

数据的输入与输出

SMART 支持的数据格式有纯二进制文件、SU 格式数据文件和 SEGY 格式数据文件和部分的探地雷达 LTE 和 RD 数据文件。下面是 I/O 函数的汇总表。

表 1.1 输入函数列表

函数名称	简单描述
<code>data=gfreadat(fileName, nz, nx, [startAt, endAt])</code>	读入无道头的二进制数据体
<code>data=gfreadsu(fileName, [startAt, endAt])</code>	读入无卷头，有道头的 SU 格式的数据
<code>data=gfreadsegy(fileName, [startAt, endAt])</code>	读入有卷头，有道头的 SEGY 格式的数据
<code>hdr=gfreadhdr(fileName, [startAt, endAt, 'su'])</code>	读入 SU 和 SEGY 文件道头信息
<code>[data, hdr]= gfreadsegyhdr(fileName, [startAt, endAt])</code>	读入 SU 和 SEGY 文件数据及道头信息
<code>data=gfin(fileName, nz, nx, 'dat')</code>	读入无道头的二进制数据体
<code>data=gfin(fileName, 'su', [startAt, endAt])</code>	读入无卷头，有道头的 SU 格式的数据
<code>data=gfin(fileName, 'segy', [startAt, endAt])</code>	读入有卷头，有道头的 SEGY 格式的数据
<code>data=gfin(fileName, [startAt, endAt])</code>	读入有卷头，有道头的 SEGY 格式的数据
<code>data=gfin(fileName, 'hdr', [startAt, endAt])</code>	读入 SEGY 文件道头信息
<code>[data, hdr, hdrself]=gfreadfile(fileName)</code>	自动判断文件格式，可以读取SU、SEGY、LTE（中电22所探地雷达数据格式）、RAD（RD3、RD7，mala探地雷达数据格式）

表 1.2 输出函数列表

函数名称	简单描述
<code>gfwritedat(data, [fileName, 'a'])</code>	将数据写入纯二进制格式文件，
<code>gfwritesu(data, [fileName, dt, 'a'])</code>	将数据写入 SU 格式文件，
<code>gfwriteseggy(data, [fileName, dt])</code>	将数据写入 SEGY 格式文件
<code>gfwriteseggyhdr (data, tracehdr,[fileName,'ibm'])</code>	将数据写入 SEGY 格式文件
<code>gfwriteseggy2(data,[fileName,dt,'ibm'])</code>	将数据添加到 SEGY 格式文件
<code>gfwriteseggyhdr 2(data, tracehdr,[fileName,'ibm'])</code>	将数据添加到 SEGY 格式文件
<code>gfout(data,[fileName,dt,'ibm', 'a'])</code>	将数据写入\添加到 SEGY 格式文件
<code>gfout(data, tracehdr, 'hdr',[fileName,'ibm', 'a'])</code>	将数据写入\添加到 SEGY 格式文件

1. 二进制数据

二进制数据文件没有任何的道头信息，一般以 .dat 或者 .bin 为后缀名，有关二进制文件读写的函数有：

名称: `gfreadat`

功能: 读入纯二进制文件

用法: `data=gfreadat(fileName, nz, nx, [startAt, endAt])`

输入: 文件名称 `fileName`；数据体的大小 `nz` 和 `nx`，`nz` 表示沿纵向的数据点数，`nx` 表示沿横向的数据点数；可选参数 `startAt` 表示起始点（默认是从第一道开始），`endAt` 表示结束点（默认是 `nx`）

输出: 单精度 `nz-by-(endAt-startAt+1)` 二维数据体（三维数据体利用 `reshape` 转换）

名称: `gfin`

功能: 多种数据格式输入函数

用法: `data=gfin(fileName, nz, nx, 'dat')`

输入: 文件名称 `fileName`; 数据体的大小 `nz` 和 `nx`, 标识符 `'dat'` 说明文件格式 (不可省略, 为输入方便该标识符可位于除第一位置之外的其他地方)

输出: 单精度 `nz-by-nx` 二维数据体 (三维数据体利用 `reshape` 转换)

说明: `gfin` 封装了多个输入函数, `gfin` 在读入纯二进制文件式不支持任意起点读入。

名称: `gfwritedat`

功能: 输出数据为二进制格式

用法: `gfwritedat(data, [fileName, 'a'])`

输入: 需要输出的数据体 `data`; 输出文件名称 `fileName`, 文件名称默认为 `data` 变量的名称, 比如 `gfwritedat(testData)`, 则输出文件为 `testData.dat`; 标识符 `'a'` 表示把数据添加在 `fileName` 文件, 若 `fileName` 文件不存在则自动创建。

输出: 无

2. SU 数据

SU 数据格式一般是无卷头有道头的小头的 (`little-endian`) IEEE 浮点型数据格式, SMART 中关于 SU 数据体读写的函数有:

名称: `gfreedsu`

功能: SU 数据格式输入函数

用法: `data=gfreedsu(fileName, [startAt, endAt])`

输入: 文件名 `fileName`; 起始位置 `startAt` (可选参数, 默认为 1), 结束位置 `endAt` (可选参数, 默认为最大道数)

输出: 单精度 `nz-by-(endAt-startAt+1)` 二维数据体 (三维数据体利用 `reshape` 转换)

名称: `gfin`

功能: 多种数据格式输入函数

用法: `data=gfin(fileName, 'su', [startAt, endAt])`

输入: 文件名 `fileName`; 起始位置 `startAt` (可选参数, 默认为 1), 结束位置 `endAt` (可选参数, 默认为最大道数); 字符串 `'su'` 说明文件格式 (不可省略, 可位于除第一位置之外的其他位置)

输出: 单精度 `nz-by-(endAt-startAt+1)` 二维数据体 (三维数据体利用 `reshape` 转换)

名称: `gfreahdr`

功能: 读取 SU 和 SEG Y 格式的道头文件

用法: `hdr=gfreahdr(fileName, 'su', [startAt, endAt])`

输入: 文件名 `fileName`; 起始位置 `startAt` (可选参数, 默认为 1), 结束位置 `endAt` (可选参数, 默认为最大道数); 标识符 `'su'` 说明从 SU 文件中读取道头 (不可省略, 可位于除第一位置之外的其他位置), 省略掉该标识符则表示从 SEG Y 文件中读取道头

输出: `endAt-startAt+1` 道头结构体 `hdr`, (具体见下一节)

名称: `gfwritesu`

功能: 输出数据为 SU 格式

用法: `gfwritesu(data, [fileName, dt, 'a'])`

输入: 需要输出的数据体 `data`; 输出文件名称 `fileName`, 文件名称默认为 `data` 变量的名称, 比如 `gfwritesu(testData)`, 则输出文件为 `testData.su`; 采样间隔为 `dt` (默认为 `dt=0.001`); 标识符 '`a`' 表示把数据添加在 `fileName` 文件, 若 `fileName` 文件不存在则自动创建, 输出文件的道头信息只有采样点数及采样间隔

输出: 无

说明: `gfwritesu` 没有提供相关道头信息, 建议正规数据采用 `SEG-Y` 格式。对于文件名称 `fileName`, 笔者建议给出。(因为笔者经常需要输出某个变量来检查程序是否正确, 出于方便考虑, 才将 `fileName` 设置成具有默认值的参数)。

3. SEG-Y 数据

标准 SEG-Y 文件一般包括三部分, 第一部分是 EBCDIC 文件头 (3200 字节), 由 40 个卡组成 (例如: 每行 80 个字符*40 行), 用来保存一些对地震数据体进行描述的信息; 第二部分是二进制卷头文件 (400 字节) 用来存储描述 SEG-Y 文件的一些关键信息, 包括 SEG-Y 文件的数据格式、采样点数、采样间隔、测量单位等一些信息, 这些信息一般存储在二进制文件头的固定位置上 (其中最重要的是数据格式); 第三部分是实际的地震道, 每条地震道都包含 240 字节的道头信息和地震道数据。道头数据中一般保存该地震道对应的线号、道号、采样点数、大地坐标等信息, 但一些关键的参数位置 (如线号、道号在道头中的位置) 并不固定。地震道数据是对地震信号的波形按一定时间间隔进行取样, 再把这一系列的离散振幅值以某种方式记录下来。地震数据格式可以是 IBM 浮点型、IEEE 浮点型、整型、长整型等, 一个三维地震工区同一次处理的地震数据格式是唯一的。地震道采样点数由该地震道道头中采样点数决定, 大部分 SEG-Y 文件的所有地震道采样点数是一致的, 但也存在不同地震道采样点数不同的情况, 一般称这种 SEG-Y 文件为变道长格式的 SEG-Y 文件 (SMART 不支持变道长格式的 SEG-Y 格式)。详细 SEG-Y 文件格式参见附录 1。

3.1 道头文件

除了数据格式之外 (用以说明地震数据格式的类型, 比如 IBM 浮点型、IEEE 浮点型、整型、长整型等), 卷头文件中绝大部分信息在 240 字节的道头信息中都有所包含, 因此 SMART 没有提供专门的函数读取卷头信息。SMART 中道头数据中保存的线号、道号、采样点数、大地坐标等信息被存储在一个道头结构体当中, `hdr` 可查看个成员变量名称及其含义。与道头相关的函数有:

名称: `gfreahdr`

功能: 读取 SU 和 SEG-Y 格式的道头文件

用法: `hdr=gfreahdr(fileName, [startAt, endAt])`

输入：文件名 `fileName`；起始位置 `startAt`（可选参数，默认为 1），结束位置 `endAt`（可选参数，默认为最大道数）

输出：`endAt-startAt+1` 道头结构体 `hdr`

说明：道头文件读写比较耗时。由于笔者水平有限，不知如何像 C 语言那样直接将二进制数据载入到一个具有指定内存大小的结构体中。SMART 采用的是先将 240 字节数据读入，再逐个赋值给道头结构体的方法，因此读写比较耗时。

名称：`gfin`

功能：多种数据格式输入函数

用法：`data=gfin(fileName, 'hdr', [startAt, endAt])`

输入：`fileName` 文件为 SEGY 格式；起始位置 `startAt`（可选参数，默认为 1），结束位置 `endAt`（可选参数，默认为最大道数）；字符串 '`hdr`' 说明读取道头文件（不可省略，可位于除第一位置之外的其他位置）

输出：数组大小为 `endAt-startAt+1` 的道头结构体 `hdr`

名称：`gfhdrcreate`

功能：生成道头文件

用法：`[hdr, hdrself]=gfhdrcreate(ntr)`

输入：`ntr` 地震道数

输出：结构体数组 `hdr`，结构体成员变量默认值为 0；`hdrself` 说明每个成员变量的名称

名称：`gfhdrstruct`

功能：查找与某一关键词相关的道头信息

用法：`okfind=gfhdrstruct(keyword, [verb])`

输入：关系词 `keyword`；`verb=1` 显示全部信息

输出：`okfind=1` 为查询到与关键词相关的道头信息并显示；`okfind=0` 表示为查询到该信息

例如：键入 `gfhdrstruct('cdp',1)`，会得到与 '`cdp`' 相关的道头信息，包括描述该信息的关键词缩写、在道头内的位置等等。这个函数是为了在制作道头文件时方便查询各关键词。

3.2 SEGY 数据输入

MATLAB 平台下读取 SEGY 数据比较麻烦的地方在于 IEEE 和 IBM 浮点型数据格式的转换，笔者起初利用 CREWES 内的 MATLAB 转换函数进行转换，但是其效率非常之低，后转换函数采用 C 语言进行实现，转换效率得到大幅提高。由于 CMEX 函数与平台有关，目前仅支持 64 位和 32 位的 Windows 和 Linux 平台，其他平台未测试。如果这些转换函数无法使用，需重新编译（编译方法见附录 2）。与 SEGY 数据输入相关的函数有：

名称：`gfreadsegy`

功能：SEGY 数据格式输入函数

用法：`data=gfreadsegy(fileName, [startAt, endAt])`

输入：文件名 `fileName`；起始位置 `startAt`（可选参数，默认为 1），结束位置 `endAt`（可选参数，默认为最大道数）

输出：单精度 `ns-by-(endAt-startAt+1)` 二维数据体（三维数据体利用 `reshape` 转换）

名称: `gfreasegyhdr`

功能: SEGY 数据格式输入函数

用法: `[data, hdr] = gfreasegyhdr(fileName, [startAt, endAt])`

输入: 文件名 `fileName`; 起始位置 `startAt` (可选参数, 默认为 1), 结束位置 `endAt` (可选参数, 默认为最大道数)

输出: 单精度 `ns-by-(endAt-startAt+1)` 二维数据体 (三维数据体利用 `reshape` 转换) 和数组大小为 `endAt-startAt+1` 的道头结构体 `hdr`

名称: `gfin`

功能: 多种数据格式输入函数

用法: `data = gfin(fileName, 'segy', [startAt, endAt])`

输入: 文件名 `fileName`; 起始位置 `startAt` (可选参数, 默认为 1), 结束位置 `endAt` (可选参数, 默认为最大道数), 标识符 'segy' 可省略

输出: 单精度 `nz-by-(endAt-startAt+1)` 二维数据体 (三维数据体利用 `reshape` 转换)

3.3 SEGY 数据输出

SEGY 格式文件是 SMART 默认的输出格式, 输出功能较全, 与 SEGY 数据输出有关的

函数有:

名称: `gfwritesegy`

功能: 输出数据为 SEGY 格式

用法: `gfwritesegy(data, [fileName, dt, 'ibm'])`

输入: 需要输出的数据体 `data`; `fileName` 文件名默认为 `data` 变量的名称, 比如 `gfwritesegy(testData)`, 则输出文件为 `testData.segy`; 采样间隔为 `dt` (默认为 `dt=0.001`); 'ibm' 标识符表示数据格式为 IBM 浮点型, 无该标识符则输出为 IEEE 浮点型; 若 `fileName` 文件不存在则自动创建, 输出文件的道头信息只有采样点数及采样间隔

输出: 无

说明: 将数据体添加的已有文件的函数为 `gfwritesegy2`

名称: `gfwritesegyhdr`

功能: 输出数据为 SEGY 格式

用法: `gfwritesegyhdr(data, tracehdr, [fileName, 'ibm'])`

输入: 需要输出的数据体 `data`; 道头结构体 `tracehdr`; `fileName` 文件名默认为 `data` 变量的名称, 'ibm' 标识符表示把数据格式为 IBM 浮点型; 若 `fileName` 文件不存在则自动创建, 输出文件的道头信息只有采样点数及采样间隔

输出: 无

说明: 由于卷头文件中除了数据格式需要设置外, 其他信息都在道头信息内, 因此没有额外提供相对应的函数; 将数据体添加的已有文件的函数为 `gfwritesegyhdr2`

名称: `gfout`

功能: 输出数据为 SEGY 格式

用法: `gfout(data, [fileName, dt, 'ibm', 'a'])`

输入: 需要输出的数据体 `data`; `fileName` 文件名默认为 `data` 变量的名称; 采样间隔为 `dt` (默认为 `dt=0.001`); 'ibm' 标识符表示数据格式为 IBM 浮点型; 标识符 'a' 表示把数据添加在 `fileName` 文件, 若 `fileName` 文件不存在则自动创建, 输出文件的道头信息只有采样点数及采样间隔

输出：无

名称：gfout

功能：输出数据为 SEG-Y 格式

用法：gfout(data,tracehdr, 'hdr',[fileName,'ibm', 'a']);

输入：需要输出的数据体 data；道头结构体 tracehdr；标识符'hdr'；fileName 文件名默认为 data 变量的名称；'ibm'标识符表示数据格式为 IBM 浮点型；标识符 'a' 表示把数据添加在 fileName 文件,若 fileName 文件不存在则自动创建，输出文件的道头信息只有采样点数及采样间隔

输出：无

4. I/O 函数示例

下面是一个数据输出的例子：

程序 1 数据读写

```

1  ntr=400;
2  data=peaks(ntr);           % 利用 peaks 函数生成一个 400-by-400 的测试数据
3  traceHeader=gfhdrcreate(ntr); % 生成空道头结构体
4  for ix=1:400               % 几个重要的参数
5      traceHeader(ix).fldr=1; % 野外记录号
6      traceHeader(ix).cdp=ix; % CDP 号
7      traceHeader(ix).offset=(ix-1)*20; % 偏移距，假设炮点在 (0,0)，道间距 20m
8      traceHeader(ix).sx=0; % 炮点坐标，假设炮点在 (0,0)
9      traceHeader(ix).sy=0; % 炮点坐标，假设炮点在 (0,0)
10     traceHeader(ix).gx=(ix-1)*20; % 检波器坐标，假设侧线沿 X 轴方向
11     traceHeader(ix).sy=0; % 检波器坐标，假设侧线沿 X 轴方向，故 Y 为 0
12     traceHeader(ix).ns=ntr; % 采样点数，必填参数
13     traceHeader(ix).dt=1e-3*1e6; % 采样间隔，微秒
14 end
15 gfwritesu(data,'peak.su'); % 输出为 SU 格式
16 gfwritesegy(data,'peak.segy','ibm'); % 输出为 IBM SEG-Y 格式
17 gfwritesegyhdr(data,traceHeader,'peak2.segy'); % 输出为 IEEE SEG-Y 格式，无'ibm'标识符即为 32-bit
18 % IEEE 浮点型数据

```

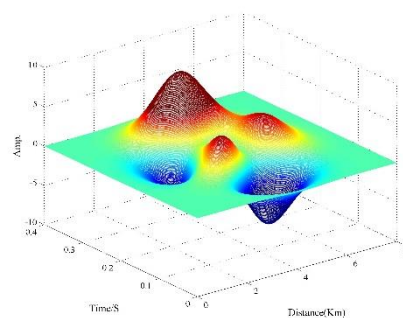
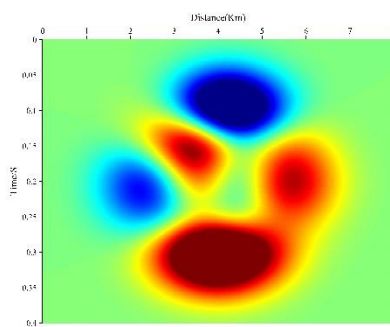


图 1 测试数据：左二维平面图，右三维网格图

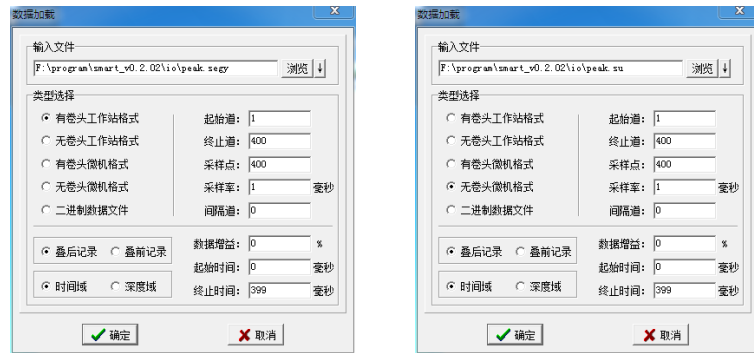


图 2 Fimage 进行显示，SU 格式被称作无卷头微机格式，SEGY 格式被称作有卷头工作站格式