558.2419	593.8793	629.5166	665.154
	90	415.6925	415.6925	415.6925	415.6925	415.6925	415.6925	415.6925	415.6925
	135	415.6923	380.0549	344.4175	308.7802	273.1428	237.5054	201.868	166.2306
	180	415.6922	365.2933	314.8945	264.4956	214.0967	163.6979	113.299	62.90017
	225	415.6923	380.0549	344.4175	308.7802	273.1428	237.5054	201.868	166.2306
	270	415.6925	415.6925	415.6925	415.6925	415.6925	415.6925	415.6925	415.6925
	315	415.6923	451.3297	486.9671	522.6045	558.2419	593.8793	629.5166	665.154

下图 6-8 为使用 z -极坐标表示的覆盖宽度随测线方向夹角和测量船距海域中心点处的距离变化的图示。

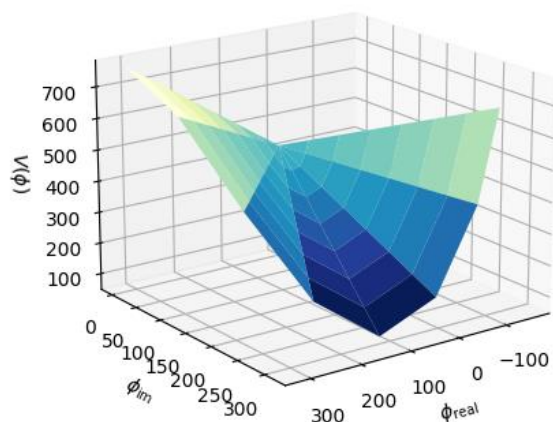


图 6-8: z -极坐标表示的覆盖宽度随测线方向夹角和测量船距海域中心点处的距离变化示意图

5.3 问题三：设计一组满足要求的合理测线

5.3.1 矩形海域中测线走向的确定和测线初始位置分析

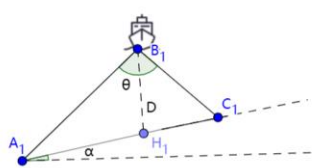
问题三的目的是设计一组满足要求的合理测线，其海水深度、剖面、海域形状已经给出，仅需确定测线方向、测线的起始位置和在计算 d 时使覆盖率在最低10%恒定。

5.3.1.1 测线方向的分析（以下分析中测线方向为唯一改变量）

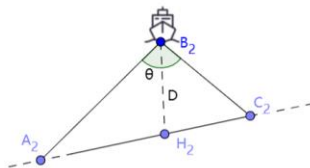
通过参考文献 1 中原文内容“区域内的测线应为彼此平行关系，且走向也要符合航道底部等深线走向，这样布置可尽量扩大测线范围，提高覆盖率^[1]”。由此原文可知，当测线的走向在符合等深线走向的情况下，测线范围最大，覆盖宽度最大，总测线的测量长度最短。得到最优的测线行进方向（即：沿着等深线走向行进）

5.3.1.2 测线的起始位置分析（以下分析中起始位置为唯一改变量）

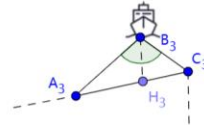
测线的起始位置仅有三种可能最西侧测线条带边界与矩形海域最西侧重合、内部、最东侧测线条带边界与矩形海域最东侧重合，情况分别如下图 6-9-1、图 6-9-2、图 6-9-3 所示：



图**.1: 测线起始位置在矩形海域最西测重合情况



图**.2: 测线起始位置在矩形海域中间的情况



图**.3: 测线起始位置在矩形海域最右侧重合情况

图 6-9: 从左到右依次为图 6-9-1、6-9-2、6-9-3

当测线从中间开始向两边延伸，当测线总测量长度最短时，仅有三种探测情况的可能：

- ① 存在两条测线条带边界恰好分别在矩形海域最西和最东侧（最西与最东分别满足上图 6-9-1 和 6-9-3）
- ② 存在某一条测线条带边界恰好在矩形海域最东或最西侧（最西与最东满足上图 6-9-1 和 6-9-3 中其中一个）
- ③ 无条测线条带边界恰好在矩形海域最东和最西侧（如下图 6-10 所示）。

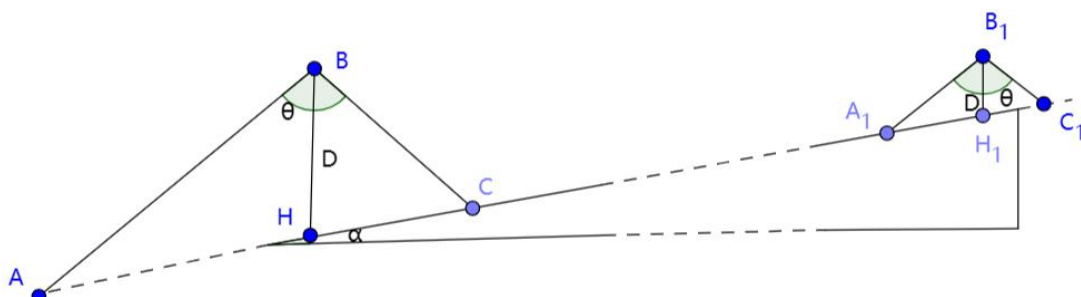


图 6-10: 无条测线条带边界恰好在矩形海域最东和最西侧示意图

当出现情况 ③ 时，据上图 6-10，可以明显看出，适当平移距矩形海域最东或最西侧最近的条带位置可使探测情况变为情况②或情况①。所以必存在一条使总测线的测量长度最短的情况（即：情况二）。这种情况下也就是上述起始位置为图 6-9-1 与图 6-9-3。我们即可通过模型建立后依次计算两者情况并比较总测线长度大小，由此确定一个使总测线的测量长度最短，最优的初始位置中其中一个最优初始位置。

覆盖率可以比较明显的判断出，当覆盖率取值为要求范围内最低值10%时，其可使总测线的测量长度最短，这里不再详细赘述。

5.3.2 模型的建立

首先，我们需要获得首个测线与矩形海域边界的距离和海水深度。该测线对应条带右侧边界应恰好与海水边界重合，故这需要通过最大或最小海水深度辅助计算得出，如图 6-11 所示：

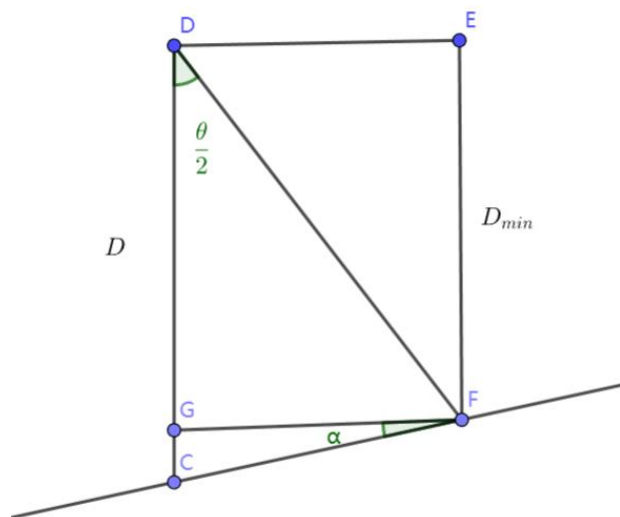


图 6-11: 首个测线与矩形海域边界的距离参数计算示意图

由此可得首个测线与矩形海域边界的距离 d 和其海水深度 D :

$$d = D_{edge} \cdot \tan\left(\frac{\theta}{2}\right)$$

$$D = D_{edge} + d \cdot \tan\alpha$$

除了首个测线外，其余测线均满足以下等式。

假定我们已知前一个测线的水深为 D_1 。设当前测线的水深为 D_2 ，这两个测线间的水平距离为 d 。则易得：

$$D_2 = D_1 + d \cdot \tan\alpha \quad (19)$$

如图 6-12 所示，为使得可完全覆盖整个海域测线最短，则需要单个测线的数量最小，这要求相邻条带之间的重叠率 η 最低。

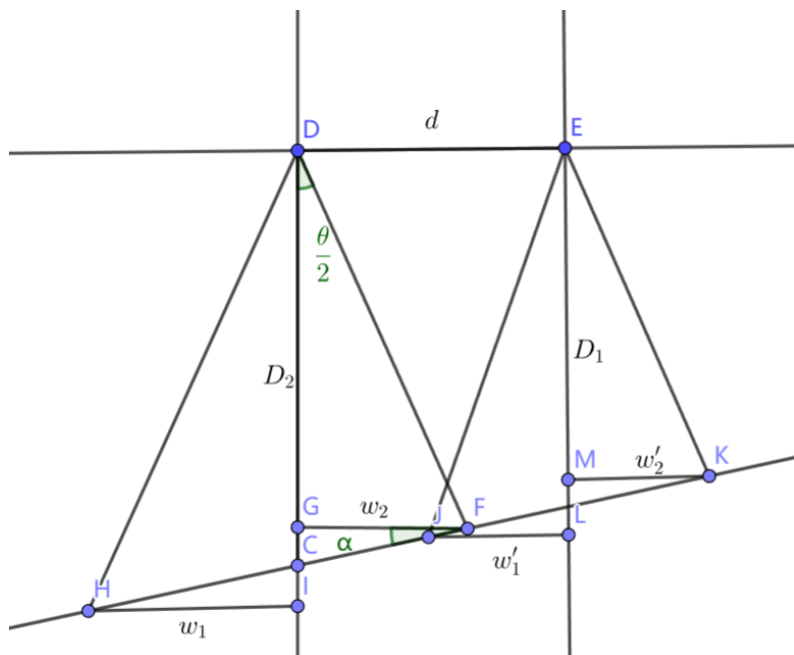


图 6-12: 相邻条带之间的重叠率计算示意图

由问题一的分析可知前一个测线条带的左半部分水平方向长度 w_1' :

$$w_1' = D_1 \times \frac{\sin\left(\frac{\theta}{2}\right)}{\sin\left(90^\circ - \frac{\theta}{2} - \alpha\right)} \times \cos\alpha \quad (20)$$

同理亦可得当前测线条带的右半部分水平方向长度 w_2 :

$$\begin{aligned} w_2 &= D_2 \times \frac{\sin\left(\frac{\theta}{2}\right)}{\sin\left(90^\circ - \frac{\theta}{2} + \alpha\right)} \times \cos\alpha \\ &= (D_1 + d \cdot \tan\alpha) \times \frac{\sin\left(\frac{\theta}{2}\right)}{\sin\left(90^\circ - \frac{\theta}{2} + \alpha\right)} \times \cos\alpha \end{aligned} \quad (21)$$

根据题目要求, 相邻条带之间的重叠率 η 最低为10%, 由此可得到关系式:

$$\begin{aligned} \eta &= 1 - \frac{d}{w_2 - w_1'} = 10\% \\ \Rightarrow d &= 0.9 \times w_1' + 0.9 \times w_2 \end{aligned} \quad (22)$$

至此可利用多等式联立求解思想, 联立上述方程(6)、(7)、(8)、(9), 带入求解, 得到:

$$d = 0.9 \times D_1 \times \frac{\sin\left(\frac{\theta}{2}\right)}{\sin\left(90^\circ - \frac{\theta}{2} - \alpha\right)} \times \cos\alpha + 0.9 \times (D_1 + d \cdot \tan\alpha) \times \frac{\sin\left(\frac{\theta}{2}\right)}{\sin\left(90^\circ - \frac{\theta}{2} + \alpha\right)} \times \cos\alpha \quad (23)$$

最终化简得到：

$$d = \frac{0.9 \times D_1 \times \frac{\sin\left(\frac{\theta}{2}\right)}{\sin\left(90^\circ - \frac{\theta}{2} - \alpha\right)} \times \cos\alpha + 0.9 \times D_1 \times \frac{\sin\left(\frac{\theta}{2}\right)}{\sin\left(90^\circ - \frac{\theta}{2} + \alpha\right)} \times \cos\alpha}{1 - \tan\alpha \cdot \frac{\sin\left(\frac{\theta}{2}\right)}{\sin\left(90^\circ - \frac{\theta}{2} + \alpha\right)} \times \cos\alpha} \quad (24)$$

随后利用迭代计算方法，依次求取每条测线之间的水平距离 d ，从而求得每一条侧线的具体位置。这些测线即符合要求的最优、最短测线组。

5.3.3 最优可完全覆盖整个海域测线的求解

最优路线以沿等深线方向布线（测线船沿南北方向边行驶），经 Python 程序的分别计算，发现在给定条件下，无论从水深最大处还是从水深最小处开始布线，得到的最小测线数均为34条，将测线数与待测海域南北长相乘即可得到可覆盖全待测海域的最短总测量长度为125,936m。具体测线位置数据如下表格三所示。详细数据见附录。

表格 6-3：一组符合条件的测线布设位置

轮次	测线距中心点处的距离/m	轮次	测线距中心点处的距离/m	轮次	测线距中心点处的距离/m
1	3681.470615	13	2810.679426	25	479.5842844
2	3637.064371	14	2691.804567	26	161.3580437
3	3588.860578	15	2562.763727	27	-184.0823762
4	3536.534478	16	2422.687529	28	-559.0642866
5	3479.733536	17	2270.632248	29	-966.1140266
6	3418.075072	18	2105.573452	30	-1407.973984
7	3351.143679	19	1926.399102	31	-1887.62107
8	3278.488425	20	1731.90206	32	-2408.286778
9	3199.619814	21	1520.771953	33	-2973.478951
10	3114.00649	22	1291.586351	34	-3587.00542
11	3021.071657	23	1042.801177		
12	2920.189192	24	772.7403092		

这一组符合条件的测线距中心点处的距离与海水深度、条带宽度展示如下图 6-13 所示：

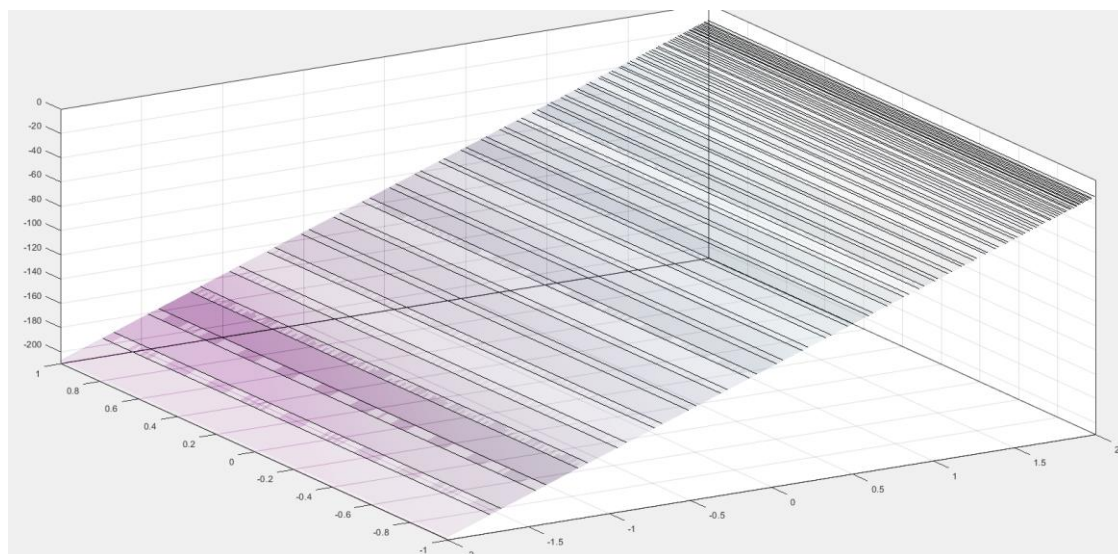


图 6-13: 测线距中心点处的距离与海水深度、条带示意图

5.4 问题四的模型建立与求解

5.4.1 模型的建立

1. 获取海水深度数据：首先，我们要通过 Matlab 读取附件中提供多年前单波束测线水深数据，并绘制海底地形三维图，方便模型的构建和分析，具体结果如下图 6-14 所示（附件内容的海底地形三维图）：

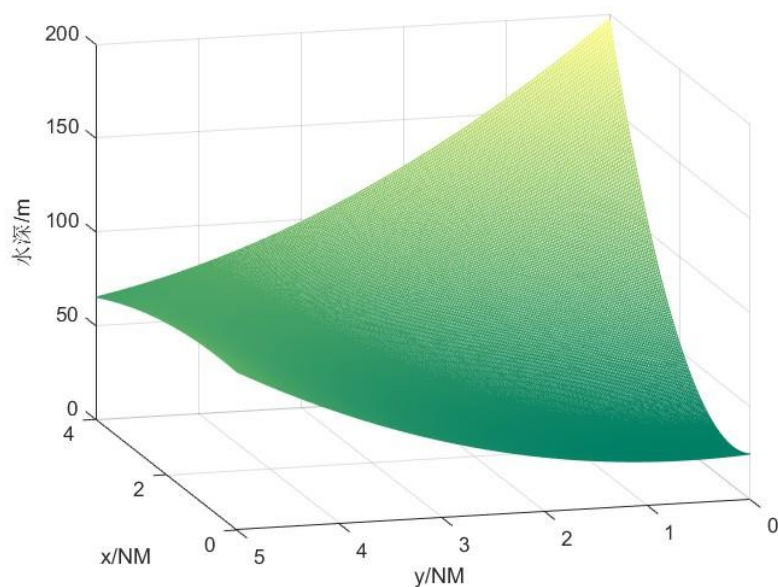


图 6-14: 多年前单波束测线水深数据 3D 热力图

2. 地形观察与假设：观察海底地形，可以发现海底地形以对角线方向为最低处，向两侧逐渐上升，其中(0,0)点处半侧坡脚更大，(0,0)点也是海底地形最高处。可以看到两侧坡度相对均匀，且多年前的单波束测线数据可能会有一定偏差，因此假设两侧斜坡坡度大致恒定。
3. 多目标优化算法：问题的主要考量三个指标：① 测线的总长度；② 漏测海区占总待测海域面积的百分比；③ 在重叠区域中，重叠率超过 20% 部分的总长度。分析各指标间的关系，易知过大的重叠说明我们设计的测线可能有较多的冗余，因而使总测线长度变长。根据多目标优化算法的原理，决定尽最大可能先满足重叠率尽可能较小（但不要低于10%）的要求，具体的：以10% ~ 20%的重复率为必须参数代入程序，同时在保证极大的覆盖面积占比的前提下尽可能的减少测线总长度。
4. 设计测线布局：考虑到(0,0)点除极大的地形高度，决定以(0,0)点往对角方向作为初始测线，并使用它结合之前的模型来计算一个初步的覆盖区域。随后设计算法，进入一个迭代过程，在这个过程中，算法会尝试调整测线的位置和角度来增加覆盖区域，同时确保测线之间的重叠率在可接受的范围内（10% ~ 20%）。在每一步迭代中，都会沿当前测线位置的正方向移动一个固定的距离（例如25m），并在当前测线角度的正方向旋转一个小的角度（例如0.5°）。对于每个新的测线位置和角度，都会计算出一个新的覆盖区域，然后与已有的覆盖区域合并。而这里重叠率是通过计算重叠区域面积与新的覆盖区域面积之比来得到的。当覆盖区域覆盖了整个海域，或当测线的角度或位置达到了一定的上限时，迭代过程会结束。

5.5 模型求解

1. 计算测线的总长度：通过 Matlab 代码多次迭代来确定测线的方向和位置，以便能够最佳地覆盖给定的海域。对于每一次迭代，都计算了当前测线的长度，并将这个长度累加到总的测线长度上。
2. 计算漏测海区占总待测海域面积的百分比：通过对覆盖矩阵 `cover` 进行分析得到的。每次迭代后，都使用当前的测线来更新覆盖矩阵。如果某个区域被覆盖了，那么该区域在矩阵中的值为1；否则，其值为0。最后算出矩阵中的 0 值占比。
3. 计算重区域中超过20%重叠率部分的总长度：由于算法的最初设定，以10% ~ 20%的重复率为必须参数代入程序，使用该算法的重区域中超过20%重叠率部分的总长度为0。

下面的几张图为 Matlab 程序输出的部分视觉展示图。下图 6-15 的两张图展示了待测区域的覆盖情况：

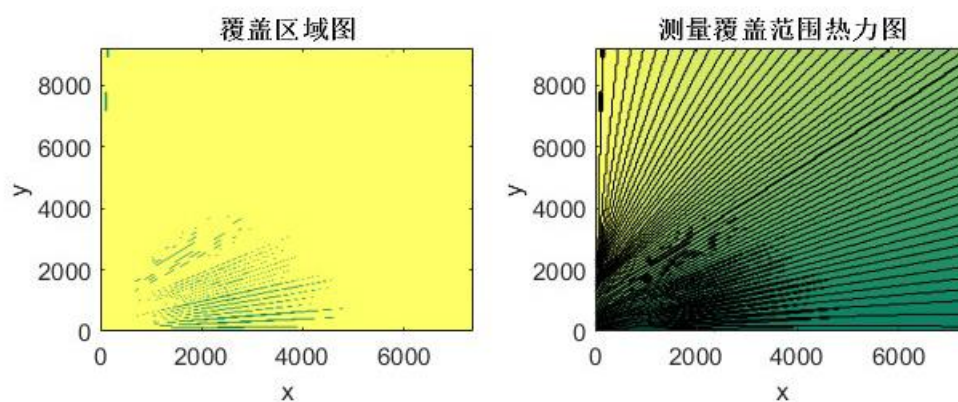


图 6-15: 测区域的覆盖情况示意图

下图 6-16 则为某一点在测量过程中被覆盖的次数:

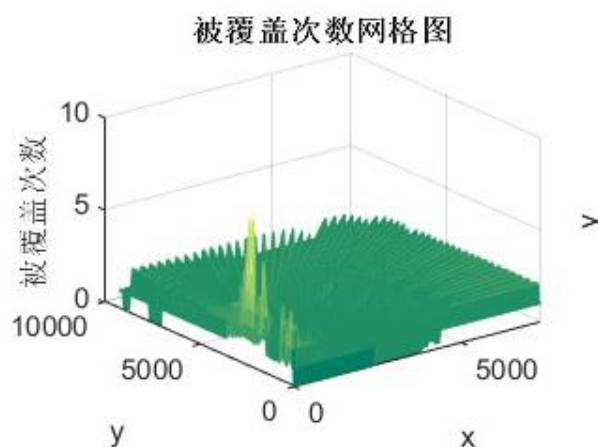


图 6-16: 某一点在测量过程中被覆盖的次数示意图

最终的计算结果展示:

- 对应优化方案下, 测线数量为: 54条
- 测线总长度为: 442,650m
- 重面积之和占总面积比例为: 17.72%
- 漏测面积占比为: 3.71%
- 重叠率超过20%的部分长度为: 0

六、 模型评价与改进

6.1 模型的评价

6.1.1 优点

1. 问题一模型在处理斜坡情景时,采用更合理化的相邻两测线的交互型覆盖宽度 W' 代替测线某一个简单的覆盖投影 W ,来进行重叠率的计算。 W' 使重叠率的描述更加合理化,与结果表格中“与前一条测线的重叠率”的说法也更为贴近。
2. 问题二模型采用空间向量运算的方式得出不同测线夹角下的 α 值,以此来调用模型一来解决问题,算法巧妙。
3. 问题三模型的构建采用了已有的专业研究论文的结论:走向要符合航道底部等深线走向,直接确定了布线方向,节省了大量测试成本。同时,结果不仅给出了精确的布线数据,还给出了三维布线图。

6.1.2 缺点

1. 问题三模型对其他可能较优情况的测试较少,可能忽略某种更优的布线方式。
2. 问题四模型对数据的处理太过理想化,可能会带来较大的误差。

6.2 模型改进

1. 对于模型三的改进,可以借助 Python 程序测试更多的可能更优布线方式,与之前的最优路线比较,查看是否有更优的路线。
2. 对于模型四的改进,采用微分的思想来计算每一个微小的坡度可以使模型的精确度得到极大的提高。

七、 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(09):46-47.DOI:10.13646/j.cnki.42-1395/u.2017.09.018.

[2] 付勇先.海底地形的多波束测深数据预处理及三维建模方法研究[D].中国地质大学(北京),2017.

[3] (美)CLEVE B. MOLER 著;张志涌等编译.,莫勒.,张志涌.,2015. MATLAB 数值计算: 2013 修订版. 中译本: Revised in 2013[M/OL]. Di 1 ban. Beijing: 北京航空航天大学出版社.

[4] 成芳,胡迺成.多波束测量测线布设优化方法研究[J].海洋技术学报,2016,35(02):87-91.

八、 附录

8.1 目录

- 附录 1: 支撑材料文件列表
- 附录 2: 相关代码展示

8.2 附录 1: 支撑材料文件列表

支撑材料编号	文件位置	说明、使用
1	question1.py	问题一计算 Python 代码
2	output/result1.xlsx	问题一结果表格
3	question2.py	问题二计算 Python 代码
4	plot/question2_plot.py	问题二绘图 Python 代码
5	output/result2.xlsx	问题二结果表格
6	question3.py	问题三计算 Python 代码
7	output/result3.xlsx	问题三结果表格
8	question3_1.py	问题三另一假设的 Python 代码（仅用于验证）
9	output/result3_1.xlsx	问题三另一假设结果表格
10	question4.m	问题四计算 Matlab 代码
11	calcdis.m	问题四相关 Matlab 封装函数 1
12	calcdis2.m	问题四相关 Matlab 封装函数 2
13	plot/question4_plot.py	问题四绘图 Python 代码

8.3 附录 2: 相关代码展示

question1.py:

```
import numpy as np
import pandas as pd

# 初始化参数
origin_D = 70 # 0 时海底深度（单位：m）
d = 200 # 测线间隔（单位：m）
theta = 120 # 换能器的开角（单位：度）
alpha = 1.5 # 坡度（单位：度）

# 恒定数
A = (
    np.sin(np.radians(theta / 2))
    / np.sin(np.radians(90 - theta / 2 - alpha))
    * np.cos(np.radians(alpha))
)
```

```

)
B = (
    np.sin(np.radians(theta / 2))
    / np.sin(np.radians(90 - theta / 2 + alpha))
    * np.cos(np.radians(alpha))
)

# 获取数据
df = pd.read_excel("question/result1.xlsx", sheet_name="Sheet1")
# print(df)

# 计算海水深度
def calc_depth(origin_D, distance, alpha) -> float:
    return origin_D - distance * np.tan(np.radians(alpha))

# 计算覆盖宽度
def calc_coverage_width(depth_D) -> (float, float):
    # 计算 W1
    W_1 = A * depth_D
    # 计算 W2
    W_2 = B * depth_D

    return W_1, W_2

# 计算重叠率
def calc_overlap(d, W_1, W_2_before) -> float:
    return 1 - d / (W_2_before + W_1)

W_before = 0 # 初始化前一条测线的右覆盖宽度
for column in df.columns[1:]:
    # 获取列中的值
    # print(df[column])

    # 计算测线距中心点处的距离
    distance = column
    # print("Distance: ", distance)

    # 计算海水深度
    depth_D = calc_depth(origin_D, int(distance), alpha)
    # print("Depth: ", depth_D)
    df.at[0, column] = depth_D

```

```

# 计算覆盖宽度
W_1, W_2 = calc_coverage_width(depth_D)
W = W_1 + W_2
# print("Coverage width: ", W)
df.at[1, column] = W

# 计算重叠率
if W_before != 0:
    overlap = calc_overlap(d, W_1, W_before)
    # print("Overlap: ", overlap * 100, "%")
    df.at[2, column] = overlap * 100
W_before = W_2

print(df)

# 将 DataFrame 保存为 Excel 文件
df.to_excel("output/result1.xlsx", index=False)

question2.py:

import numpy as np
import pandas as pd

# 初始化参数
convert = 1852 # 海里换算为米: 1 海里 = 1852 米
origin_D = 120 # 0 时海底深度 (单位: m)
d = 0.3 * convert # 测线间隔 (单位: m)
theta = 120 # 换能器的开角 (单位: 度)
alpha = 1.5 # 坡度 (单位: 度)

# 获取数据
df = pd.read_excel("question/result2.xlsx", sheet_name="Sheet1")
# print(df)

# 计算海水深度
def calc_depth(origin_D, distance, alpha, beta) -> float:
    return origin_D - distance * np.tan(np.radians(alpha)) * np.sin(
        np.radians(beta - 90)
    )

# 计算覆盖角度
def calc_coverage_angle(alpha, beta) -> float:
    inner = (

```



```

1
+ ((1 + np.tan(np.radians(alpha)) ** 2) * np.tan(np.radians(beta - 90))) ** 2
+ np.tan(np.radians(alpha)) ** 2
)
sin_value = np.tan(np.radians(alpha)) / np.sqrt(inner)
return np.arcsin(sin_value)

```

计算覆盖宽度

```
def calc_coverage_width(depth_D, alpha, theta) -> (float, float, float):
```

```
    # 计算 W1
```

```
    W_1 = (
        depth_D
        * np.sin(np.radians(theta / 2))
        / np.sin(np.radians(90 - theta / 2 - alpha))
        * np.cos(np.radians(alpha))
    )
```

```
    # 计算 W2
```

```
    W_2 = (
        depth_D
        * np.sin(np.radians(theta / 2))
        / np.sin(np.radians(90 - theta / 2 + alpha))
        * np.cos(np.radians(alpha))
    )
```

```
    # 计算 W
```

```
    W = W_1 + W_2
```

```
    return W, W_1, W_2
```

```
beta_list = df.iloc[1:, 1].tolist()
```

```
# print("Beta list: ", beta_list)
```

```
for column in df.columns[2:]:
```

```
    print("=====")
```

```
    # 获取列中的值
```

```
    # print(df[column])
```

```
    # 计算测线距中心点处的距离
```

```
    distance = df[column][0] * convert
```

```
    print("Distance: ", distance)
```

```
for index, beta in enumerate(beta_list):
```

```
    print(f"Index: {index}, Beta: {beta}")
```

```
    # 计算海水深度
```

```

depth_D = calc_depth(origin_D, distance, alpha, beta)
print("Depth: ", depth_D)

# 计算覆盖角度
alpha_new = calc_coverage_angle(alpha, beta)

# 计算覆盖宽度
W, W_1, W_2 = calc_coverage_width(depth_D, alpha_new, theta)
print("Coverage width: ", W)
df.at[index + 1, column] = W

```

```
# print(df)
```

```

# 将 DataFrame 保存为 Excel 文件
df.to_excel("output/result2.xlsx", index=False)

```

question3.py:

```

import numpy as np
import pandas as pd

# 定义基本参数
convert = 1852 # 海里换算为米: 1 海里 = 1852 米
origin_D = 110 # 海域中心点处的海底深度 (单位: m)
theta = 120 # 换能器的开角 (单位: 度)
alpha = 1.5 # 坡度 (单位: 度)
a = 4 * convert # 东西宽 (单位: m)
b = 2 * convert # 南北长 (单位: m)
min_D = origin_D - a / 2 * np.tan(np.radians(alpha)) # 海底最小深度 (单位: m)
# print("min_D: ", min_D)

# 恒定数
A = (
    np.sin(np.radians(theta / 2))
    / np.sin(np.radians(90 - theta / 2 - alpha))
    * np.cos(np.radians(alpha))
)
B = (
    np.sin(np.radians(theta / 2))
    / np.sin(np.radians(90 - theta / 2 + alpha))
    * np.cos(np.radians(alpha))
)

# 计算海水深度

```

```

def calc_depth(origin_D, distance, alpha) -> float:
    return origin_D + distance * np.tan(np.radians(alpha))

# 计算覆盖宽度
def calc_coverage_width(depth_D) -> (float, float):
    # 计算 W1
    W_1 = A * depth_D
    # 计算 W2
    W_2 = B * depth_D

    return W_1, W_2

# 计算距离
def calc_distance(depth_D_before, W_before, alpha, theta) -> float:
    return (0.9 * W_before + 0.9 * depth_D_before * B) / (
        1 - np.tan(np.radians(alpha)) * B
    )

data = []

i = 1
total = 0
W_before = 0
depth_D_before = min_D
while total + W_before < a:
    if W_before != 0:
        # 计算距离
        distance = calc_distance(depth_D_before, W_before, alpha, theta)
    else:
        distance = min_D * np.tan(np.radians(theta / 2))
    print(total)

    # 计算海水深度
    depth_D = calc_depth(depth_D_before, distance, alpha)

    # 计算 W1
    W_1, W_2 = calc_coverage_width(depth_D)

    total += distance
    print(total + W_1)

    data.append([i, a / 2 - total, depth_D, W_1 + W_2, i * b])
    depth_D_before = depth_D

```

```
W_before = W_1  
i += 1
```

```
df = pd.DataFrame(data, columns=["轮次", "测线距中心点处的距离/m", "海水深度/m", "覆盖宽度/m",  
"总测量长度/m"])  
df.to_excel("output/result3.xlsx", index=False)
```

question3_1.py:

```
import numpy as np  
import pandas as pd
```

定义基本参数

```
convert = 1852 # 海里换算为米: 1 海里 = 1852 米  
origin_D = 110 # 海域中心点处的海底深度 (单位: m)  
theta = 120 # 换能器的开角 (单位: 度)  
alpha = -1.5 # 坡度 (单位: 度)  
a = 4 * convert # 东西宽 (单位: m)  
b = 2 * convert # 南北长 (单位: m)  
min_D = origin_D - a / 2 * np.tan(np.radians(alpha)) # 海底最小深度 (单位: m)  
# print("min_D: ", min_D)
```

恒定数

```
A = (  
    np.sin(np.radians(theta / 2))  
    / np.sin(np.radians(90 - theta / 2 - alpha))  
    * np.cos(np.radians(alpha))  
)  
B = (  
    np.sin(np.radians(theta / 2))  
    / np.sin(np.radians(90 - theta / 2 + alpha))  
    * np.cos(np.radians(alpha))  
)
```

计算海水深度

```
def calc_depth(origin_D, distance, alpha) -> float:  
    return origin_D + distance * np.tan(np.radians(alpha))
```

计算覆盖宽度

```
def calc_coverage_width(depth_D) -> (float, float):  
    # 计算 W1  
    W_1 = A * depth_D  
    # 计算 W2
```

```

W_2 = B * depth_D

return W_1, W_2

# 计算距离
def calc_distance(depth_D_before, W_befeore, alpha, theta) -> float:
    return (0.9 * W_befeore + 0.9 * depth_D_before * B) / (
        1 - np.tan(np.radians(alpha)) * B
    )

data = []

i = 1
total = 0
W_before = 0
depth_D_before = min_D
while total + W_before < a:
    if W_before != 0:
        # 计算距离
        distance = calc_distance(depth_D_before, W_before, alpha, theta)
    else:
        distance = min_D * np.tan(np.radians(theta / 2))
    print(total)

    # 计算海水深度
    depth_D = calc_depth(depth_D_before, distance, alpha)

    # 计算 W1
    W_1, W_2 = calc_coverage_width(depth_D)

    total += distance
    print(total + W_1)

    data.append([i, a / 2 - total, depth_D, W_1 + W_2, i * b])
    depth_D_before = depth_D
    W_before = W_1
    i += 1

df = pd.DataFrame(data, columns=["轮次", "测线距中心点处的距离/m", "海水深度/m", "覆盖宽度/m",
"总测量长度/m"])
df.to_excel("output/result3_1.xlsx", index=False)

```

question4.m:

```

clear all; % 清空工作空间中的变量
clc; % 清除命令行窗口中的内容
close all; % 关闭所有图形窗口

% 初始化参数
convert = 1852; % 海里换算为米: 1 海里 = 1852 米
lines = [0, 50, 0]; % 设置初始测线序列
rad = pi / 180; % 角度转换为弧度
osf = 1; % 偏移量字段
angleStepMax = 5; % 最大步长角度

% 读取表格数据
[Data] = xlsread('./question/附件.xlsx');
x = Data(1, 2:end); % x 坐标
y = Data(2:end, 1); % y 坐标
z = Data(2:end, 2:end); % 水深数据

% 绘制三维网格图
figure(1);
colormap('summer');
mesh(x, y, z);
xlabel('x/NM');
ylabel('y/NM');
zlabel('水深/m');

xi = x * convert; % 转换为米
yi = y * convert; % 转换为米
[xi2, yi2] = meshgrid(0:xi(2)/osf:xi(end), 0:yi(2)/osf:yi(end));
z2 = interp2(xi, yi, z, xi2, yi2, 'linear');
meshXY = [xi2(:), yi2(:), -z2(:)]; % 网格坐标
% 计算初始测线覆盖情况:
% - coverOrigin 为初始测线覆盖情况数据
% - distanceOrigin 为初始测线覆盖的总距离
[coverOrigin, distanceOrigin] = calcdis(...
    lines(1, 1:2), lines(1, 3) * rad, (-15e3:25:15e3), ...
    meshXY, size(z2), xi(1), xi(end), yi(1), yi(end) ...
);
cover = coverOrigin;
linesOrigin = lines(end, 1:3); % 设置初始测线序列

% 测线优化循环
while (linesOrigin(2) < yi(end) && linesOrigin(3) < 89.5 && mean(cover(:)) < 0.99)
    linesOrigin = linesOrigin + [0, 25, 0];
    flag = 1;
    fprintf('Overlap: ');

```

```

% 从 linesOrigin(3) 到 linesOrigin(3) + angleStepMax 以 0.5 为步长
for phi = linesOrigin(3):0.5:linesOrigin(3) + angleStepMax
    [coverOrigin] = calcdis(...
        linesOrigin(1:2), phi * rad, (-15e3:25:15e3), ...
        meshXY, size(z2), xi(1), xi(end), yi(1), yi(end) ...
    );
    coverArea = sum(coverOrigin(:));
    cover2Area = sum(cover(:) + coverOrigin(:) == 2);

    % 计算重叠率
    overlap = cover2Area / coverArea;
    fprintf('%1.2f%%\t', overlap * 100)
    % 0.1 < overlap < 0.2 时，认为是重叠率刚好
    if (overlap > 0.1 && overlap < 0.2)
        flag = 0;
        break;
    elseif (overlap < 0.05) % ! ! ! !
        break;
    end
end

if (flag == 0)
    % 更新覆盖情况，并将大于等于 1 的部分设置为 1
    cover = (cover + coverOrigin) >= 1;
    % 更新测线序列
    lines = [lines; linesOrigin(1:2), phi];
    [linesOrigin(1:2), phi]
    linesOrigin = lines(end, 1:3);
end
fprintf('\n');
end

% 计算最终测线覆盖情况和总距离
[cover, coverTotal, distance] = calcdis2(...
    lines(:, 1:2), lines(:, 3) * rad, (-15e3:25:15e3), ...
    meshXY, size(z2), xi(1), xi(end), yi(1), yi(end) ...
);

% 绘制四个子图
clf;
figure(2);
colormap('summer');
% 覆盖区域图
subplot(221);

```

```

contourf(xi2, yi2, cover >= 1, 'linestyle', 'none');
xlabel('x');
ylabel('y');
title('覆盖区域图');
% 测量覆盖范围热力图
subplot(222);
contourf(xi2, yi2, coverTotal, 55);
xlabel('x');
ylabel('y');
zlabel('测线编号');
title('测量覆盖范围热力图');
% 被覆盖次数网格图
subplot(223);
mesh(xi2, yi2, cover);
xlabel('x');
ylabel('y');
zlabel('被覆盖次数');
title('被覆盖次数网格图');
% 重叠区域热力图
subplot(224);
contourf(xi2, yi2, cover > 1, 'linestyle', 'none');
xlabel('x');
ylabel('y');
title('重叠区域热力图');

fprintf('该优化方案下，测线条数%d\n 侧线总长度为%1.2fm\n', size(lines, 1), distance);
fprintf('重叠面积之和占总面积比例%1.2f%%\n', (sum(cover(:)) - sum(cover(:) >= 1)) / (sum(cover(:) > 0)) * 100);
fprintf('漏测面积占比%1.2f%%\n 重叠率超过 20%%的部分长度%d\n', 100 * mean(cover(:) == 0), 0);

```

calcdis.m:

```

function [cover,distance]=calcdis(lines,beta,dirList,meshXYZ,outersize,xMin,xMax,yMin,yMax)

cover = zeros(outersize); % 初始化数据矩阵 cover
ceDir = [cos(beta) sin(beta) 0]; % 计算 ceDir 向量
lineZ = ones(length(dirList), 1) * lines + dirList * ceDir(1:2); % 计算 lineZ 矩阵
Ind = lineZ(:, 1) >= xMin & lineZ(:, 1) <= xMax & lineZ(:, 2) >= yMin & lineZ(:, 2) <= yMax; % 按照条件筛选 lineZ 中的数据
lineZ = lineZ(Ind, :); % 更新 lineZ 矩阵，只保留满足条件的数据
distance = norm(lineZ(1, :) - lineZ(end, :)); % 计算 lineZ 的起点和终点之间的距离

for jj = 1:size(lineZ, 1)
    dir11 = meshXYZ - ones(size(meshXYZ, 1), 1) * [lineZ(jj, :) 0]; % 计算 dir11 矩阵

```



```

distanceAng = (-dir11(:, 3) ./ (sum(dir11.^2, 2).^0.5)); % 计算 distanceAng 向量
cover(distanceAng > 0.5) = 1; % 根据条件更新 cover 矩阵
end

calcdis2.m:

function [cover, cover2, distance] = calcdis2(LineLocSeq, betaSeq, dirLine, meshXYZ, OutSize, xMin,
xMax, Ymin, Ymax)
    cover = zeros(OutSize); % 初始化 cover 矩阵
    cover2 = zeros(OutSize); % 初始化 cover2 矩阵
    distance = 0; % 初始化 distance 变量
    linenum = size(LineLocSeq, 1); % 获取 LineLocSeq 的行数

    for ii = 1:linenum
        cover0 = zeros(OutSize); % 初始化 cover0 矩阵
        LineLoc = LineLocSeq(ii, :); % 获取 LineLocSeq 中的一行
        beta = betaSeq(ii); % 获取 betaSeq 中的一个元素
        CEDir = [cos(beta) sin(beta) 0]; % 计算 CEDir 向量
        LineZ = ones(length(dirLine), 1) * LineLoc + dirLine' * CEDir(1:2); % 计算 LineZ 矩阵
        ind = LineZ(:, 1) >= xMin & LineZ(:, 1) <= xMax & LineZ(:, 2) >= Ymin & LineZ(:, 2) <= Ymax; %
按照条件筛选 LineZ 中的数据
        LineZ = LineZ(ind, :); % 更新 LineZ 矩阵，只保留满足条件的数据
        distanceOrigin = norm(LineZ(1, :) - LineZ(end, :)); % 计算 LineZ 的起点和终点之间的距离

        for jj = 1:size(LineZ, 1)
            dir11 = meshXYZ - ones(size(meshXYZ, 1), 1) * [LineZ(jj, :) 0]; % 计算 dir11 矩阵
            distanceAngle = (-dir11(:, 3) ./ (sum(dir11.^2, 2).^0.5)); % 计算 distanceAngle 向量
            cover0(distanceAngle > 0.5) = 1; % 根据条件更新 cover0 矩阵
        end

        cover = cover + cover0; % 累加 cover0 到 cover
        cover2(cover0 == 1) = ii; % 将 cover0 中为 1 的位置标记为当前的线路编号 ii
        distance = distance + distanceOrigin; % 累加 distanceOrigin 到 distance
    end
end

```

question2_plot.py:

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# 获取数据
df = pd.read_excel("output/result2.xlsx", sheet_name="Sheet1")

```

```

print(df)

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

# Create the mesh in polar coordinates and compute corresponding Z.
r = df.iloc[1:, 1].tolist()
p = df.iloc[0, 2:].tolist()
R, P = np.meshgrid(r, p)
Z = np.array(df.iloc[1:, 2:].values.tolist())

# Express the mesh in the cartesian system.
X, Y = R*np.cos(P), R*np.sin(P)

# Plot the surface.
ax.plot_surface(X, Y, Z, cmap=plt.cm.YlGnBu_r)

# Tweak the limits and add latex math labels.
ax.set_xlabel(r'$\phi_{\mathrm{real}}$')
ax.set_ylabel(r'$\phi_{\mathrm{im}}$')
ax.set_zlabel(r'$V(\phi)$')

plt.show()

```

question4_plot.py:

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# 获取数据
df = pd.read_excel("../question/附件.xlsx", sheet_name="Sheet1")

# 获取 x 轴数据
x = df.iloc[0, 2:]
# print(x)

# 获取 y 轴数据
y = df.iloc[1:, 1]
# print(y)

# 获取每个 (xi,yi) 下的 z 轴数据
z = df.iloc[1:, 2:]
# print(z)

```

```
xi, yi = np.meshgrid(x.tolist(), y.tolist())
zi = np.array(z.values.tolist())

# 设置中文字体
plt.rcParams["font.sans-serif"] = ["SimHei"]
plt.rcParams["axes.unicode_minus"] = False
# 绘制
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
surf = ax.plot_surface(xi, yi, zi, cmap="BuPu", linewidth=0, antialiased=False)
fig.colorbar(surf)
ax.set_title("单波束测量海水深度数据三维图")
plt.show()
```