

# 最短路径法射线追踪的MATLAB实现

李志辉 刘争平 (西南交通大学土木工程学院 成都 610031)

**【摘要】** MATLAB 是一种广泛应用的优秀的科学计算语言和工具。本文简要阐述了最短路径法的基本原理, 探讨了在 MATLAB 环境中实现最短路径射线追踪的方法和步骤, 并通过数值模拟演示了所编程序在射线追踪正演计算中的应用。

**【关键词】** 最短路径法 射线追踪 MATLAB 数值模拟

利用地震初至波确定近地表介质结构, 在矿产资源的勘探开发及工程建设中有重要作用。地震射线追踪方法是研究地震波传播的有效工具, 目前常用的方法主要有有限差分解方程法和最小路径法。最短路径方法起源于网络理论, 首次由 Nakanishi 和 Yamaguchi 应用于地震射线追踪中。Moser 以及 Klimes 和 Kvasnicka 对最短路径方法进行了详细研究。通过科技人员的不断研究, 最短路径方法目前已发展较为成熟, 其基本算法的计算程序也较为固定。

被称作是第四代计算机语言的 MATLAB 语言, 利用其丰富的函数资源把编程人员从繁琐的程序代码中解放出来。MATLAB 语言用更直观的、符合人们思维习惯的代码, 为用户提供了直观、简洁的程序开发环境。本文介绍运用 Matlab 实现最短路径法的方法和步骤, 便于科研院校教学中讲授、演示和理解最短路径方法及其应用。

## 1 最短路径法射线追踪方法原理

最短路径法的基础是 Fermat 原理及图论中的最短路径理论。其基本思路是, 对实际介质进行离散化, 将这个介质剖分成一系列小单元, 在单元边界上设置若干节点, 并将彼此向量的节点相连构成一个网络。网络中, 速度场分布在离散的节点上。相邻节点之间的旅行时为他们之间欧氏距离与其平均慢度之积。将波阵面看成是由有限个离散点次级源组成, 对于某个次级源 (即某个网格节点), 选取与其所有相邻的点 (邻域点) 组成计算网格点; 由一个源点出发, 计算出从源点到计算网格点的透射走时、射线路径和射线长度; 然后把除震源之外的所有网格点相继当作次级源, 选取该节点相应的计算网格点, 计算出从次级源点到计算网格点的透射走时、射线路径和射线长度; 将每次计算出来走时加上从震源到次级源走时, 作为震源点到该网格节点的走时, 记录下相应射线路径位置及射线长度 (见图 1)。

根据 Fermat 原理逐步计算最小走时及射线方向。设  $\Omega$  为已知走时点  $q$  的集合,  $p$  为与其相邻的未知走时点,  $t_q$  分别和  $p$  点的最小走时,  $t_{qp}$  为  $q$  至  $p$  最小走时。  $r$  为  $p$  的次级源位置,

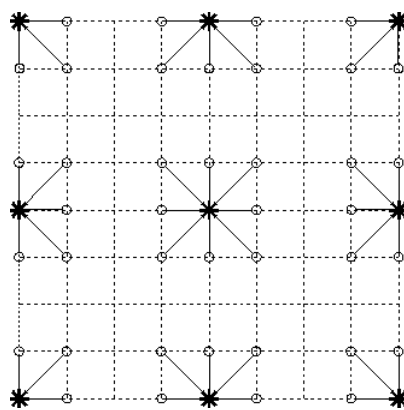


图 1 离散化模型 (星点表示震源或次级震源, 空心点为对应计算网格点)

则:

$$\{r = q : t_p = \min_{q \in \Omega} (t_q + t_{qp})\}$$

根据 Huygens 原理,  $q$  只需遍历  $Q$  的边界 (即波前点), 当所有波前邻点的最小走时都求出时, 这些点又成为新的波前点。应用网络理论中的最短路径算法, 可以同时求出从震源点传至所有节点之间的连线的近似地震射线路径。

## 2 最短路径法射线追踪基本算法步骤

把网格上的所有节点分成集合  $p$  和  $q$ ,  $p$  为已知最小旅行时的结点总数集合,  $q$  为未知最小旅行时的节点的集合。若节点总数为  $n$ , 经过  $n$  次迭代后可求出所有节点的最小旅行时。过程如下:

- 1) 初始时  $q$  集合包含所有节点, 除震源  $s$  的旅行时已知为  $t_s=0$  外, 其余所有节点的旅行时均为  $t_i$  ( $i$  属于  $Q$  但不等于  $s$ )。  $P$  集合为空集。
- 2) 在  $Q$  中找一个旅行时最小的节点  $i$ , 它的旅行时为  $t_i$ ;
- 3) 确定与节点  $i$  相连的所有节点的集合  $V$ ;
- 4) 求节点  $j$  ( $j$  属于  $V$  且  $j$  不属于  $P$ ) 与节点  $i$  连线的旅行时  $dt_{ij}$ ;
- 5) 求节点  $j$  的新旅行时  $t_j$  (取原有旅行时  $t_j$  与  $t_j+dt_{ij}$  的最小值);
- 6) 将  $i$  点从  $Q$  集合转到  $P$  集合;
- 7) 若  $P$  集合中的节点个数小于总节点数  $N$ , 转 2), 否则结束旅行时追踪;
- 8) 从接收点开始倒推出各道从源点到接收点的射线路径, 只要每个节点记下使它形成最小旅行时的前一个节点号, 就很容易倒推出射线路径。

Matlab 语言可以十分方便地构造用户自己的函数 (可以同其它目录里的函数同名), 供主函数调用, 并且通过 Matlab 结构形式, 根据属性名将节点上不同类型的数据组织起来, 从而简洁地实现算法中有关节点的判断、调用和运算。

## 3 数值模拟

### 3.1 构造数值模型

如图 2 所示, 数值模拟探测区域长  $Length=20m$ , 宽  $Width=8m$ 。假设已知背景介质速度为  $V$  低, 存在三个黑色高速介质区, 其中区域 1、2 相对于水平中线对称, 区域 3 相对水平居中,  $V$  高  $=4V$  低。可离散化为行间距  $0.5m$ 、列间距为  $1m$  的 17 行 21 列的网络。

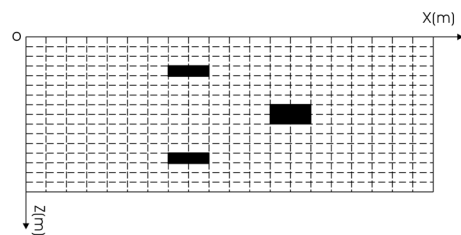


图 2 探测区域数值模型 (空白区为低速介质区, 黑色区为高速介质区)

### 3.2 私有函数构造

- (1) 构造函数  $Vnew\_jishuan()$ , 用以确定子波源点所在的结点相连的所有结点集合  $V$ 。  
 $[Vi, Vj]=Vnew\_jishuan(jiedian\_i, jiedian\_j, m, n)$

其中, 输出  $Vi$ ,  $Vj$  分别为与子波源点所在的结点相连的所有结点的行向量与列向量; 输

入中  $jiedian\_I$ ,  $jiedian\_j$  分别为子波源点所在的结点的网络行号与列号,  $m$ ,  $n$  为离散网络行列数。

(2) 构造函数  $wanzhengyan()$ , 用以实现从(单一)震源激发后, 追踪出已知速度场探测区域内离散其它节点旅行时及其射线路径。

$[DOTMN]=wanzhengyan(Length, Width, m, n, dotfa\_i, dotfa\_j, VDOTMN)$

其中, 输出  $DOTMN$  为 Matlab 结构构造出所在节点的各种属性 ( $velocity$ ,  $x$ ,  $z$ ,  $flag\_A$ ,  $flag\_Q$ ,  $before\_i$ ,  $before\_j$ ,  $time$ ,  $uptime$ ,  $lujing\_I$ ,  $lujing\_J$ ); 输入中  $Length, Width$  为探测区域长宽,  $dotfa\_i, dotfa\_j$  为震源点所在的结点的网络行号与列号,  $VDOTMN$  为数值模型速度场。

### 3.3 程序及相关说明

本节程序, 构成函数  $wanzhengyan()$ , 可以实现从(单一)震源激发后, 追踪出已知速度场探测区域内离散其它节点旅行时及其射线路径。它们分别为  $DOTMN(i,j).time$ 、 $DOTMN(dotjie\_i, dotjie\_j).lujing\_I$ 、 $DOTMN(dotjie\_i, dotjie\_j).lujing\_J$ 。

程序及其说明如下:

`clear,clc %清除工作空间及显示屏幕。`

`%第一步: 初始化实际介质离散化方式及速度场分布, 并设置震源点。`

`%输入以下输入探测区域 x 轴方向长度 Length, z 轴方向宽度 Width。`

`Length=20; Width=8;`

`m=17; n=21; %区域离散化 m 行 n 列。`

`%输入以下输入结点矩阵速度模型 VDOTMN:`

`VDOTMN=ones(m,n); VDOTMN([4:5],[8:10])=4;`

`VDOTMN([13:14],[8:10])=4; VDOTMN([8:10],[13:15])=4;%在此构造图 2 的速度场模型。`

`%定义 Q 为未作过子波源点的集合; P 作过子波源点的集合; A 为已经计算过走时的结点集合`

`Q=zeros(m,n); P=zeros(m,n); A=zeros(m,n); S=zeros(m,n);`

`%结点矩阵 DOTMN 初始化。`

`for i=1:m`

`for j=1:n`

`DOTMN(i,j).velocity=VDOTMN(i,j);`

`DOTMN(i,j).x=(j-1)*Length/(n-1);`

`DOTMN(i,j).z=(i-1)*Width/(m-1);`

`DOTMN(i,j).flag_A=0;`

`DOTMN(i,j).flag_Q=1;`

`DOTMN(i,j).before_i=NaN;`

`DOTMN(i,j).before_j=NaN;`

`DOTMN(i,j).time=inf;`

`DOTMN(i,j).uptime=NaN;`

```
end
end
%输入      以下输入震源点初值
dotfa_i=1; dotfa_j=3;
    DOTMN(dotfa_i,dotfa_j).flag_A=1;
    DOTMN(dotfa_i,dotfa_j).flag_Q=0;
    DOTMN(dotfa_i,dotfa_j).time=0;
jjedian_i=dotfa_i;
jjedian_j=dotfa_j;
for i=1:m
    for j=1:n
        if DOTMN(i,j).flag_Q == 0
            P(i,j)=1;
        end
    end
end
end
%计算集合 P 的结点个数。
counter_P=nnz(P); %第一步初始化结束。
%第二步:找 Q 中一个旅行时最小的结点。
%第七部分判断部分开始。
while counter_P < (m*n)
    %第三步: 确定与子波源点所在的结点相连的所有结点集合 V, 调用函数 Vnew_jishuan( )
    [V_i, V_j]=Vnew_jishuan(jjedian_i,jjedian_j ,m,n);
    for ii=1:length(V_i)
        i=V_i(ii);
        j=V_j(ii);
        DOTMN(i,j).flag_A=1;
    end
    ii=[];
    for i=1:m
        for j=1:n
            if DOTMN(i,j).flag_A == 1
                A(i,j)=1;
            end
        end
    end
end
%第四步: 计算  $S=(A-P)$  不为零结点到子波源的走时。
S=A-P;
[S_i,S_j]=find(S);
for ii=1:length(S_i)
    i=S_i(ii);
    j=S_j(ii);
    fanshu=(DOTMN(i,j).x-DOTMN(jjedian_i,jjedian_j).x).^2+(DOTMN(i,j).z-DOTMN(jjedian_i,jjedian_j).z).^2;
    DOTMN(i,j).uptime
```

```

=2*sqrt(fanshu)/(DOTMN(i,j).velocity+DOTMN(jiedian_i,jiedian_j).velocity );
%第五步：求 S 集合各结点的新旅行时
tmin_panju=min(DOTMN(i,j).time,DOTMN(jiedian_i,jiedian_j).time+DOTMN(i,j).uptime);
    if tmin_panju < DOTMN(i,j).time
        DOTMN(i,j).time=tmin_panju;
        DOTMN(i,j).before_i=jiedian_i;
        DOTMN(i,j).before_j=jiedian_j;
    end
end
%第二步：找 Q 中一个旅行时最小的结点。
time_min=inf;
for ii=1:length(S_i)
    i=S_i(ii);
    j=S_j(ii);
    DotTime(i,j)=DOTMN(i,j).time;
    if DotTime(i,j) <= time_min
        time_min=DotTime(i,j);
        jiedian_i=i;
        jiedian_j=j;
    end
end
%第六步：将子波结点移出集合 Q 并计算 P 集合结点的个数。
DOTMN(jiedian_i,jiedian_j).flag_Q=0;
for i=1:m
    for j=1:n
        if DOTMN(i,j).flag_Q == 0
            P(i,j)=1;
        end
    end
end
counter_P=nnz(P);
end
%第七部分判断结束结束。
%第八步：倒推出射线路径 DOTMN().lujing_I 与,DOTMN().lujing_J
%预备接收点循环。
for dotjie_i=1:m
    for dotjie_j=1:n
        lu_i=dotjie_i;
        lu_j=dotjie_j;
        Xi(1)=lu_i;
        Zj(1)=lu_j;
        counter_lu=2;
        lu_panju=isnan( DOTMN(lu_i,lu_j).before_i );
        while lu_panju< 1
            Xi(counter_lu)=DOTMN(lu_i,lu_j).before_i;

```

```
Zj(counter_lu)=DOTMN(lu_i,lu_j).before_j;  
    lu_temp=lu_i;  
    lu_i=DOTMN(lu_i,lu_j).before_i;  
    lu_j=DOTMN(lu_temp,lu_j).before_j;  
    lu_temp=[];  
    lu_panju=isnan( DOTMN(lu_i,lu_j).before_i );  
    counter_lu=counter_lu+1;  
end  
DOTMN(dotjie_i,dotjie_j).lujing_I=Xi;  
DOTMN(dotjie_i,dotjie_j).lujing_J=Zj;  
Xi=[];Zj=[];  
end  
end  
%第八步结束。
```

单一激发点最短路径射线追踪出来后,在以上程序段中加上激发点循环,可得到各激发点射线追踪路径和走时,这样就完成对探测区域数值正演计算。

在图 2 所示模拟区域上,标号 1 的列上的节点依次作为震源,在 21 列上各节点放置检波器,程序运行后,画出接收节点(检波器放置点)到各激发点射线路径叠加(见图 3)。

从图 3 中所示的的射线路径可见,射线集中通过高速介质区域,射线透射走时最短。射线路径的对称性与介质高速区域对称性完全一致,印证了程序运行的正确。

#### 4 小结

本文侧重于通过数值模拟介绍运用 MATLAB 实现最短路径射线追踪的简要方法和基本步骤,文中数值模型和其计算节点设置都仅考虑十分简单的情形。在实际应用中,除计算节点设置增多外,人们不断通过引入一系列其它方法进一步修正射线路径,来提高计算效率,减少计算误差。以上最短路径射线追踪程序在 MATLAB6.5 运行通过,供大家交流。

(收稿日期:2004-09-19; Email: lizhihui1689@163.com)

#### 参考文献

- [1] 苏晓生.掌握 MATLAB6.0 及其工程应用.北京:科学出版社,2002 年 1 月
- [2] Moser T J.Shortest path calculation of seismic[J].Geophysics,1991, 56(1):59-67
- [3] 张建中,陈世军,余大祥.最短路径射线追踪方法及其改进.地球物理学进展 Vol.18, No.1.2003.3
- [4] 林伯香,孙晶梅,刘清林.层析成像低速带速度反演和静校正方法.石油物探 Vol.41, No.2.2002.6

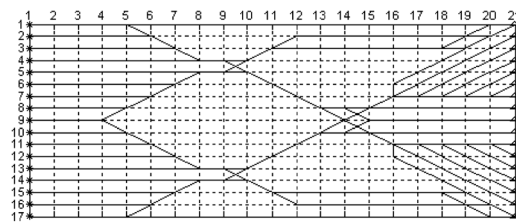


图 3 多震源穿透射线路径  
(\*点为震源, Δ 检波器, — 为初至波射线路径)