

为了感谢惠俊英老师的谆谆教诲，特为好奇诸君奉上水声物理--海洋传播声场模拟的一份武林秘籍！

当好奇诸君阅毕此卷，再对照轻松跑完六份作业（六路招式），即入“水声物理”之江湖！

BELLHOP 与 AcousticsToolbox 遵照 GNU 公共许可证进行分发。感谢 Michael B. Porter 对 BELLHOP 的原创，也感谢 Orlando Camargo Rodríguez 此份大道至简的入门秘籍。更感谢由 Free Software 凝聚起来的自由科学家社区。

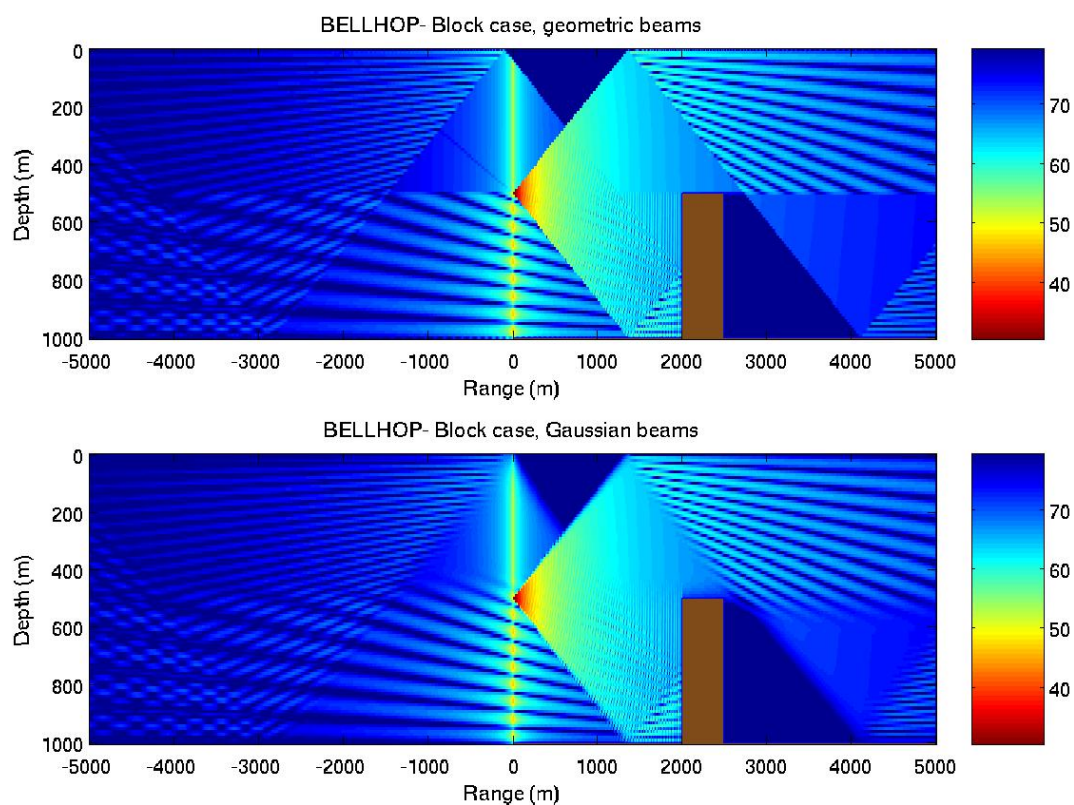
吴开明 2018.12

招式说明：

- 1、在 64 位 windows 平台上，将本程序包的文件夹添加在 matlab “设置路径”的“添加并包含子文件夹”。
- 2、读完文档，跑完本程序包所有招式（作业），射线追踪算法的运行就算初步通关，对于海洋传播声场的二维射线表达方法基本上建立感性认识。单单溜达作业（六路招式）耗时半天。
- 3、对于每个作业的主程序文件一句一句地消化，深化理解每份作业的内涵，算是提高。耗时估计 3-7 天。
- 4、本文档适合大三以上和非水声物理专业同志们的快速参学。

# BELLHOP 射线追踪程序概述

(2008 年 6 月发布)



Orlando Camargo Rodríguez

物理系信号处理实验室

葡萄牙 阿尔加夫大学

*Universidade do Algarve*

版本 1.0 (13/06/2008)

吴开明 编译 (2018.08-10)

## 1 引言

Bellhop 是一款高效的射线追踪程序，由 Michael Porter (<http://hlsresearch.com/personnel/porter/index.php>) 用 Fortran 语言编写，并纳为声学工具箱的一部分（可在海洋声学库网站 (<http://oalib.hlsresearch.com/>) 获取）。

Bellhop 设计用于执行二维声线追踪。适合给定声速剖面  $c(z)$  或声速剖面场  $c(r,z)$  的海洋波导，其边界可以是水平的，也可以是变化的吸收边界。

输出选项包括射线坐标、传播时间、振幅、本征射线、声压或传播损失(包括相干的、非相干的或半相干的)。声压计算基于高斯波束理论[1, 2]，可以采用不同方法近似，包括：

- 几何波束(默认选项)[3]；
- 具有中心射线坐标的波束；
- 具有笛卡尔坐标的波束；
- 高斯射线近似[4]。

本说明书针对在海洋声学领域工作，并对在特定应用中使用 Bellhop 感兴趣的人。在各章节中，我们将简要讨论如何获得和安装模型，它求解的方程，并通过一组范例来展现 Bellhop 的各种能力。

## 2 理论背景

射线追踪需要求解射线方程来确定射线坐标。要得到振幅和声压，需要求解动态射线方程，这在文献[1]中有详细描述。

对于圆柱对称系统，射线方程可以写成[2]

$$\begin{aligned}\frac{dr}{ds} &= c\xi(s), & \frac{d\xi}{ds} &= -\frac{1}{c^2} \frac{\partial c}{\partial r}, \\ \frac{dz}{ds} &= c\zeta(s), & \frac{d\zeta}{ds} &= -\frac{1}{c^2} \frac{\partial c}{\partial z},\end{aligned}\quad (1)$$

其中  $r(s)$  和  $z(s)$  代表圆柱坐标系中的射线坐标， $s$  是沿射线的弧长；沿射线的切线矢量表示为  $c(s)[\xi(s), \zeta(s)]$  对。  $r(s)$ 、 $z(s)$ 、 $\xi(s)$  和  $\zeta(s)$  的初始条件是

$$r(0) = r_s, \quad z(0) = z_s, \quad \xi(0) = \frac{\cos \theta_s}{c_s}, \quad \zeta(0) = \frac{\sin \theta_s}{c_s},$$

其中  $\theta_s$  代表出射角， $(r_s, z_s)$  是声源位置， $c_s$  是声源位置处的声速。有了这些坐标，就足以确定射线的传播时间：

$$\tau = \int_{\Gamma} \frac{ds}{c(s)} \quad (2)$$

该式沿曲线  $[r(s), z(s)]$  计算。

### 3 安装

所有包含在声学工具箱中的模型的 Fortran 源程序都提供了一个 makefile，以及用于展示射线坐标、声压或传播损失的一套 Matlab 应用程序。该工具箱已通过以下编译器的成功测试：

- Linux 平台上的 Intel fortran 编译器；
- gfortran 编译器；
- g95 编译器。

对各 makefile 稍作修改（这取决于所用的特定 Fortran 编译器），在命令行键入 make clean 和 make install，就能编译 Bellhop 在内的所有模型。一旦生成二进制（执行）文件，就可以使用工具箱中的（Matlab）脚本调用模型（进行计算）。

### 4 数值问题

在 Bellhop 中，采用两步多边形方法对射线和动态方程进行积分。对于给定的输出选项，模型针对用户指定的（位置）网格点，（在输出文件中）写入从每个出射角发出的声线的坐标、或逐步计算出的传播时间数据、声压或传播损失。

## 5 输入输出文件

要求 Bellhop 的输入文件（扩展名\*.env）必须与 Kraken（一套简正波模型）的输入文件相兼容。因此，在射线追踪过程中，一些参量并不使用。Bellhop 使用的\*.env 输入文件的总体结构如下：

表 1 Bellhop 使用的\*.env 输入文件的总体结构表（英文原版）

TITLE	
Frequency (in Hz)	
nmedia (dummy integer < 20)	
OPTIONS1	
SURFACE-LINE	
nmesh sigmas z(nssp) z(1) cp(1) / z(2) cp(2) / . . . . . . z(nssp) cp(nssp) /	Sound Speed Block
OPTIONS2 sigmab BOTTOM-LINE	Bottom Block
nsources (number of sources) source-depth(1) source-depth(nsources) / (in m) nrd (number of receivers x depth) receiver-depth(1) receiver-depth(nrd) / (in m) nrr (number of receivers x range) receiver-range(1) receiver-range(nrr) / (in km)	Array Block
OPTIONS3 nbeams (number of launching angles) theta(1) theta(nbeams) (launching angles in degrees) ray-step zmax rmax	Output Block
OPTIONS4 epmult rloop isingl nimage ibwin component	Beam Block

表 2 Bellhop 使用的\*.env 输入文件的总体结构表（中文说明）

条目	内 容		说 明		
1	TITLE		绘图的标题，简述模拟内容。		
2	Frequency(Hz)	频率（Hz）	声线与频率无关，频率影响声线步长大小；更高频率可画出更精确的声线轨迹。		
3	nmedia	整数<20	介质分层，Bellhop 总为 1。		
4	OPTIONS1	顶部选项	海面版块		
5	SURFACE-LINE	海表面参数行			
6	nmesh sigmas z(nssp)		Bellhop 只使用第三个参数：海深--最后一行声速剖面的深度。		
7	z(1) cp(1) / z(2) cp(2) / . . . . . . z(nssp) cp(nssp) /		声速剖面版块		
			用“/”结束每一行，表示停止读取该行，本行其他参数（cs、ρ、α）使用默认值。		
8	OPTIONS2 sigmab	底端选项	海底版块		
9	BOTTOM-LINE	海底参数行			
10	nsources		声源数目	阵列版块	
11	source-depth(1) source-depth(nsources) /		声源深度（m）；只简单输入第一个与最后一个深度，用“/”结束该行，表示其他深度均匀内插。		
12	nrd (number of receivers x depth)		接收器数目		
13	receiver-depth(1) receiver-depth(nrd) /		接收器深度（m）		
14	nrr (number of receivers x range)		x 轴向接收器数目		
15	receiver-range(1) receiver-range(nrr) /		x 轴向接收器距离（km）		
16	OPTIONS3		运行类型(RunType)		输出版块
17	nbeams		出射角数目；超过 50 根，绘图就凌乱。设为 0 时，代码会自动优选数目。		
18	theta(1) theta(nbeams)		出射角扇面，单位度(°)；朝向底部出射的射线为正角度。		
19	ray-step zmax rmax		ray-step: 声线步长(m)，Bellhop会动态调整步长，确保每根声线精确着陆在声速给出深度。 zmax, rmax: 追踪声线的深度-距离范围（box）。		
20	OPTIONS4 epmult rloop isingl		当OPTIONS3为单个字符时，无此两行。		波束版块
21	nimage ibwin component				

OPTIONS1（选项1）是用单引号括起来的5个字符的字符串。下面是对每个字符的说明：

- OPTIONS1(1): 设定 Bellhop 用来计算声速以及声速沿声线的导数的插值方法。对应如下：

----'S': 三次样条插值；

----'C': C-线性插值；

----'N': N2 线性插值；

----'A': 解析插值（需要调整子程序 SSP 和并重新编译模型代码）；

----'Q': 声速场二次逼近（需要创建包含声速场的\*.ssp 文件）。

- OPTIONS1(2): 设置表面类型，对应如下：

----'V': 表面以上为真空（不需要表面参数行“SURFACE-LINE”）；

----'R': 表面以上为完全刚性介质（不需要表面行）；

----'A': 声学半空间；表面行“SURFACE-LINE”应写为  
表面高度 z 表面 cp 表面 cs 表面密度  $\rho$  表面  $\alpha$  /

----'F': 从\*.irc 文件中读取反射系数表。需要先运行  
bounce（反弹）程序得到\*.irc 文件。

- OPTIONS1(3): 设定底部衰减单位（详参文献[5]），对应如下：

----'F': 衰减单位采用(dB/m)kHz；



----'L': 衰减单位采用参数损失;

----'M': 衰减单位采用 dB/m;

----'N': 衰减单位采用 Nepers/m;

----'Q': 衰减单位采用 Q 因子;

----'W': 衰减单位采用 dB/ $\lambda$ (波长)

- **OPTIONS1(4):** 描述水体 Thorpe 体积衰减系数的可选参数; 如果设置, 对应为'T'。
- **OPTIONS1(5):** 描述表面形状的可选参数; 如果不具体说明, 就认为表面是平的; 如果具体说明, 则设置为'\*.'. 在后一种情况下, 表面坐标需要用\*.ati 文件来描述, 其结构如下:

interpolation type (插值类型)	
npoints (点数)	
r(1)	z(1)
r(2)	z(2)
.	.
.	.
.	.
r(npoints)	z(npoints)

参数“插值类型”是一个字符, 等于'L' (用于对表面进行线性插值) 或'C' (用于对表面进行曲线插值); 表面的距离单位为 km, 表面的深度单位为 m。

在 Bellhop 中不使用参数 nmesh 和 sigmas ( $\sigma$ ), 参数 z(nssp)用于检测声速剖面的最后一行。z( )和 cp( )的数值分别对应于以 m 为单位的深度和以 m/s 为单位的 p 波声速。

**OPTIONS2** (选项 2) 是用单引号括起来的两个字符的字符串。下面是对每个字符的说明:

- OPTIONS2(1): 设定水体之下的介质类型，对应如下：

----'V': 水体之下为真空（不需要底端参数行“BOTTOM-LINE”）；

----'R': 水体之下为刚性介质（不需要底端参数行）；

----'A': 声学半空间；底端行“BOTTOM-LINE”应写为：

底端深度  $z$  底端  $cp$  底端  $cs$  底端密度  $\rho$  底端  $\alpha$  /

很明显，“底端深度  $z$ ”等于  $z(nssp)$ ，“底端  $cs$ ”一般被忽略，“底端密度  $\rho$ ”单位为  $g/cm^3$ ，对“底端（衰减系数） $\alpha$ ”的单位规定同 OPTIONS1(3)；

----'F': 从\*.brc 文件中读取反射系数表。需要先运行 **bounce（反弹）** 程序得到\*.brc 文件。

- OPTIONS2(2): 描述底部形状。可以空着(对应于平坦底面)，也可以设置为 '\*'。在后一种情况下，底部坐标用\*.bty 文件来描述，结构如下：

interpolation type（插值类型）	
npoints（点数）	
r(1)	z(1)
r(2)	z(2)
.	.
.	.
.	.
r(npoints)	z(npoints)

同样，“插值类型”可设置为'L'或'C'；底端的距离单位为 km，底端的深度单位为 m。

下面六行说明声源数目及对应深度(单位为 m)，以及沿距离和深度分布的接收器数目。其行意已自明。

OPTIONS3（选项 3）是 5 个字符的字符串，用于设定输出选项。

以下乃各字符之说明：

- OPTIONS3(1): 设定将写入输出文件的信息类型。分别对应：

----'A': 写入振幅和传播时间；

----'E': 写入本征射线坐标；

----'R': 写入射线坐标；

----'C': 写入相干声压；

----'I': 写入非相干声压；

----'S': 写入半相干声压。

- OPTIONS3(2): 设置声压计算采用的近似方法；它可以空着，也可以对应以下字符：

----'G': 采用几何波束（默认值）

----'C': 采用笛卡尔波束；

----'R': 采用中心射线波束；

----'B': 采用高斯波束。

- OPTIONS3(3): 设定包含波束移位效果等的选项；它可以空着，也可以选择以下字符：

----'': 不包括波束移位效果（默认值）；

----'S': 包括波束移位效果；

----'\*': 应用声源波束模式文件（需要一个\*.sbp 文件，类似于\*.ati 和\*.bty 文件，以角度和振幅代替距离和深度）。

- **OPTIONS3(4):** 设置声源类型；可以空着，也可以选择以下字符：

----'R': 圆柱坐标系中的点源(默认值)；

----'X': 笛卡尔坐标系中的线源。

- **OPTIONS3(5):** 设置阵列类型；它可以空着，也可以选择以下字符：

----'R': 线列接收阵网格，接收器置于  $rr(:) \times rd(:)$  网格点（默认值）；

----'I': 不规则网格，接收器在  $rr(:)$ ,  $rd(:)$ 。

整数  $nbeams$  表示出射角的数目， $\theta(1)$ 和  $\theta(nbeams)$ 是以度 ( $^{\circ}$ ) 为单位设定的第一个和最后一个出射角，以指向底部的出射角为正值，指向表面的出射角为负值。参数  $ray-step(m)$ 、 $zmax(m)$ 和  $rmax(km)$ 定义射线和动态方程积分中的射线步距  $ds$  和 “声线追踪区域 (box)” 的边界范围，“追踪区域” 外就停止射线的追踪。

当**OPTIONS3**的参数由单个字符组成时，就不需要在\*.env输入文件中包含更多的行。否则，模型就需要额外的两行，它们是关于波束特征的附加信息。在这两行中，**OPTIONS4**（选项4）的参数是由两个字符组成的字符串。下面说明其中各个字符。

- **OPTIONS4(1):** 设置波束类型，对应如下：

----'C': Cerveny 类型；

----'F': 空间填充；

----'M': 最小宽度;

----'W': WKB 波束。

● OPTIONS4(2): 设定波束曲率类型, 对应如下:

----'D': 采用曲率加倍;

----'S': 采用标准曲率;

----'Z': 采用零曲率。

参数 `epmult` 和 `rloop` 应该为正实数, `isingl`、`nImage` 和 `ibwin` 应该为整数。整数 `nImage` 可以取 1、2 或 3。`Component` 是单个字符, 只有在采用中心射线坐标 (`OPTIONS3(2)='R'`) 计算声压时才使用; 它可以空着 (将声压写进输出文件)、等于 'H' (将声压的水平分量写入输出文件) 或等于 'V' (将声压的垂直分量写入输出文件)。

## 6 算例

在本节中，我们将提供输入文件，通过各种类型的算例来描述 Bellhop 的能力。在所有算例中，我们均采用 Munk 深海声速剖面，界于 0 到 5000m 深度之间，声源频率 50Hz，位于 1000m 深度，声线步距 100m，声线 70 根，出射角扇面为  $-13^\circ \sim 13^\circ$ ，海底声速 1600m/s，海底密度  $1.8 \text{ g/cm}^3$ ，海底衰减系数  $0.8 \text{ dB}/\lambda$ ，海洋环境如图 1 所示。

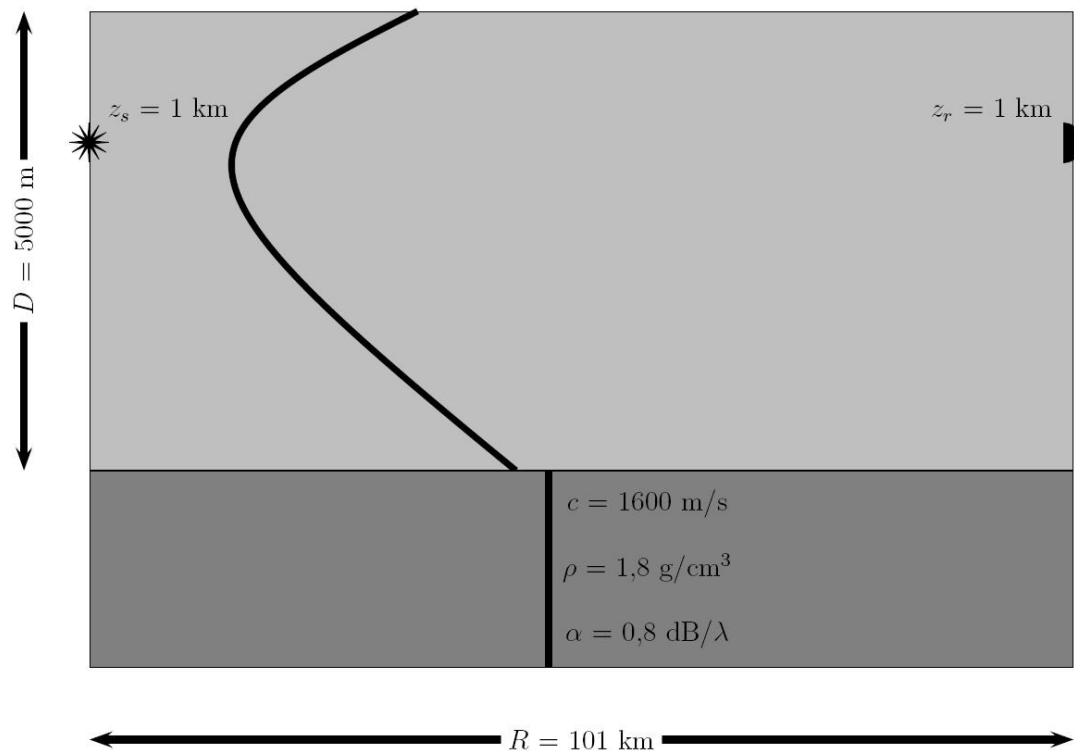


图 1：在所有算例中应用的深海场景示意图。

6.1 水平海底波导

让我们以从 0 到 101km 之间简单的声线轨迹（计算）作为开始，  
输入文件名为 flatwav\_R.env。如下所示：

flatwav_R.env / Ray trace					
行号	内 容	说 明			
1	'Munk / Flat / Ray trace'	TITLE			
2	50.0	FREQ (Hz)			
3	1	NMEDIA			
4	'SVW'	SSP-TOP-WATER-OPT			
5	51 0.0 5000.0	NMESH SIGMA Z(NSSP)			
6	0.0 1548.52 /	Z()	CP()	CS()	RHO()
7	200.0 1530.29 /	AP()	AS()		
8	250.0 1526.69 /	.	.	.	.
9	400.0 1517.78 /	.	.	.	.
10	600.0 1509.49 /	.	.	.	.
11	800.0 1504.30 /	.	.	.	.
12	1000.0 1501.38 /	.	.	.	.
13	1200.0 1500.14 /	.	.	.	.
14	1400.0 1500.12 /	.	.	.	.
15	1600.0 1501.02 /	.	.	.	.
16	1800.0 1502.57 /	.	.	.	.
17	2000.0 1504.62 /	.	.	.	.
18	2200.0 1507.02 /	.	.	.	.
19	2400.0 1509.69 /	.	.	.	.
20	2600.0 1512.55 /	.	.	.	.
21	2800.0 1515.56 /	.	.	.	.
22	3000.0 1518.67 /	.	.	.	.
23	3200.0 1521.85 /	.	.	.	.
24	3400.0 1525.10 /	.	.	.	.
25	3600.0 1528.38 /	.	.	.	.
26	3800.0 1531.70 /	.	.	.	.
27	4000.0 1535.04 /	.	.	.	.
28	4200.0 1538.39 /	.	.	.	.
29	4400.0 1541.76 /	.	.	.	.
30	4600.0 1545.14 /	.	.	.	.
31	4800.0 1548.52 /	.	.	.	.
32	5000.0 1551.91 /	.	.	.	.
33	'A' 0.0	BOTOPT SIGMA			
34	5000.0 1600.00 0.0 1.8 .8 0.0	ZB	CPB	CSB	RHOB
35	1	APB	ASB		
36	1000.0 /	NSD			
37	1	SD(1:NSD) (m)			
38	1000.0 /	NRD			
39	1	RD(1:NRD) (m)			
40	101.0 /	NRR			
41	'R'	RR(1:NRR ) (km)			
42	70	OPTION: 'R/E/C/A/I/S'			
		NBEAMS ISINGLE			

43	-13.0 13.0 /	ALPHA(1:NBEAMS) (°)
44	100.0 5500.0 102.0	STEP (m) ZBOX (m) RBOX (km)

在命令行调用 Bellhop，输入：

whatevershell\$ bellhop flatwav\_R <ENTER>

一旦 Bellhop 完成计算，可以查验是否创建了两个文件：第一个是 flatwav\_R.prt，包含波导特性、出射角数目、计算时间等相关的综合信息；第二个是 flatwav\_R.ray，它是包含射线坐标的 ASCII 码文件，并可用 M-文件 plotray.m 来绘进行图，结果如图 2 所示：

>> plotray( 'flatwav\_R' ) <ENTER>

将 OPTIONS3(1) = 'R' 改为 OPTIONS3(1) = 'E', 和前面同样地运行程序，可以得到图 3 所示的本征声线。

计算相干传播损失需要对输入文件做点小修改。首先，设置 OPTIONS3(1) = 'C'；其次，需要考虑是在 501×501 点的距离-深度矩形网格上计算传播损失。最后，我们设置 nbeams = 0，让 Bellhop 自行决定所需的射线根数。然后，输入文件变为：

flatwav_C.env / Coherent TL						
行号	内 容	说 明				
1	'Munk / Flat / Coherent TL'	TITLE				
2	50.0	FREQ (Hz)				
3	1	NMEDIA				
4	'SVW'	SSP-TOP-WATER-OPT				
5	51 0.0 5000.0	NMESH SIGMA Z(NSSP)				
6	0.0 1548.52 /	Z()	CP()	CS()	RHO()	AP() AS()
7	200.0 1530.29 /	.	.	.	.	.
8	250.0 1526.69 /	.	.	.	.	.
9	400.0 1517.78 /	.	.	.	.	.
10	600.0 1509.49 /	.	.	.	.	.
11	800.0 1504.30 /	.	.	.	.	.
12	1000.0 1501.38 /	.	.	.	.	.
13	1200.0 1500.14 /	.	.	.	.	.
14	1400.0 1500.12 /	.	.	.	.	.
15	1600.0 1501.02 /	.	.	.	.	.
16	1800.0 1502.57 /	.	.	.	.	.
17	2000.0 1504.62 /	.	.	.	.	.



18	2200.0 1507.02 /	.	.	.	.	.	.
19	2400.0 1509.69 /	.	.	.	.	.	.
20	2600.0 1512.55 /	.	.	.	.	.	.
21	2800.0 1515.56 /	.	.	.	.	.	.
22	3000.0 1518.67 /	.	.	.	.	.	.
23	3200.0 1521.85 /	.	.	.	.	.	.
24	3400.0 1525.10 /	.	.	.	.	.	.
25	3600.0 1528.38 /	.	.	.	.	.	.
26	3800.0 1531.70 /	.	.	.	.	.	.
27	4000.0 1535.04 /	.	.	.	.	.	.
28	4200.0 1538.39 /	.	.	.	.	.	.
29	4400.0 1541.76 /	.	.	.	.	.	.
30	4600.0 1545.14 /	.	.	.	.	.	.
31	4800.0 1548.52 /	.	.	.	.	.	.
32	5000.0 1551.91 /	.	.	.	.	.	.
33	'A' 0.0	BOTOPT SIGMA					
34	5000.0 1600.00 0.0 1.8 .0 .0 /	ZB	CPB	CSB	RHOB	APB	ASB
35	1	NSD					
36	1000.0 /	SD(1:NSD) (m)					
37	501	NRD					
38	0.0 5000.0 /	RD(1:NRD) (m)					
39	501	NRR					
40	0.0 101.0 /	RR(1:NRR) (km)					
41	'C'	OPTION: 'R/E/C/A/I/S'					
42	0	NBEAMS ISINGLE					
43	-14.0 14.0 /	ALPHA(1:NBEAMS) (°)					
44	100.0 5500.0 102.0	STEP (m)	ZBOX (m)	RBOX (km)			

现在，运行 **Bellhop**，我们得到名为 **flatwav\_C.shd** 的二进制文件，它实际上包含了经过相干计算的声压。我们用 M-文件 **plotshd.m** 来绘制传播损失图，结果如图 4 所示：

>> **plotshd('flatwav\_C.shd')** <ENTER>

最后，让我们设置 **OPTIONS3(1)='A'** 来修改 **flatwav\_C.env**，如下所示：

flatwav_A.env / Arrives		
行号	内 容	说 明
1	'Munk / Flat / Arrives'	TITLE
2	50.0	FREQ (Hz)
3	1	NMEDIA
4	'SVW'	SSP-TOP-WATER-OPT
5	51 0.0 5000.0	NMESH SIGMA Z(NSSP)
6	0.0 1548.52 /	Z() CP() CS() RHO() AP() AS()
7	200.0 1530.29 /	.
8	250.0 1526.69 /	.
9	400.0 1517.78 /	.
10	600.0 1509.49 /	.
11	800.0 1504.30 /	.
12	1000.0 1501.38 /	.

13	1200.0 1500.14 /	.	.	.	.	.	.
14	1400.0 1500.12 /	.	.	.	.	.	.
15	1600.0 1501.02 /	.	.	.	.	.	.
16	1800.0 1502.57 /	.	.	.	.	.	.
17	2000.0 1504.62 /	.	.	.	.	.	.
18	2200.0 1507.02 /	.	.	.	.	.	.
19	2400.0 1509.69 /	.	.	.	.	.	.
20	2600.0 1512.55 /	.	.	.	.	.	.
21	2800.0 1515.56 /	.	.	.	.	.	.
22	3000.0 1518.67 /	.	.	.	.	.	.
23	3200.0 1521.85 /	.	.	.	.	.	.
24	3400.0 1525.10 /	.	.	.	.	.	.
25	3600.0 1528.38 /	.	.	.	.	.	.
26	3800.0 1531.70 /	.	.	.	.	.	.
27	4000.0 1535.04 /	.	.	.	.	.	.
28	4200.0 1538.39 /	.	.	.	.	.	.
29	4400.0 1541.76 /	.	.	.	.	.	.
30	4600.0 1545.14 /	.	.	.	.	.	.
31	4800.0 1548.52 /	.	.	.	.	.	.
32	5000.0 1551.91 /	.	.	.	.	.	.
33	'A' 0.0	BOTOPT SIGMA					
34	5000.0 1600.00 0.0 1.8 .0 .0 /	ZB	CPB	CSB	RHOB	APB	ASB
35	1	NSD					
36	1000.0 /	SD(1:NSD) (m)					
37	1	NRD					
38	1000.0 /	RD(1:NRD) (m)					
39	1	NRR					
40	101.0 /	RR(1:NRR) (km)					
41	'A'	OPTION: 'R/E/C/A/I/S'					
42	101	NBEAMS ISINGLE					
43	-14.0 14.0 /	ALPHA(1:NBEAMS) (°)					
44	100.0 5500.0 102.0	STEP (m)	ZBOX (m)	RBOX (km)			

运行 Bellhop 之后，得到名为 flatwav\_A.arr 的 ascii 码文件，其中包含到达接收器位置的射线的振幅和传播时间（我们仅指一个位置点，该模型能够计算到达阵列块中所有阵列点的声线的传播时间和振幅）。

包含在 \*.arr 文件中的数据可以使用 M-文件 read\_arrivals\_asc.m 读取：

```
>> [ a, tau, theta0, thetaR, srefl, brefl, narr, rz ] = ...
```

```
read_arrivals_asc( 'flatwav_A.arr' ) <ENTER>
```

### 小贴士：

以上各条命令是基于 2008 年的 Linux 声学工具箱编译并运行的

过程。以下运算过程和各绘图的命令脚本是基于 2018 年 Win10 平台的声学工具箱完成的。这也是读此文本诸君的第一份练习作业。

### 作业一：

品味各环境文件，运行 Bellhop，品鉴\*.prt 文件对输入文件进行反馈和辅导纠错的设计初心，对各运行类型输出进行绘图。

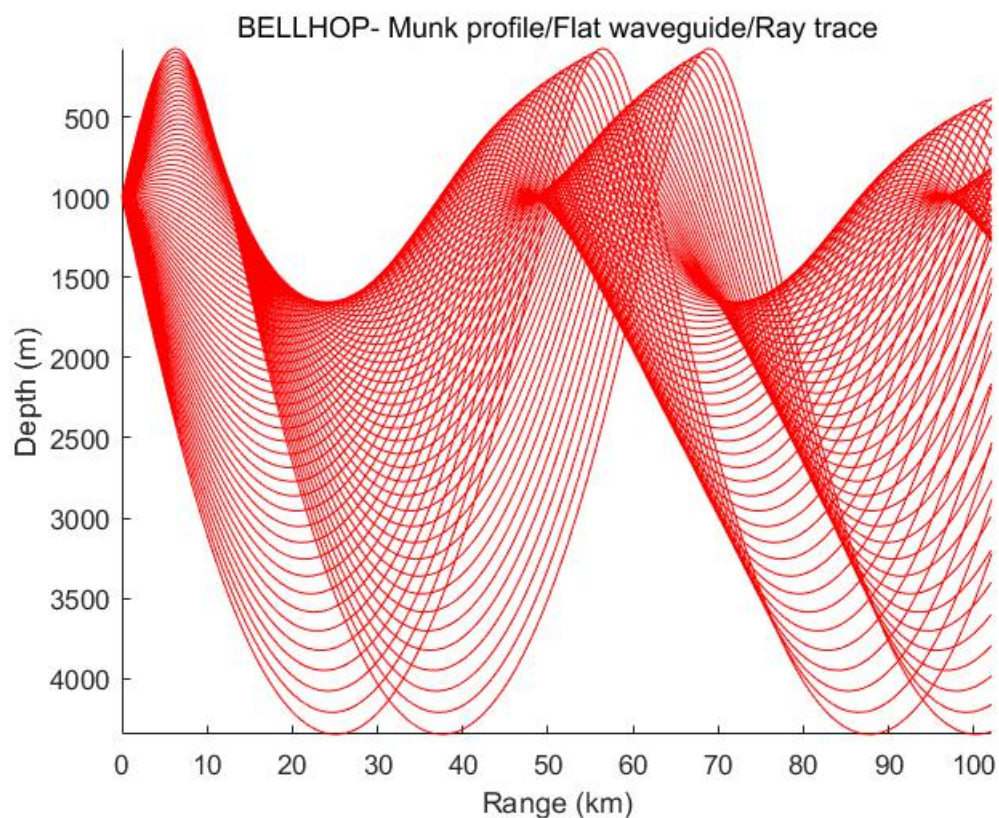


图 2 由 Bellhop 计算的海底水平的深海波导中的声线。

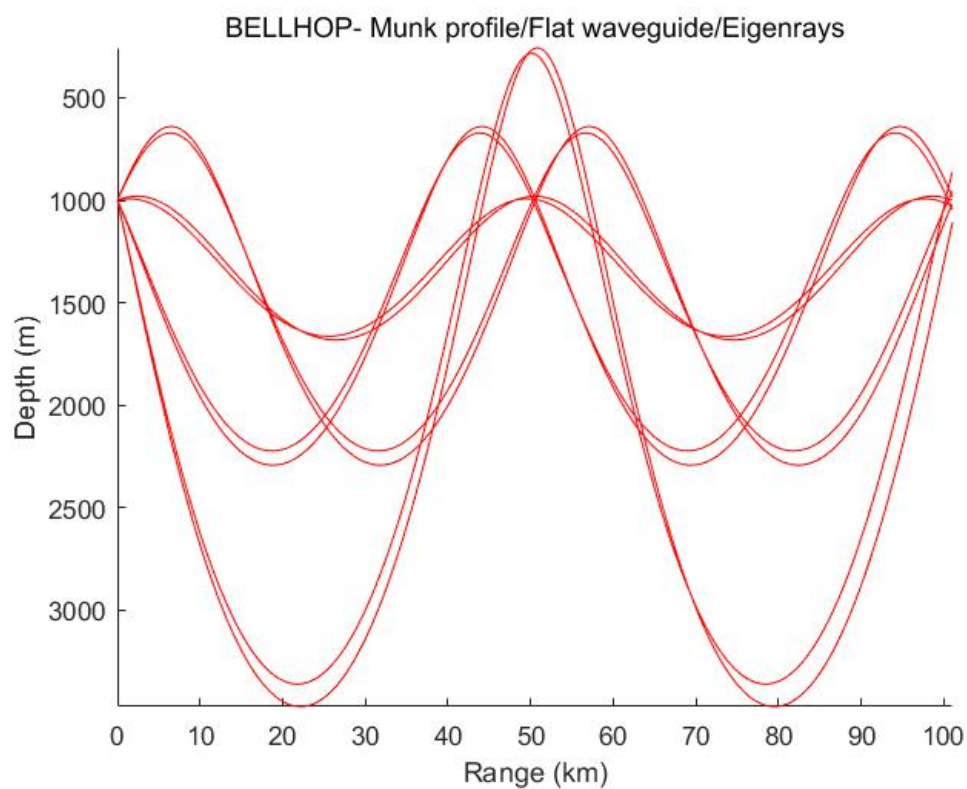


图 3 由 Bellhop 计算的海底水平的深海波导中的本征声线。

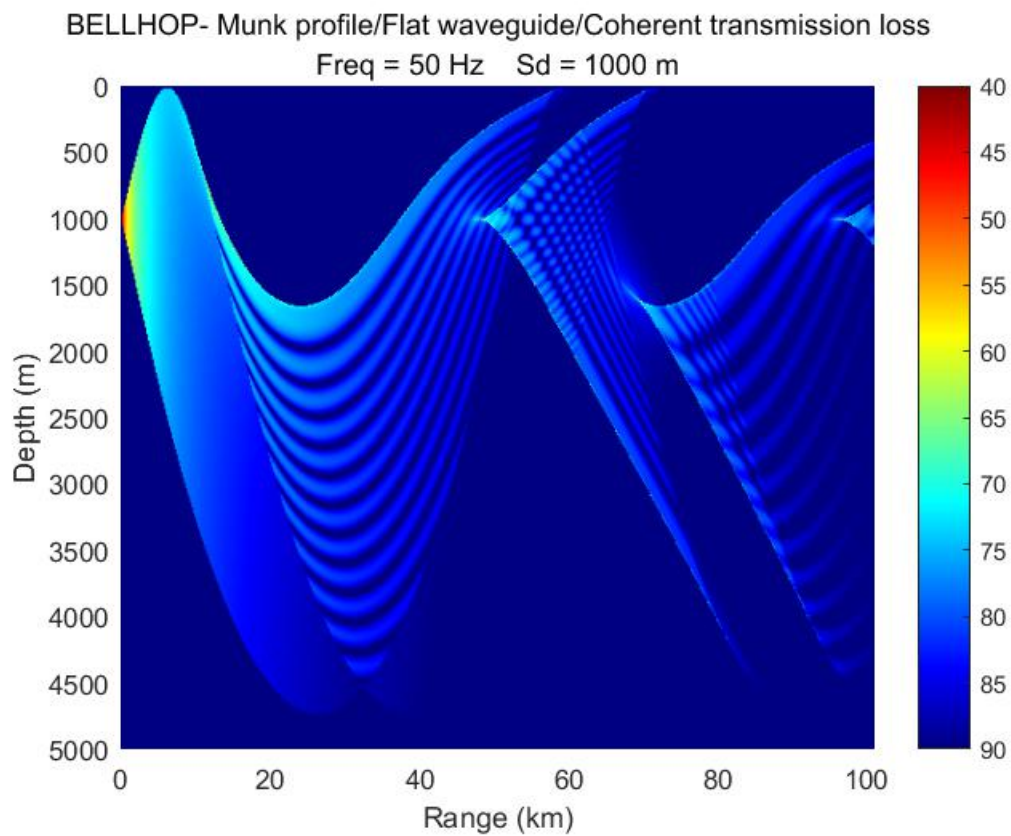


图 4 由 Bellhop 计算的相干传播损失。

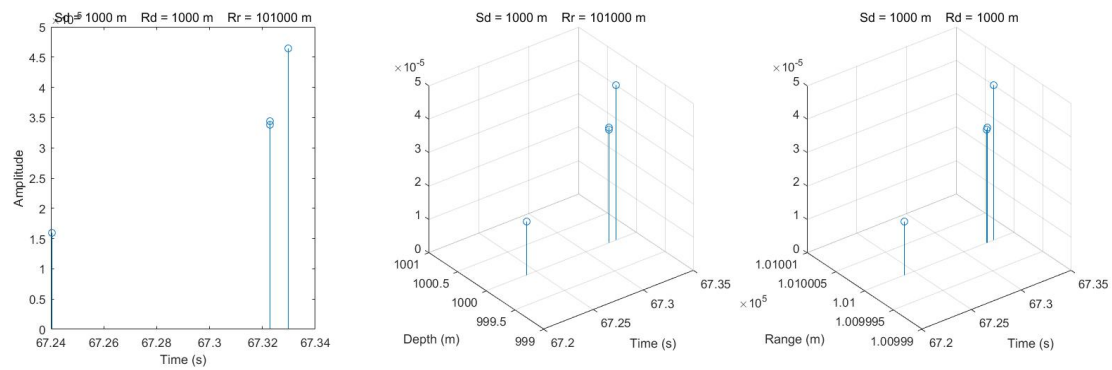


图 5 由 Bellhop 计算到达接收器处的脉冲响应。

Matlab 命令脚本文件为，Munk\_flat.m，代码如下。

文件路径：...\GeneralDescription\Task_GD_1_flat
作业脚本文件名：command_draft_Task1_Munk_flat.m
<pre> clc; clear; close all; global units; units = 'km'; plotssp flatwav_R; bellhop flatwav_R % Runtype = 'R' figure; plotray flatwav_R bellhop flatwav_E % Runtype = 'E' figure; plotray flatwav_E bellhop flatwav_C % Runtype = 'C' figure; plotshd flatwav_C.shd bellhop flatwav_A % Runtype = 'A' % [ Arr, Pos ] = read_arrivals_asc( ARRFile, Narmmx ) [ Arr, Pos ] = read_arrivals_asc( 'flatwav_A.arr' ); % plotarr( filename, irr, ird, isd ) plotarr( 'flatwav_A.arr', 1, 1, 1 ) </pre>

辅助程序和文件：

声线轨迹：flatwav\_R.env; bellhop.m; bellhop.exe; plotray.m。

本征声线：flatwav\_E.env。

相关传播损失：flatwav\_C.env; plotshd.m; read\_shd.m;  
read\_shd\_bin.m; caxisrev.m。

到达声线（脉冲响应）：flatwav\_A.env; read\_arrivals\_asc.m;  
plotarr.m。

```
Munk_flat.m* x +
1      clc;      clear;  close all;
2 -    global units;  units = 'km';
3 -    bellhop flatwav_R  %  Runtype = 'R'
4 -    figure; plotray flatwav_R
5 -    bellhop flatwav_E  %  Runtype = 'E'
6 -    figure; plotray flatwav_E
7 -    bellhop flatwav_C  %  Runtype = 'C'
8 -    figure; plotshd flatwav_C.shd
9 -    bellhop flatwav_A  %  Runtype = 'A'
10 -    % [ Arr, Pos ] = read_arrivals_asc( ARRFile, Narrmx )
11 -    [ Arr, Pos ] = read_arrivals_asc( 'flatwav_A.arr' );
12    % plotarr( filename, irr, ird, isd )
13 -    plotarr( 'flatwav_A.arr', 1, 1, 1 )
```

6.2 高斯海山(变化的海底)

在本小节，我们将描述针对不平海底应用 Bellhop 计算声线的算例。应用高斯函数生成一个理想化的海山，并写入文件 seamount.bty，如下所示：

seamount.bty		
'L'	插值类型	
	点数	
	r()	z()
101		
0 4997.16		
1.01 4997.15	.	.
2.02 4997.15	.	.
3.03 4997.14	.	.
4.04 4997.12	.	.
5.05 4997.1	.	.
. .	.	.
. .	.	.
. .	.	.
95.95 4997.12	.	.
96.96 4997.14	.	.
97.97 4997.15	.	.
98.98 4997.15	.	.
99.99 4997.16	.	.
101 4997.16	.	.

接下来，我们复制 flatwav.env，将其命名为 seamount.env，再进行如下修改：

seamount_R.env		
行号	内 容	说 明
1	'Munk/Sea Mountain/Ray trace'	TITLE
2	50.0	FREQ (Hz)
3	1	NMEDIA
4	'SVW'	SSP-TOP-WATER-OPT
5	51 0.0 5000.0	NMESH SIGMA Z(NSSP)
6	0.0 1548.52 /	Z() CP() CS() RHO() AP() AS()
7	200.0 1530.29 /	.
8	250.0 1526.69 /	.
9	400.0 1517.78 /	.

10	600.0 1509.49 /	.	.	.	.	.	.
11	800.0 1504.30 /	.	.	.	.	.	.
12	1000.0 1501.38 /	.	.	.	.	.	.
13	1200.0 1500.14 /	.	.	.	.	.	.
14	1400.0 1500.12 /	.	.	.	.	.	.
15	1600.0 1501.02 /	.	.	.	.	.	.
16	1800.0 1502.57 /	.	.	.	.	.	.
17	2000.0 1504.62 /	.	.	.	.	.	.
18	2200.0 1507.02 /	.	.	.	.	.	.
19	2400.0 1509.69 /	.	.	.	.	.	.
20	2600.0 1512.55 /	.	.	.	.	.	.
21	2800.0 1515.56 /	.	.	.	.	.	.
22	3000.0 1518.67 /	.	.	.	.	.	.
23	3200.0 1521.85 /	.	.	.	.	.	.
24	3400.0 1525.10 /	.	.	.	.	.	.
25	3600.0 1528.38 /	.	.	.	.	.	.
26	3800.0 1531.70 /	.	.	.	.	.	.
27	4000.0 1535.04 /	.	.	.	.	.	.
28	4200.0 1538.39 /	.	.	.	.	.	.
29	4400.0 1541.76 /	.	.	.	.	.	.
30	4600.0 1545.14 /	.	.	.	.	.	.
31	4800.0 1548.52 /	.	.	.	.	.	.
32	5000.0 1551.91 /	.	.	.	.	.	.
33	'A*' 0.0	BOTOPT SIGMA					
34	5000.0 1600.00 0.0 1.8 .0 0.0	ZB	CPB	CSB	RHOB	APB	ASB
35	1	NSD					
36	1000.0 /	SD(1:NSD) (m)					
37	1	NRD					
38	1000.0 /	RD(1:NRD) (m)					
39	1	NRR					
40	101.0 /	RR(1:NRR ) (km)					
41	'R'	OPTION: 'R/E/C/A/I/S'					
42	71	NBEAMS ISINGLE					
43	-14.0 14.0 /	ALPHA(1:NBEAMS) (°)					
44	100.0 5500.0 102.0	STEP (m)	ZBOX (m)	RBOX (km)			

运行 Bellhop 之后，用 plotrays.m 绘制声线(如图 6 所示)，仿照前面算例，将 OPTIONS3(1) = 'R'改为 OPTIONS3(1) = 'E'和 OPTIONS3(1) = 'C'，就能够分别计算得到本征声线和相干传播损失。但是，须记住，对每个选项都应相应地写入阵列块。



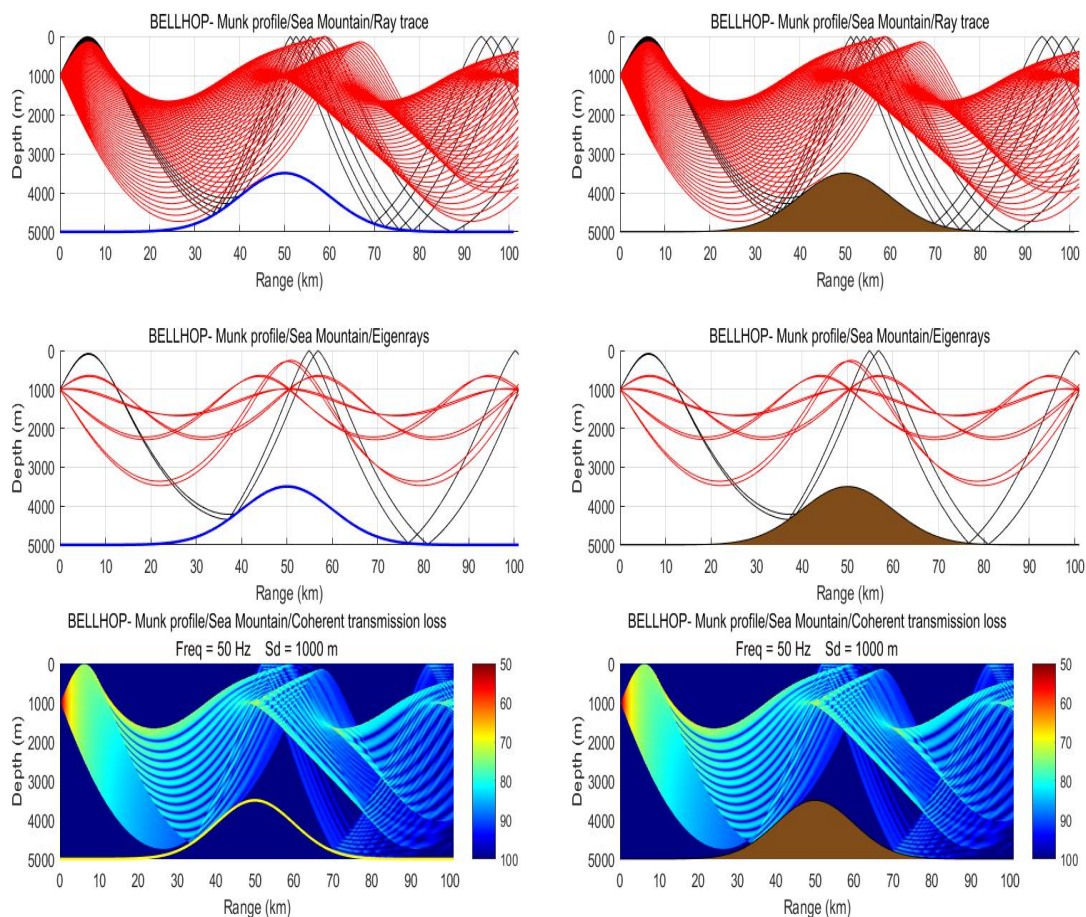


图 6：在有高斯海山的深海波导中，由 Bellhop 计算的声线。

## 作业二：

生成海山文件，并对海山绘图；仔细阅读和修改 env 文件，运行 Bellhop，并随心所欲完成图 6 中所有绘图。

## Answer (答案)：

Win10 平台下的 Matlab 命令脚本：

文件路径：...\GeneralDescription\Task\_GD\_2\_Gauss

作业脚本文件名：command\_draft\_Task2\_MountainLine3Fig.m

```
clc; clear all; close all;
global units; units = 'km';
```

```
a = 5; sigma = 1;
x = linspace( 0,10.1,101 );
```

```
y = (1/((sqrt(2*pi)) * sigma)) * exp(-((x-a).^2) / (2*sigma.^2));  
y = 4997.1624 - y / max(y) * 1500;
```

```
fid = fopen('seamount.bty','wt');  
fprintf(fid,'%1s%1s%1s\n',char(39),'L',char(39));  
fprintf(fid,'%3d\n',length(y));  
for mi = 1 : length(y)  
    fprintf(fid,'%g %g \n',x(mi)*10,y(mi));  
end  
fclose(fid);
```

```
subplot(321); bellhop('seamount_R');  
plotray('seamount_R');ylim([0 5000])  
hold on; grid on;  
plot(x*1e1,y,'b','LineWidth',1.5);
```

```
subplot(323); bellhop('seamount_E');  
plotray('seamount_E');ylim([0 5000])  
hold on; grid on;  
plot(x*1e1,y,'b','LineWidth',1.5);
```

```
subplot(325); bellhop('seamount_C');  
plotshd('seamount_C.shd');ylim([0 5000])  
hold on;  
plot(x*1e1,y,'y','LineWidth',1.5);
```

```
subplot(322); bellhop('seamount_R');  
plotray('seamount_R');ylim([0 5000])  
plotbty('seamount_R'); grid on;
```

```
subplot(324); bellhop('seamount_E');  
plotray('seamount_E');ylim([0 5000])  
plotbty('seamount_E'); grid on;
```

```
subplot(326); bellhop('seamount_C');  
plotshd('seamount_C.shd');ylim([0 5000])  
plotbty('seamount_C');
```

```

MountainLine3Fig.m  x  +
1 -   clc; clear all; close all;
2 -   global units; units = 'km';
3
4 -   a = 5; sigma = 1;
5 -   x = linspace( 0,10.1,101 );
6 -   y = (1/((sqrt(2*pi)) * sigma)) * exp(-((x-a).^2) / (2*sigma.^2));
7 -   y = 4997.1624 - y / max(y) * 1500;
8
9 -   fid = fopen('seamount.bty','wt');
10 -  fprintf(fid,'%1s%1s%1s\n',char(39),'L',char(39));
11 -  fprintf(fid,'%3d\n',length(y));
12 -  for mi = 1 : length(y)
13 -      fprintf(fid,'%g %g \n',x(mi)*10,y(mi));
14 -  end
15 -  fclose(fid);
16
17 -  subplot(321); bellhop('seamount_R');
18 -  plotray('seamount_R');ylim([0 5000])
19 -  hold on; grid on;
20 -  plot(x*1e1,y,'b','LineWidth',1.5);
21
22 -  subplot(323); bellhop('seamount_E');
23 -  plotray('seamount_E');ylim([0 5000])
24 -  hold on; grid on;
25 -  plot(x*1e1,y,'b','LineWidth',1.5);
26
27 -  subplot(325); bellhop('seamount_C');
28 -  plotshd('seamount_C.shd');ylim([0 5000])
29 -  hold on;
30 -  plot(x*1e1,y,'y','LineWidth',1.5);
31
32 -  subplot(322); bellhop('seamount_R');
33 -  plotray('seamount_R');ylim([0 5000])
34 -  plotbty('seamount_R'); grid on;
35
36 -  subplot(324); bellhop('seamount_E');
37 -  plotray('seamount_E');ylim([0 5000])
38 -  plotbty('seamount_E'); grid on;
39
40 -  subplot(326); bellhop('seamount_C');
41 -  plotshd('seamount_C.shd');ylim([0 5000])
42 -  plotbty('seamount_C');

```

所需辅助文件和程序：

seamount\_R.env; seamount\_E.env; seamount\_C.env;

seamount\_R.bty; seamount\_E.bty; seamount\_E.bty;

plotbty.m; readbty.m。

6.3 变化的边界

Bellhop 不仅能够处理变化的海底，还能够同时处理海面、海底都起伏变化的场景。实现如下：将 seamount.bty 制作一个副本并命名为 varbounds.bty，然后，创建包含波浪表面坐标的 varbounds.atl 文件，如下所示：

varbounds.atl		
L'	插值类型	
	点数	
	r()	z()
101		
0.00 41.2215		
1.01 69.0983	.	.
2.02 100	.	.
3.03 130.902	.	.
4.04 158.779	.	.
5.05 180.902	.	.
.	.	.
.	.	.
.	.	.
95.95 19.0983	.	.
96.96 4.89435	.	.
97.97 0	.	.
98.98 4.89435	.	.
99.99 19.0983	.	.
101.0 41.2215	.	.

然后，将 seamount.env 复制为 varbounds.env，并修改如下：

Varbounds_C.env		
行号	内 容	说 明
1	'Munk/ Varbounds /Coherent TL'	TITLE
2	50.0	FREQ (Hz)
3	1	NMEDIA
4	'SVW *'	SSP-TOP-WATER-OPT
5	51 0.0 5000.0	NMESH SIGMA Z(NSSP)
6	0.0 1548.52 /	Z() CP() CS() RHO() AP() AS()
7	200.0 1530.29 /	.
8	250.0 1526.69 /	.
9	400.0 1517.78 /	.
10	600.0 1509.49 /	.
11	800.0 1504.30 /	.

12	1000.0 1501.38 /	.	.	.	.	.	.
13	1200.0 1500.14 /	.	.	.	.	.	.
14	1400.0 1500.12 /	.	.	.	.	.	.
15	1600.0 1501.02 /	.	.	.	.	.	.
16	1800.0 1502.57 /	.	.	.	.	.	.
17	2000.0 1504.62 /	.	.	.	.	.	.
18	2200.0 1507.02 /	.	.	.	.	.	.
19	2400.0 1509.69 /	.	.	.	.	.	.
20	2600.0 1512.55 /	.	.	.	.	.	.
21	2800.0 1515.56 /	.	.	.	.	.	.
22	3000.0 1518.67 /	.	.	.	.	.	.
23	3200.0 1521.85 /	.	.	.	.	.	.
24	3400.0 1525.10 /	.	.	.	.	.	.
25	3600.0 1528.38 /	.	.	.	.	.	.
26	3800.0 1531.70 /	.	.	.	.	.	.
27	4000.0 1535.04 /	.	.	.	.	.	.
28	4200.0 1538.39 /	.	.	.	.	.	.
29	4400.0 1541.76 /	.	.	.	.	.	.
30	4600.0 1545.14 /	.	.	.	.	.	.
31	4800.0 1548.52 /	.	.	.	.	.	.
32	5000.0 1551.91 /	.	.	.	.	.	.
33	'A*' 0.0	BOTOPT SIGMA					
34	5000.0 1600.00 0.0 1.8 .0 .0 /	ZB	CPB	CSB	RHOB	APB	ASB
35	1	NSD					
36	1000.0 /	SD(1:NSD) (m)					
37	501	NRD					
38	0.0 5000.0 /	RD(1:NRD) (m)					
39	501	NRR					
40	0.0 101.0 /	RR(1:NRR) (km)					
41	'C'	OPTION: 'R/E/C/A/I/S'					
42	0	NBEAMS ISINGLE					
43	-14.0 14.0 /	ALPHA(1:NBEAMS) (°)					
44	100.0 5500.0 102.0	STEP (m)	ZBOX (m)	RBOX (km)			

用这个输入文件来运行 Bellhop，就得到图 7。

### 作业三：

生成海面坐标文件，绘制海面测高图。仿照作业一生成一批环境文件，按照以上提示生成对应的海面、海底起伏文件，运行 Bellhop 并参考图 7 随心所欲地绘图。

Answer（答案）：

Win10 平台下的 Matlab 作业脚本文件为 VariousBounds.m。其中四幅子图汇集在一起，所用到的辅助文件：

varbounds\_R.env; varbounds\_R.bty; varbounds\_R.atl;

varbounds\_E.env; varbounds\_E.bty; varbounds\_E.atl;

varbounds\_C.env; varbounds\_C.bty; varbounds\_C.atl;

程序: plotati.m; 辅助程序: readati.m

程序: plotssp.m; 辅助程序: read\_env.m; read\_env\_core.m;

readvector.m; topbot.m; crci.m; readsdrd.m 等。

文件路径: ...\\GeneralDescription\\Task\_GD\_3\_VB

作业脚本文件名: command\_draft\_Task3\_VariousBounds.m

```
clc; clear all; close all;
global units; units = 'km';
%===== Sea Surface =====
xs = linspace(0,10*pi,101);
ys = 100 + sin(xs - pi/5) * 100;
xs = xs / max(xs) * 101;
%===== Write the Sea Surface file =====
% fid = fopen('varbounds_R.atl','wt');
% fprintf(fid,'%1s%1s%1s\n',char(39),'L',char(39));
% fprintf(fid,'%3d\n',length(ys));
% for mi = 1 : length(ys)
%     fprintf(fid,'%g %g \n',xs(mi),ys(mi));
% end
% fclose(fid);
%===== Sea Bottom =====
a = 5; sigma = 1;
x = linspace(0,10.1,101);
y = (1/((sqrt(2*pi)) * sigma)) * exp(-((x-a).^2)/(2*sigma.^2));
y = 4997.1624 - y / max(y) * 1500;
%===== Write the Sea Bottom file =====
% fid = fopen('varbounds_R.bty','wt');
% fprintf(fid,'%1s%1s%1s\n',char(39),'L',char(39));
% fprintf(fid,'%3d\n',length(y));
% for mi = 1 : length(y)
%     fprintf(fid,'%g %g \n',x(mi)*10,y(mi));
% end
% fclose(fid);
%===== Calculating and Plotting =====
```

```

subplot(3,6,2.5:6); bellhop('varbounds_R');
plotray('varbounds_R');ylim([0 5000])
hold on;    grid on;
plotati('varbounds_E');plotbty('varbounds_E')
% plot(x*1e4,y,'b','LineWidth',1.5);
% plot(xs*1e3,ys,'b','LineWidth',1.5);

subplot(3,6,8.5:12); bellhop('varbounds_E');
plotray('varbounds_E');ylim([0 5000])
hold on;    grid on;
plotati('varbounds_E');plotbty('varbounds_E'); xlabel("");
% plot(x*1e4,y,'b','LineWidth',1.5);
% plot(xs*1e3,ys,'b','LineWidth',1.5);

subplot(3,6,14.5:18); bellhop('varbounds_C');
plotshd('varbounds_C.shd');ylim([0 5000])
hold on;    grid on;
plotati('varbounds_E');plotbty('varbounds_E')
% plot(x*1e4,y,'y','LineWidth',1.5);
% plot(xs*1e3,ys,'y','LineWidth',1.5);

%===== Other Plottings =====
subplot(3,6,[1 7 13]);plotssp('varbounds_C.env')

```

### 小贴士：

运行 plotssp.m 读 Bellhop 的环境\*.env 文件的阵列模块时，曾经跳行了，输出的\*.prt 提醒有误。经必要的修正后，顺利完成作业。



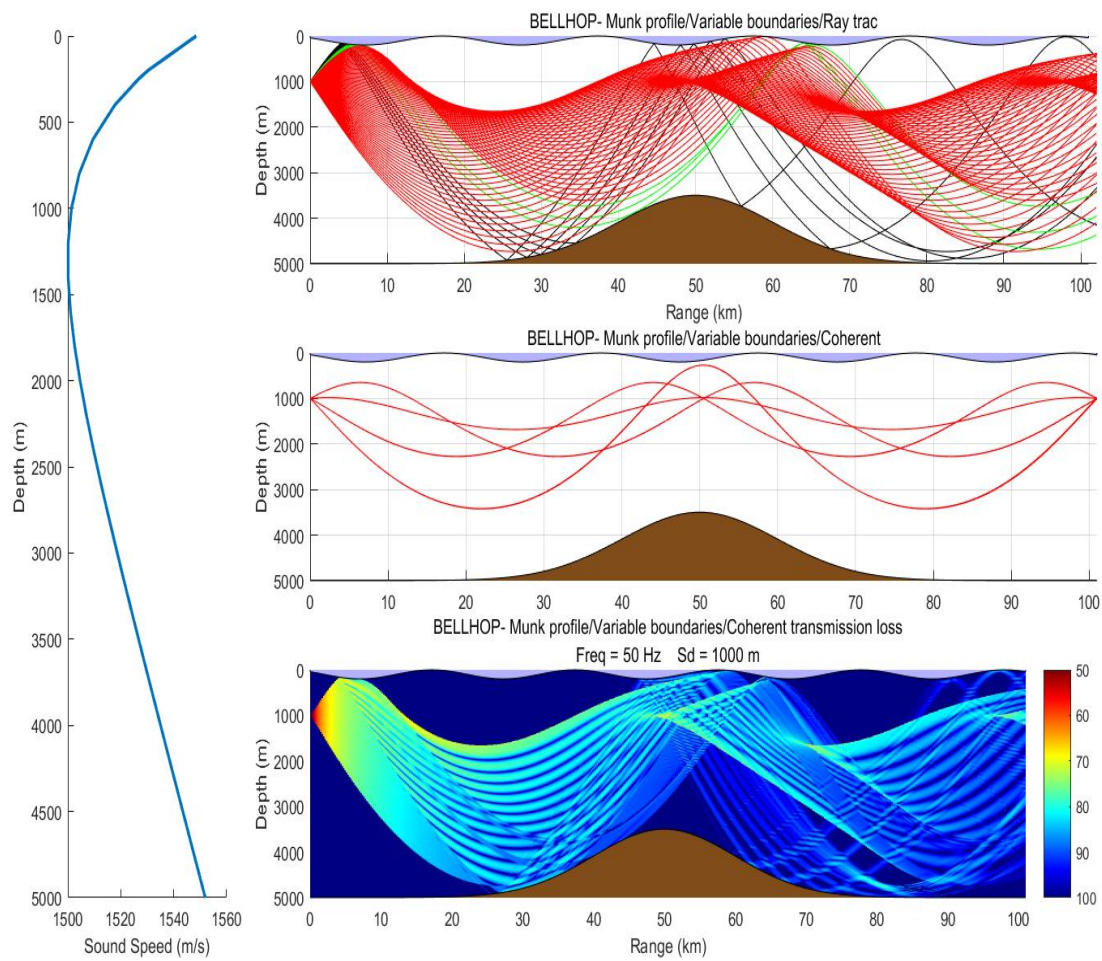


图 7 在有波浪海面和高斯海山的深海波导中, 由 Bellhop 计算的声线。

```

VariousBounds.m  ✕  +
1 -   clc;    clear all; close all;
2 -   global units;    units = 'km';
3 -   %===== Sea Surface =====
4 -   xs = linspace(0,10*pi,101);
5 -   ys = 100 + sin(xs - pi/5) * 100;
6 -   xs = xs / max(xs) *101;
7 -   %===== Write the Sea Surface file =====
8 -   % fid = fopen('varbounds_R.ati','wt');
9 -   % fprintf(fid,'%ls%ls%ls\n',char(39),'L',char(39));
10 -  % fprintf(fid,'%3d\n',length(ys));
11 -  % for mi = 1 : length(ys)
12 -  %     fprintf(fid,'%g %g \n',xs(mi),ys(mi));
13 -  % end
14 -  % fclose(fid);
15 -  %===== Sea Bottom =====
16 -  a = 5;    sigma = 1;
17 -  x = linspace( 0,10.1,101 );
18 -  y = (1/((sqrt(2*pi)) * sigma)) * exp(-(x-a).^2)/(2*sigma.^2));
19 -  y = 4997.1624 - y / max(y) * 1500;
20 -  %===== Write the Sea Bottom file =====
21 -  % fid = fopen('varbounds_R.bty','wt');
22 -  % fprintf(fid,'%ls%ls%ls\n',char(39),'L',char(39));
23 -  % fprintf(fid,'%3d\n',length(y));
24 -  % for mi = 1 : length(y)
25 -  %     fprintf(fid,'%g %g \n',x(mi)*10,y(mi));
26 -  % end
27 -  % fclose(fid);
28 -  %===== Calculating and Plotting =====
29 -  subplot(3,6,2.5:6); bellhop('varbounds_R');
30 -  plotray('varbounds_R');ylim([0 5000])
31 -  hold on;    grid on;
32 -  plotati('varbounds_E');plotbty('varbounds_E')
33 -  % plot(x*1e4,y,'b','LineWidth',1.5);
34 -  % plot(xs*1e3,ys,'b','LineWidth',1.5);
35 -
36 -  subplot(3,6,8.5:12); bellhop('varbounds_E');
37 -  plotray('varbounds_E');ylim([0 5000])
38 -  hold on;    grid on;
39 -  plotati('varbounds_E');plotbty('varbounds_E'); xlabel('');
40 -  % plot(x*1e4,y,'b','LineWidth',1.5);
41 -  % plot(xs*1e3,ys,'b','LineWidth',1.5);
42 -
43 -  subplot(3,6,14.5:18); bellhop('varbounds_C');
44 -  plotshd('varbounds_C.shd');ylim([0 5000])
45 -  hold on;    grid on;
46 -  plotati('varbounds_E');plotbty('varbounds_E')
47 -  % plot(x*1e4,y,'y','LineWidth',1.5);
48 -  % plot(xs*1e3,ys,'y','LineWidth',1.5);
49 -
50 -  %===== Other Plottings =====
51 -  subplot(3,6,[1 7 13]);plotssp('varbounds_C.env')

```

6.4 声速场中的声线轨迹

最新版本的 Bellhop 允许计算声速场中的声场，这里仅用声线计算来展开介绍。我们创建一个 gulf.env 输入文件，如下所示：

gulf.env / Ray trace						
行号	内 容	说 明				
1	'Gulf sound speed field/Ray trace'	TITLE				
2	50.0	FREQ (Hz)				
3	1	NMEDIA				
4	'QVW'	SSP-TOP-WATER-OPT				
5	0 0.0 5000.0	NMESH SIGMA Z(NSSP)				
6	0.0 1536.00 /	Z()	CP()	CS()	RHO()	AP() AS()
7	200.0 1506.00 /	.	.	.	.	.
8	700.0 1503.00 /	.	.	.	.	.
9	800.0 1508.00 /	.	.	.	.	.
10	1200.0 1508.00 /	.	.	.	.	.
11	1500.0 1497.00 /	.	.	.	.	.
12	2000.0 1500.00 /	.	.	.	.	.
13	3000.0 1512.00 /	.	.	.	.	.
14	4000.0 1528.00 /	.	.	.	.	.
15	5000.0 1545.00 /	.	.	.	.	.
16	'A' 0.0	BOTOPT SIGMA				
17	5000.00 1800.0 0.0 2.0 0.1 0.0 /	ZB	CPB	CSB	RHOB	APB ASB
18	1	NSD				
19	300.0 /	SD(1:NSD) (m)				
20	101	NRD				
21	0.0 5000.0 /	RD(1:NRD) (m)				
22	1001	NRR				
23	0.0 200.0 /	RR(1:NRR ) (km)				
24	'R'	OPTION: 'R/E/C/A/I/S'				
25	51	NBEAMS ISINGLE				
26	-10.0 10.0 /	ALPHA(1:NBEAMS) (°)				
27	0.0 5500.0 201.0	STEP (m)	ZBOX (m)	RBOX (km)		

设置 OPTIONS1(1) = 'Q', 就允许考虑声速场, 如文件 Gulf.ssp 所示(其语法已经自明)：

Gulf.ssp								
9								
0.0	12.5	25.0	37.5	50.0	75.0	100.0	125.0	201
1536	1536	1536	1536	1536	1536	1536	1536	1536
1506	1508.75	1511.5	1514.25	1517	1520	1524	1528	1528
1503	1503	1503	1502.75	1502.5	1502	1502	1502	1502
1508	1507	1506	1505	1504	1503	1501.5	1500	1500
1508	1506.6	1505	1503.75	1502.5	1500.5	1499	1497	1497

1497	1497	1497	1497	1497	1497	1497	1497	1497
1500	1500	1500	1500	1500	1500	1500	1500	1500
1512	1512	1512	1512	1512	1512	1512	1512	1512
1528	1528	1528	1528	1528	1528	1528	1528	1528
1545	1545	1545	1545	1545	1545	1545	1545	1545

用此输入文件来运行 Bellhop，就得到图 8。

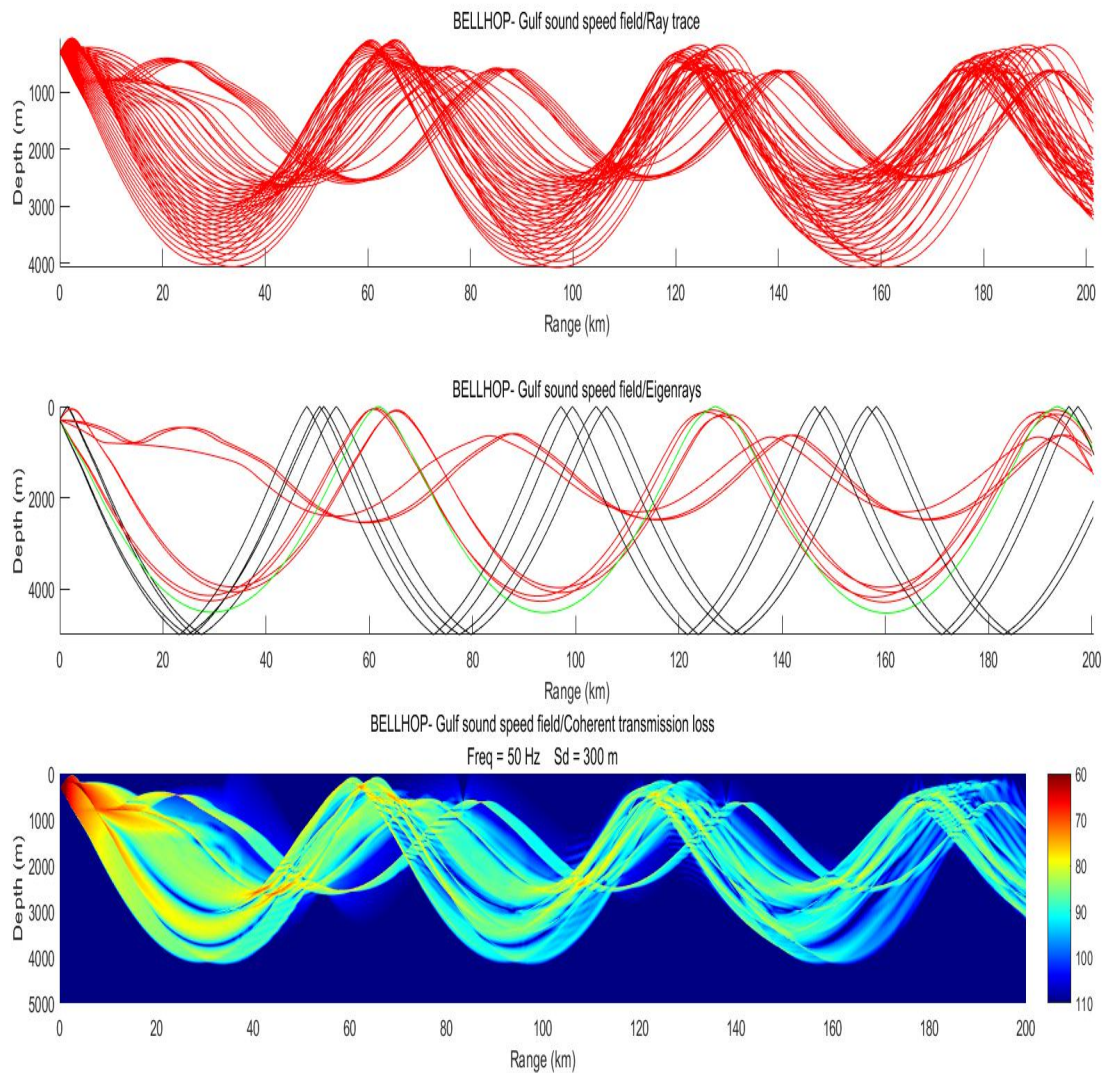


图 8 由 Bellhop 计算的声速场中的声线。

#### 作业四：

熟悉文件 Gulf.ssp，并绘制声速场。然后设置 OPTIONS3(1)，分别绘制图 8 所示的追踪声线、本征声线和相干声压场。

Answer（答案）：Win10 平台下的 Matlab 命令脚本：

文件路径: ...\\GeneralDescription\\Task\_GD\_4\_SSP\_Field

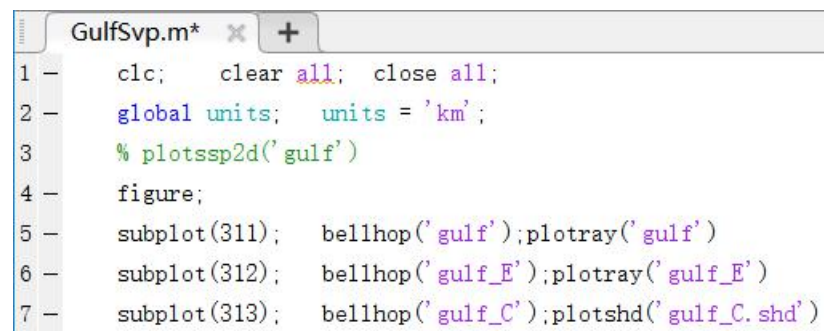
作业脚本文件名: command\_draft\_Task4\_GulfSvp.m

```
clc;    clear all;  close all;  
global units;  units = 'km';  
% plotssp2d('gulf')  
figure;  
subplot(311);  bellhop('gulf');plotray('gulf')  
subplot(312);  bellhop('gulf_E');plotray('gulf_E')  
subplot(313);  bellhop('gulf_C');plotshd('gulf_C.shd')
```

绘制声速场的相关文件:

主文件: plotssp2d.m

辅助文件: readssp2d.m; read\_bell.m; readr.m。



```
GulfSvp.m*  X  +  
1 -  clc;    clear all;  close all;  
2 -  global units;  units = 'km';  
3 -  % plotssp2d('gulf')  
4 -  figure;  
5 -  subplot(311);  bellhop('gulf');plotray('gulf')  
6 -  subplot(312);  bellhop('gulf_E');plotray('gulf_E')  
7 -  subplot(313);  bellhop('gulf_C');plotshd('gulf_C.shd')
```



6.5 声源附近的声场

前几节的例子应用 Bellhop 的默认选项计算了声压,即几何波束,有时在特定的应用中不够精确。对此效果, Bellhop 提供了额外的近似方法来提高精确度。为了展示这种改进,我们采用不同的近似方法来计算声源附近的相干传播损失,并用得到的干涉图案来揭示所用近似方法的精确性。首先, 将 flatwav.env 复制成 nearsource.env, 如下所示:

nearsource_3C.env						
行号	内 容	说 明				
1	'Munk /Near field /(3)C /(4)MS'	TITLE				
2	50.0	FREQ (Hz)				
3	1	NMEDIA				
4	'SVF'	SSP-TOP-WATER-OPT				
5	51 0.0 5000.0	NMESH SIGMA Z(NSSP)				
6	0.0 1548.52 /	Z()	CP()	CS()	RHO()	AP() AS()
7	200.0 1530.29 /	.	.	.	.	.
8	250.0 1526.69 /	.	.	.	.	.
9	400.0 1517.78 /	.	.	.	.	.
10	600.0 1509.49 /	.	.	.	.	.
11	800.0 1504.30 /	.	.	.	.	.
12	1000.0 1501.38 /	.	.	.	.	.
13	1200.0 1500.14 /	.	.	.	.	.
14	1400.0 1500.12 /	.	.	.	.	.
15	1600.0 1501.02 /	.	.	.	.	.
16	1800.0 1502.57 /	.	.	.	.	.
17	2000.0 1504.62 /	.	.	.	.	.
18	2200.0 1507.02 /	.	.	.	.	.
19	2400.0 1509.69 /	.	.	.	.	.
20	2600.0 1512.55 /	.	.	.	.	.
21	2800.0 1515.56 /	.	.	.	.	.
22	3000.0 1518.67 /	.	.	.	.	.
23	3200.0 1521.85 /	.	.	.	.	.
24	3400.0 1525.10 /	.	.	.	.	.
25	3600.0 1528.38 /	.	.	.	.	.

26	3800.0 1531.70 /	.	.	.	.	.	.
27	4000.0 1535.04 /	.	.	.	.	.	.
28	4200.0 1538.39 /	.	.	.	.	.	.
29	4400.0 1541.76 /	.	.	.	.	.	.
30	4600.0 1545.14 /	.	.	.	.	.	.
31	4800.0 1548.52 /	.	.	.	.	.	.
32	5000.0 1551.91 /	.	.	.	.	.	.
33	'A' 0.0	BOTOPT SIGMA					
34	5000.0 1600.00 0.0 1.8 .8 0.0	ZB	CPB	CSB	RHOB	APB	ASB
35	1	NSD					
36	1000.0 /	SD(1:NSD) (m)					
37	501	NRD					
38	0.0 2000.0 /	RD(1:NRD) (m)					
39	501	NRR					
40	0.2 10.0 /	RR(1:NRR ) (km)					
41	'C'	OPTION: 'R/E/C/A/I/S'					
42	201	NBEAMS ISINGLE					
43	-25.0 25.0 /	ALPHA(1:NBEAMS) (°)					
44	0.0 5500.0 102.0	STEP (m)		ZBOX (m)		RBOX (km)	
45	'MS' 1.0 100.0 0	OPTIONS4		epmult rloop		isingl	
46	3 5	nimage		ibwin		component	

我们注意到额外增加了两行(实际上就是增加“**波束版块**”)。当使用几何波束，即设置 `OPTIONS3(1) = 'C'` 时，这些行就省去了，我们在此将这两行提前包含在内，是为了方便后面自动过渡到其他的近似方法。应用 `nearsource.env` 文件运行 `Bellhop`，得到图 9。如图所示，声场严格限制在传播声线之间，声压的干涉模式只在离声源很远的地方才能观察到。使用高斯波束也会得到类似的结果，即在 `nearsource.env` 中将 `OPTIONS3(1) = 'C'` 改为 `OPTIONS3(1:2) = 'CB'`。

通过设置 `OPTIONS3(1:2) = 'CC'` 和 `OPTIONS4(1:2) = 'MS'`，采用笛卡尔坐标，就会发现波束的影响，并得到完全不同的干涉模式。图 10 所示的声场展示了贯穿整个水体的精确的干涉模式。采用中心射线的坐标 (`OPTIONS3(1:2) = 'CR'`) 也可得到相似的结果 (如图 11 所

示），虽然幅度有轻微差异。当然，干涉模式的结构取决于波束的曲率类型。比如，设置 `OPTIONS4(1:2) = 'CZ'`，就得到图 12。

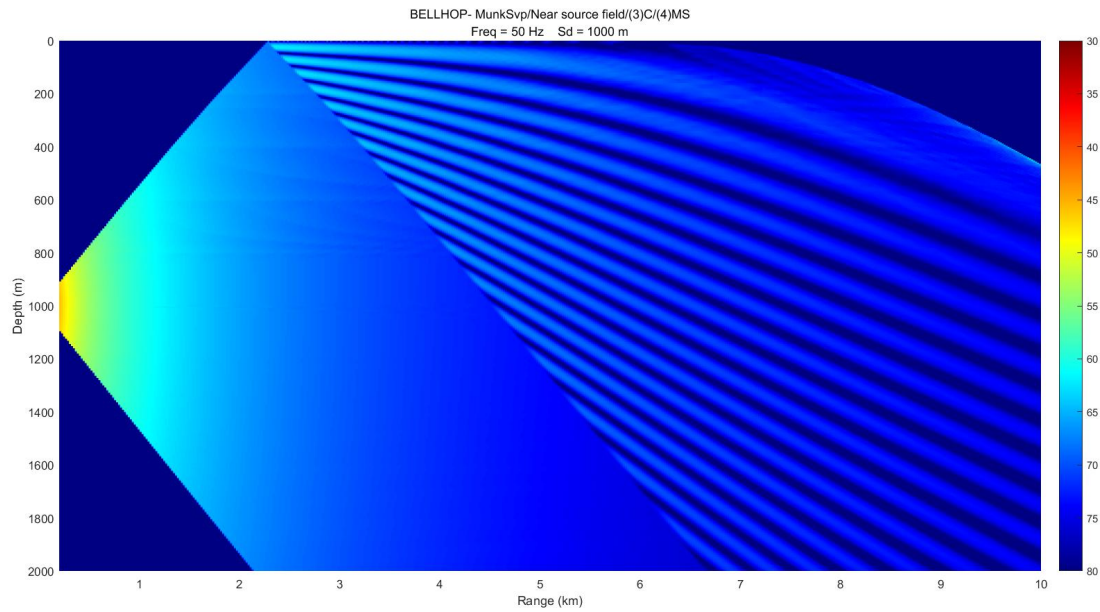


图 9：Bellhop 采用几何波束计算得到的声源附近的相干传播损失。

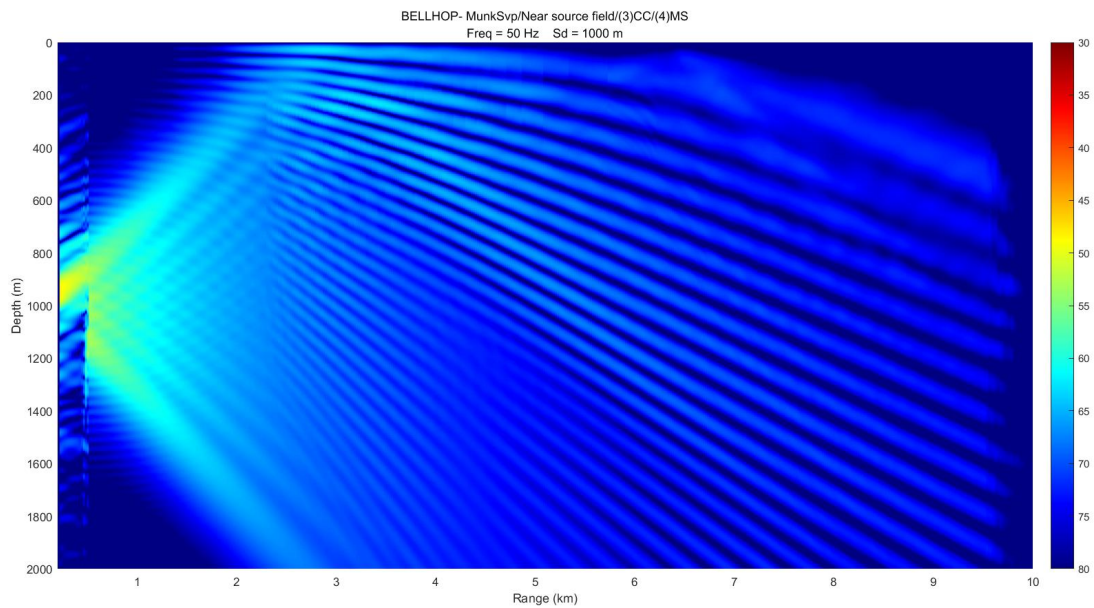


图 10：Bellhop 采用笛卡尔坐标计算声源附近的相干传播损失，以展示波束的影响。波束类型（`OPTIONS4`）为'MS'，`OPTIONS3 = 'CC'`。



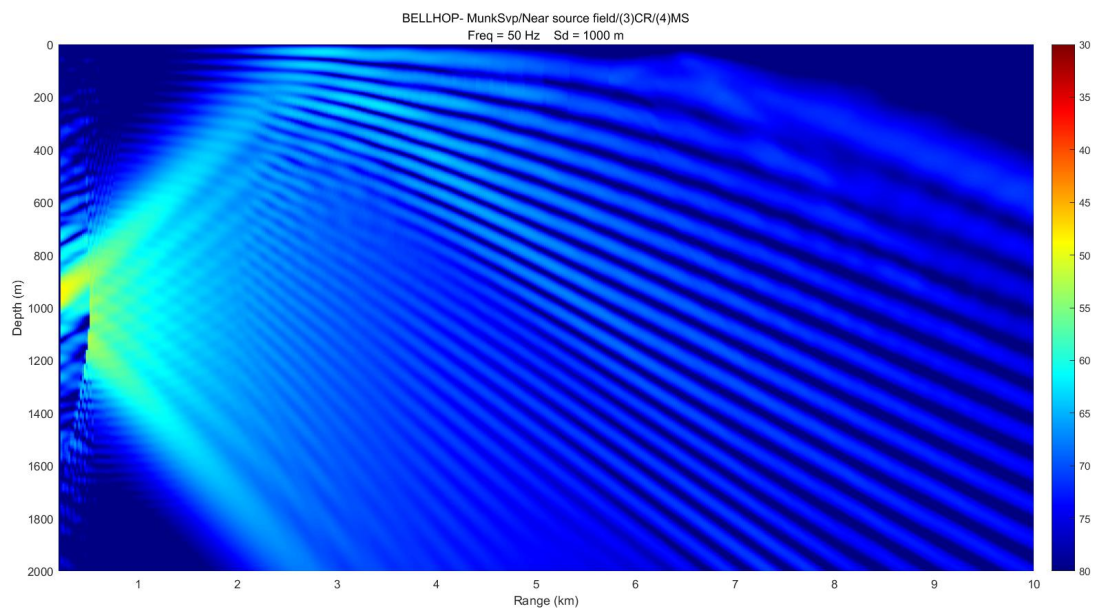


图 11: Bellhop 采用中心声线的坐标计算声源附近的相干传播损失，以展示波束的影响。波束类型为'MS'， $\text{OPTIONS3}(1:2) = \text{'CR'}$ 。

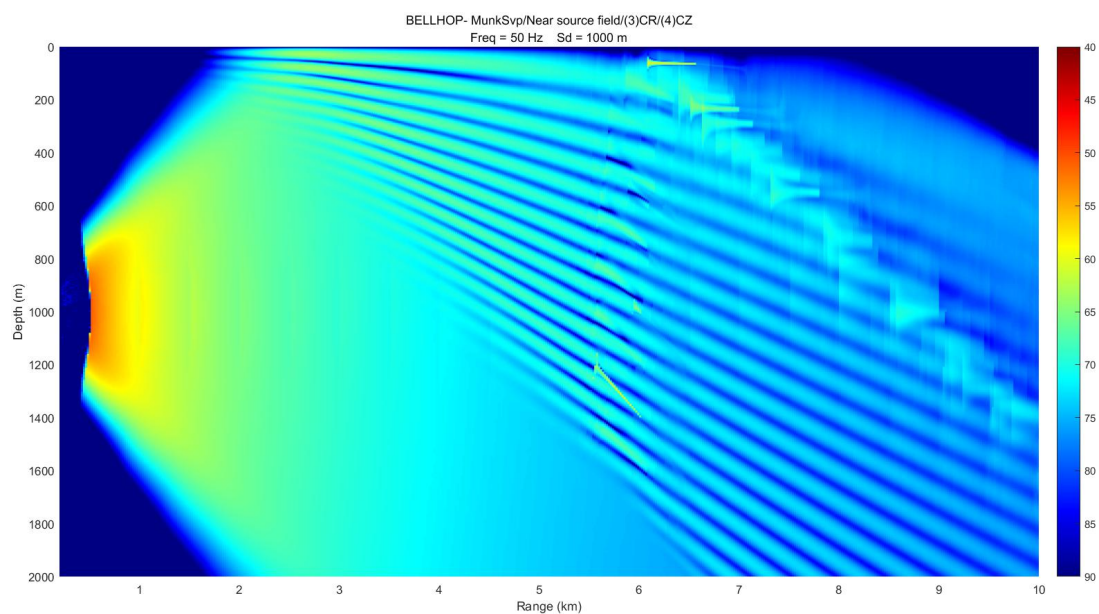


图 12: Bellhop 采用笛卡尔坐标计算声源附近的相干传播损失，以展示波束的影响。波束类型为'CZ'， $\text{OPTIONS3}(1:2) = \text{'CR'}$ 。

### 作业五:

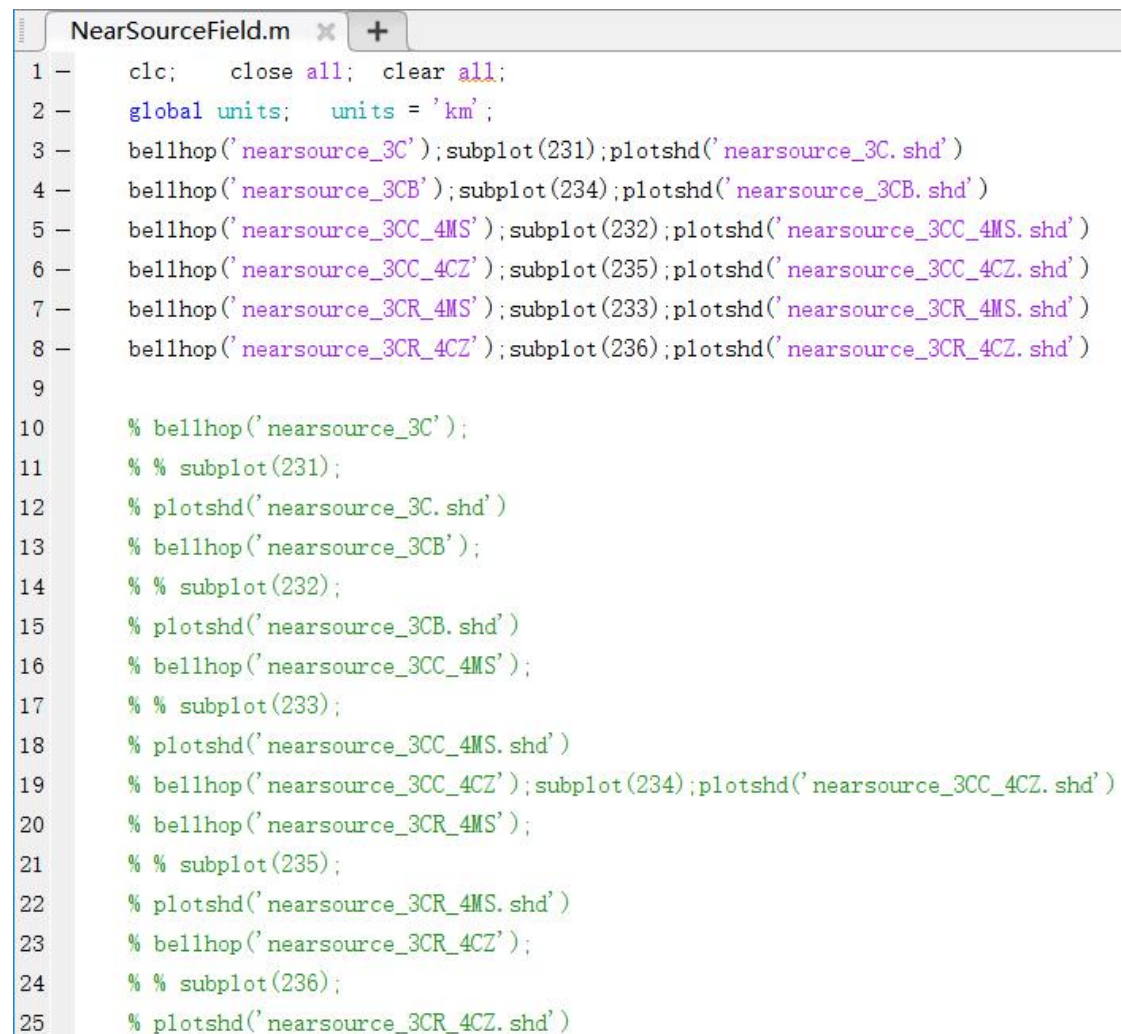
通过对\*.env 文件进行不同的波束选项设置，感触近场声场计算结果的差异。

Answer( 答案 ): Win10 平台上的 Matlab 命令脚本文件是 NearSourceField.m。

文件路径: ...\\GeneralDescription\\Task\_GD\_5\_nearsource

作业脚本文件名: command\_draft\_Task5\_NearSourceField.m

```
clc;    close all;  clear all;
global units;  units = 'km';
bellhop('nearsources_3C');subplot(231);plotshd('nearsources_3C.shd')
bellhop('nearsources_3CB');subplot(234);plotshd('nearsources_3CB.shd')
bellhop('nearsources_3CC_4MS');subplot(232);plotshd('nearsources_3CC_4MS.shd')
bellhop('nearsources_3CC_4CZ');subplot(235);plotshd('nearsources_3CC_4CZ.shd')
bellhop('nearsources_3CR_4MS');subplot(233);plotshd('nearsources_3CR_4MS.shd')
bellhop('nearsources_3CR_4CZ');subplot(236);plotshd('nearsources_3CR_4CZ.shd')
```



```
NearSourceField.m  x  +
1 -  clc;    close all;  clear all;
2 -  global units;  units = 'km';
3 -  bellhop('nearsources_3C');subplot(231);plotshd('nearsources_3C.shd')
4 -  bellhop('nearsources_3CB');subplot(234);plotshd('nearsources_3CB.shd')
5 -  bellhop('nearsources_3CC_4MS');subplot(232);plotshd('nearsources_3CC_4MS.shd')
6 -  bellhop('nearsources_3CC_4CZ');subplot(235);plotshd('nearsources_3CC_4CZ.shd')
7 -  bellhop('nearsources_3CR_4MS');subplot(233);plotshd('nearsources_3CR_4MS.shd')
8 -  bellhop('nearsources_3CR_4CZ');subplot(236);plotshd('nearsources_3CR_4CZ.shd')
9
10  % bellhop('nearsources_3C');
11  % % subplot(231);
12  % plotshd('nearsources_3C.shd')
13  % bellhop('nearsources_3CB');
14  % % subplot(232);
15  % plotshd('nearsources_3CB.shd')
16  % bellhop('nearsources_3CC_4MS');
17  % % subplot(233);
18  % plotshd('nearsources_3CC_4MS.shd')
19  % bellhop('nearsources_3CC_4CZ');subplot(234);plotshd('nearsources_3CC_4CZ.shd')
20  % bellhop('nearsources_3CR_4MS');
21  % % subplot(235);
22  % plotshd('nearsources_3CR_4MS.shd')
23  % bellhop('nearsources_3CR_4CZ');
24  % % subplot(236);
25  % plotshd('nearsources_3CR_4CZ.shd')
```

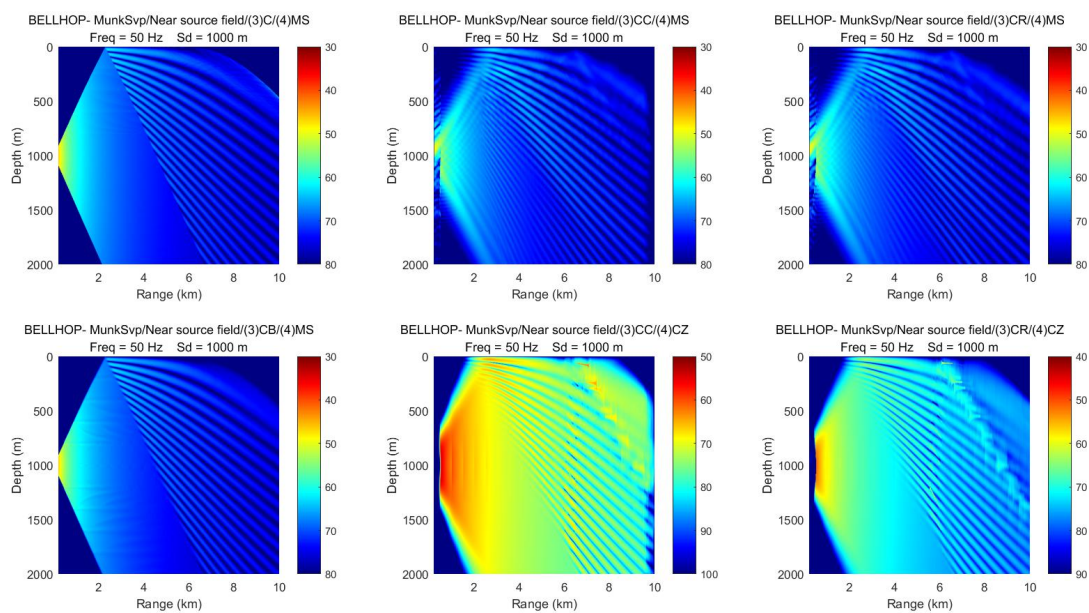


图 13: Bellhop 模型采用不同波束类型计算得到的声源附近的声场。

6.6 声压和声压分量

垂直传感器阵列（VSA）的最近发展引导在射线追踪模型中集成粒子速度的计算。因为声压分量正比于粒子速度分量，所以 Bellhop 通过计算声压分量来实现这种性能。为了展示这种性能，我们再将 nearsource.env 做如下修改：

nearsource_P.env					
行号	内 容	说 明			
1	'MunkSvp/Near field/acoustic pressure'	TITLE			
2	50.0	FREQ (Hz)			
3	1	NMEDIA			
4	'SVF'	SSP-TOP-WATER-OPT			
5	51 0.0 5000.0	NMESH SIGMA Z(NSSP)			
6	0.0 1548.52 /	Z()	CP()	CS()	RHO()
7	200.0 1530.29 /	AP()	AS()		
8	250.0 1526.69 /	.	.	.	.
9	400.0 1517.78 /	.	.	.	.
10	600.0 1509.49 /	.	.	.	.
11	800.0 1504.30 /	.	.	.	.
12	1000.0 1501.38 /	.	.	.	.
13	1200.0 1500.14 /	.	.	.	.
14	1400.0 1500.12 /	.	.	.	.
15	1600.0 1501.02 /	.	.	.	.
16	1800.0 1502.57 /	.	.	.	.
17	2000.0 1504.62 /	.	.	.	.
18	2200.0 1507.02 /	.	.	.	.
19	2400.0 1509.69 /	.	.	.	.
20	2600.0 1512.55 /	.	.	.	.
21	2800.0 1515.56 /	.	.	.	.
22	3000.0 1518.67 /	.	.	.	.
23	3200.0 1521.85 /	.	.	.	.
24	3400.0 1525.10 /	.	.	.	.
25	3600.0 1528.38 /	.	.	.	.
26	3800.0 1531.70 /	.	.	.	.
27	4000.0 1535.04 /	.	.	.	.
28	4200.0 1538.39 /	.	.	.	.
29	4400.0 1541.76 /	.	.	.	.

30	4600.0 1545.14 /	.	.	.	.	.	.
31	4800.0 1548.52 /	.	.	.	.	.	.
32	5000.0 1551.91 /	.	.	.	.	.	.
33	'A' 0.0	BOTOPT SIGMA					
34	5000.0 1600.00 0.0 1.8 .8 0.0	ZB	CPB	CSB	RHOB	APB	ASB
35	1	NSD					
36	1000.0 /	SD(1:NSD) (m)					
37	501	NRD					
38	0.0 2000.0 /	RD(1:NRD) (m)					
39	501	NRR					
40	0.0 30.0 /	RR(1:NRR ) (km)					
41	'CR'	OPTION: 'R/E/C/A/I/S'					
42	201	NBEAMS ISINGLE					
43	-25.0 25.0 /	ALPHA(1:NBEAMS) (°)					
44	0.0 5500.0 102.0	STEP (m)		ZBOX (m)		RBOX (km)	
45	'MS' 1.0 100.0 0	OPTIONS4		epmult		rloop isingl	
46	3 5 ''	nimage		ibwin		component	

再次运行 Bellhop, 我们得到图 14。（注意接收器深度数组为 0:2000）。

将最后一行修改为

3 5 'H' 和 3 5 'V'

就分别得到图 15 和图 16。

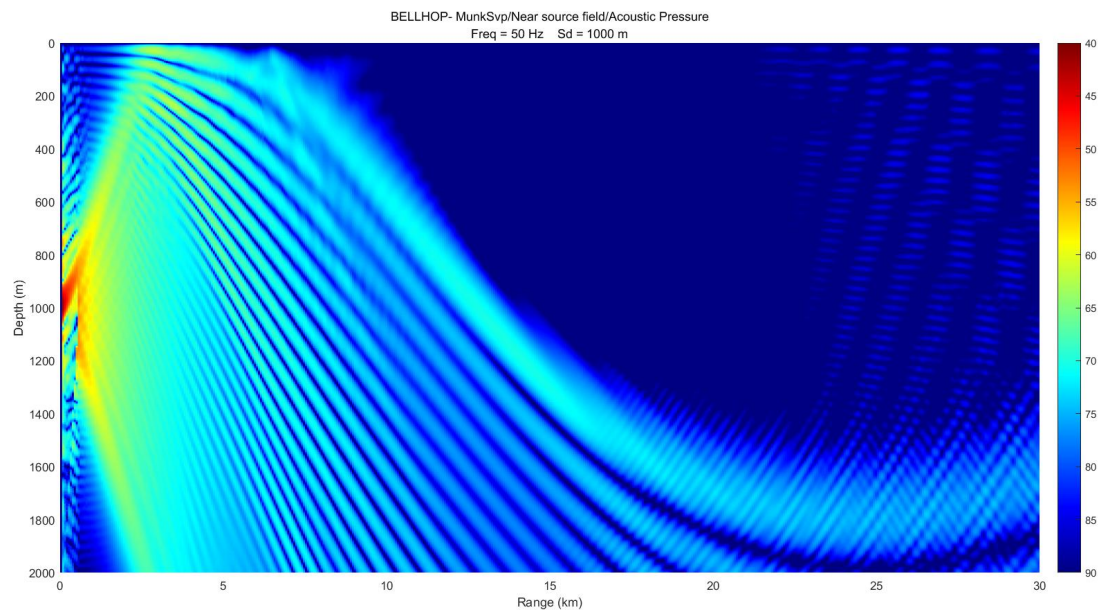


图 14：采用中心射线的坐标由声压计算得到的相干传播损失。



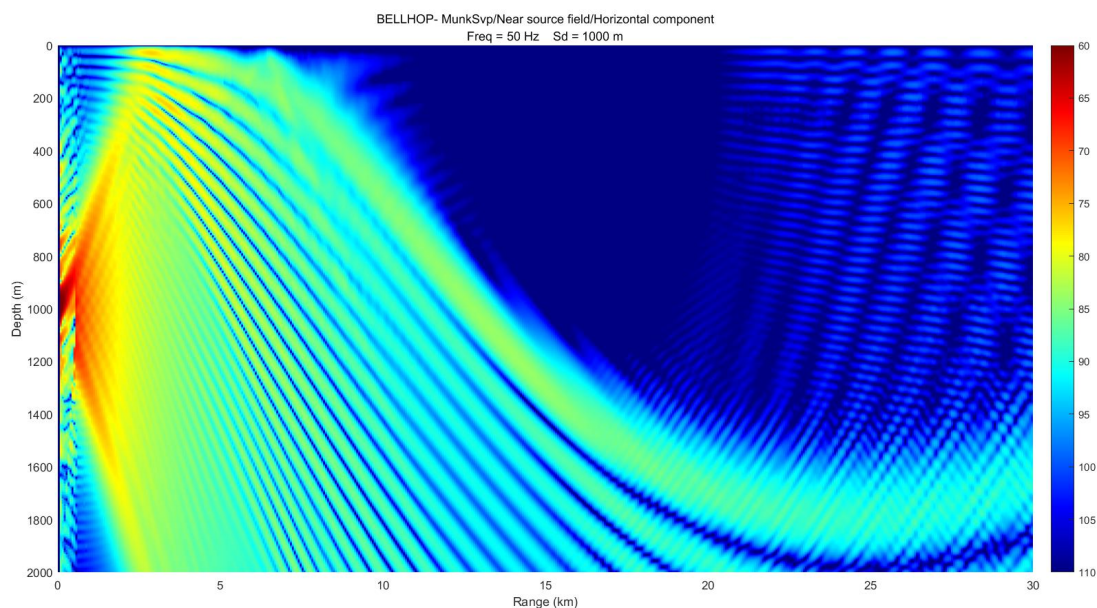


图 15：由声压的水平分量计算得到的相干传播损失。

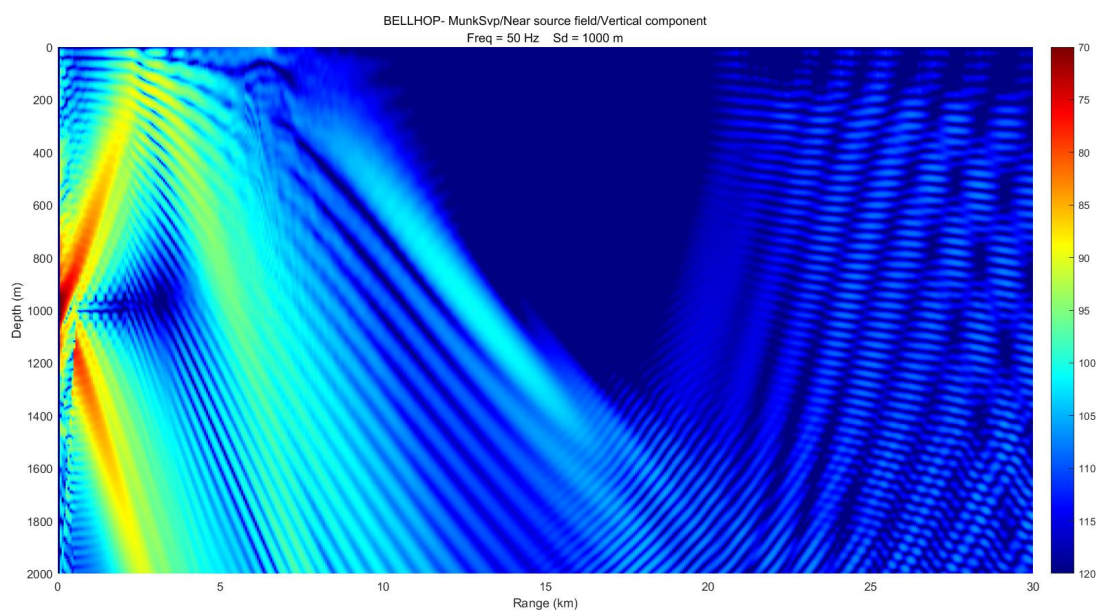


图 16：由声压的垂直分量计算得到的相干传播损失。

### 作业六：

通过改变\*.env 文件，计算由声压的水平与垂直分量导出的粒子速度场。初步感触矢量声场的表达方式。

Answer（答案）：

Win 平台上 Matlab 命令行脚本文件为 Pressure\_P\_H\_V.m，代码

文件路径: ...\\GeneralDescription\\Task\_GD\_6\_PHV.m

作业脚本文件名: command\_draft\_Task6\_Pressure\_P\_H\_V.m

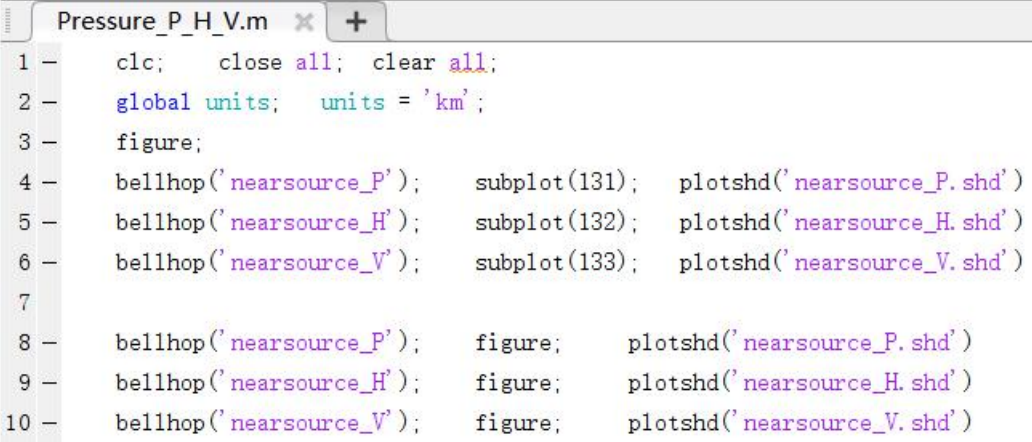
```
clc;    close all;  clear all;
global units;  units = 'km';
figure;
bellhop('nearsources_P');    subplot(131);    plotshd('nearsources_P.shd')
bellhop('nearsources_H');    subplot(132);    plotshd('nearsources_H.shd')
bellhop('nearsources_V');    subplot(133);    plotshd('nearsources_V.shd')

bellhop('nearsources_P');    figure;    plotshd('nearsources_P.shd')
bellhop('nearsources_H');    figure;    plotshd('nearsources_H.shd')
bellhop('nearsources_V');    figure;    plotshd('nearsources_V.shd')
```

如下。

需配合的文件:

nearsources\_P.env; nearsources\_H.env; nearsources\_V.env



```
Pressure_P_H_V.m  x +
1 -  clc;    close all;  clear all;
2 -  global units;  units = 'km';
3 -  figure;
4 -  bellhop('nearsources_P');    subplot(131);    plotshd('nearsources_P.shd')
5 -  bellhop('nearsources_H');    subplot(132);    plotshd('nearsources_H.shd')
6 -  bellhop('nearsources_V');    subplot(133);    plotshd('nearsources_V.shd')
7
8 -  bellhop('nearsources_P');    figure;    plotshd('nearsources_P.shd')
9 -  bellhop('nearsources_H');    figure;    plotshd('nearsources_H.shd')
10 - bellhop('nearsources_V');    figure;    plotshd('nearsources_V.shd')
```

## 7 结束语

撰写本概述的目的是希望能够帮助新手更容易地对 Bellhop 作些初级测试，也可当作一本新版模型手册，因为在实际中，现有文档相当缺乏。当然，随着模型发展，将来还需更新。因为在标准文献库中可提供的资料已经过时或者已不存在，本文档也就没从理论层面展现不同近似方法的具体细节。如果诸君对此问题有任何贡献（或展现 Bellhop 应用的其他有趣味的任何贡献），不才均将不胜感激。

在此赞叹好奇诸君能够习玩此卷并融入现实中浩瀚广袤的海洋！



## 参考文献

- [1] Porter M.B. and Bucker H.P. Gaussian beam tracing for computing ocean acoustic fields. *J. Acoust. Soc. America*, 82(4):1349{1359, 1987.
- [2] Jensen F., Kuperman W., Porter M., and Schmidt H. *Computational Ocean Acoustics*. AIP Series in Modern Acoustics and Signal Processing, New York, 1994.
- [3] Porter M. B. and Liu Y-C. Finite-Element Ray Tracing. In *Theoretical and Computational Acoustics*, Vol. 2, World Scientific Publishing Co., 1994.
- [4] Weinberg H. and Keenan R.E. Gaussian ray bundles for modeling high-frequency propagation loss under shallow-water conditions. *J. Acoust. Soc. America*, 100(3):1421-1431, September 1996.
- [5] Porter M. *The KRAKEN normal mode program*. SACLANT UNDER-SEA RESEARCH (memorandum), San Bartolomeo, Italy, 1991.