SegyMAT

Thomas Mejer Hansen

Niels Bohr Institute for Astronomy, Physics and Geophysics

University of Copenhagen, Denmark

SegyMAT
by Thomas Mejer Hansen

1.03 Edition
Published Wed Dec 15 11:29:05 CET 2004

A Matlab program for reading and writing SEG Y files.

Table of Contents

List of Tables

About Segymat

Matlab is a very helpful tool to analyze seismic data. The most common standard for reading and writing seismic data is the SEG Y standard. SegyMAT is a set m-files that aims to ease importing and exporting SEG-Y files from Matlab.

SegyMAT aims to completely support the SEG-Y revision 1.

SegyMAT aims to be easy to use in other projects.

SegyMAT aims to be compileable using the Matlab Compiler.

SegyMAT is not fast. SegyMAT makes heavy use of ’structures’. Unfortunately structures are not very effective in terms of speed in Matlab. (Or they have not been implemented very effectively in SegyMAT). However structures make the implementation easier, and more robust.

The latest version of SegyMAT is always available from Sourceforge : http://segymat.sourceforge.net/.

Mail any questions to tmh@gfy.ku.dk

Chapter 1. Install SegyMAT

1.1. Requirements

SegyMAT has been developed and tested using Matlab 6.5 (R13.5) and 7.0 (R14) running on Linux and Windows XP. Any other Matlab supported platform should work.

As of version 1.02 Octave (version >2.1.64) is supported as well.

No Matlab toolboxes are required.

1.2. Local Installation

If you run Matlab with Java extensions (the default), you can run the pathtool, select the SegyMAT install directory, and save the path and you are done ;

```
>> pathtool
```

To install SegyMAT without the Java gui (using ’`matlab -nojvm`’) you can manually add the SegyMAT folder to Matlabs search path. If you unpack SegyMAT to /usr/share/matlab/SegyMAT simply do :

```
>> addpath /usr/share/matlab/SegyMAT -begin
```

1.3. Global Installation

For a system wide installation add the following line (substituting the location of the SegyMAT directory)

```
>> addpath /usr/share/matlab/SegyMAT -begin
```

to `pathdef.m`, usually located in `$MATLAB_INSTALL/toolbox/local/pathdef.m`

Chapter 2. The SEG-Y Format

SegyMAT has been implemented using the *SEG-Y standard* (http://seg.org/publications/tech-stand/) as defined by SEG [1]

SegyMAT also has support for reading and writing the format used by CWP's *Seismic Unix* (http://www.cwp.mines.edu/cwpcodes/) package (the SU format), which is merely a simplified version the SEG-Y format.

A short description of the formats follows here

2.1. Structure of a SEG-Y file

A SEG-Y file consists of a 3600 byte header; a number of extended textual headers; a number trace headers+data.

- A 3200 byte Textual File Header, ASCII or EBCDIC formated.
- A 400 byte Binary File Header
- A (optional) number of 'Extended Textual File Headers', 3200 bytes long, ASCII or EBCDIC formatted.
- A number of traces, separated into a 240 bytes long binary Trace Header, followed by the Trace Data, that can be formatted in a number of ways : IEEE, IBM Floating Point, 1,2 and 4 byte two's complement integers.

2.2. Structure of a SU file

A SU formatted file is just a simple version of a SEG-Y file, containing only trace information :

- No 3200 byte textual header and no extended textual headers.
- No binary SEG-Y header.
- The data must be formatted as IEEE.
- Data can be both little and big endian formatted.

2.3. What is supported in SegyMAT ?

As of version 1.03 SegyMAT supports the following parts of the SEG-Y revision 0 and 1.

2.3.1. Textual file headers

The Textual 400 byte file header can be both ASCII and EBCDIC formatted, using SEG-Y revision 1. Both ASCII and EBCDIC are supported since version 0.39.

2.3.2. Extended Textual Headers

In SEG-Y revision 1 a number of extended textual file headers are allowed.

As of version 0.9 extended textual headers are supported by SegyMAT.

2.3.3. Data Sample Format / Revision

The following data formats are supported :

Table 2-1. Supported SEG-Y revision 0 (1975)

Type	DataSampleFormat	Supported
1	4 Byte IBM Floating Point	Yes
2	4 Byte Fixed Point	No
3	2 Byte Fixed Point	No
4	4 Byte Fixed Point with Gain	No

Table 2-2. Supported SEG-Y revision 1 (2002)

Type	DataSampleFormat	Supported
1	4 Byte IBM Floating Point	Yes
2	4 Byte two’s complement integer	Yes
3	2 Byte two’s complement integer	Yes
4	4 Byte Fixed Point with Gain	No
5	4 Byte IEEE FLoating Pint	Yes
6	Not Specified	
7	Not Specified	
8	1 Byte Fixed Point with Gain	Yes

The type number is the number that should be used as `'dsf'` (Data Sample Format), for functions like ReadSegy.m, WriteSegy.m, WriteSegyStructure.m.

Notes

1. The Society of Exploration Geophysicists

Chapter 3. Reading SEG-Y files

3.1. ReadSegy

`ReadSegy.m` can be used to read SEG-Y files :

```
>> [Data,SegyTraceHeaders,SegyHeader]=ReadSegy('data.segy');
>> wiggle(Data,[],SegyHeader.time,[SegyTraceHeaders.cdp],'VA')
>> imagesc([SegyTraceHeaders.cdp],[SegyHeader.time],Data)
```

This will read `data.segy` using the SEG-Y revision and data sample format specified in the binary SEG-Y header, and plot the data.

`Data` is a 2D variable containing the sesimic data. [Nsamples*Ntraces];

`SegyTraceHeaders` is a structure of size [1,Ntraces] structure containing all the header values from the SEG-Y traces. Type `SegyTraceHeaders` to see a list of header information. SegyTraceHeaders(9), list all header names and values of trace number 9.

To access an array of trace header values simply us square brackets as :

```
cdp=[SegyTraceHeaders.cdp];
offset=[SegyTraceHeaders.offset];
...
```

`SegyHeader` is a structure containing all the SEG-Y header values. Typing '>> SegyHeader' will list the names and values of all SEG-Y header values.

A number of arguments can be given to `ReadSegy`, controlling which part of the data set to read, and the format.

3.1.1. To read time slice 0.5 $<$ t $<$ 5

```
>> [Data,SegyTraceHeaders,SegyHeader]=
      ReadSegy(filename,'trange',.5,3);
```

3.1.2. Read only every 5th trace

```
>> [Data,SegyTraceHeaders,SegyHeader]=
      ReadSegy(filename,'jump',5);
```

3.1.3. Read data in a CDP header range : 5000<cdp<5800

cdp can be changed to any other valid header name

```
>> [Data,SegyTraceHeaders,SegyHeader]=
      ReadSegy(filename,'minmax','cdp'5000,5800);
```

3.1.4. Read only header values

In some cases it can be desirable only to read the SEG-Y header files (file header and traceheaders). This will return an empty Data variable.

```
>> [Data,SegyTraceHeaders,SegyHeader]=
      ReadSegy(filename,'SkipData',1);
```

3.1.5. SEG-Y format revision

SEG-Y format revision number can be '0' (1975) or '1' (2002). By default the SEG-Y format revision number is read in the binary header, but this can be overruled using :

```
>> [Data,SegyTraceHeaders,SegyHeader]=
      ReadSegy(filename,'revision',0);
```

3.1.6. A specific Data Sample Format

One can overrule the Data Sample Format listed in the binary SEG-Y header, using the dsf argument. See Section 2.3.3 for a list of valid and supported values.

```
>> % Rev 0, IBM FLOATING POINT
>>  [Data,SegyTraceHeaders,SegyHeader]=
       ReadSegy(filename,'revision',0,'dsf',1);
>> % Rev 1, IEEE FLOATING POINT
>>  [Data,SegyTraceHeaders,SegyHeader]=
       ReadSegy(filename,'revision',1,'dsf',5);
```

If dsf is set to 5 and revision to 0, a warning message will occur, since data sample format 5 is only defined in revision 1. The revision is then automatically set to 1.

3.1.7. Force the use of a specific SegyHeader

```
>> [Data,SegyTraceHeaders,SegyHeader]=
      ReadSegy(filename,'SegyHeader',SegyHeader);
```

3.2. ReadSegyFast

`ReadSegyFast.m` is a faster implementation of ReadSegy.m since no trace header values are read. Thus this function will just return the seismic data and the SEG-Y-header. e.g. :

```
>> [Data,SegyHeader]=ReadSegy('data.segy');
>> imagesc(Data)
```

This will read `data.segy` using the SEG-Y revision and data sample format specified in the binary SEG-Y header, and plot the data.

3.2.1. ReadSegyFast options

As for ReadSegy.m the SEG-Y-revision a number of option can be used. ReadSegyFast.m uses the same convention as ReadSegy.m.

The data sample format can be chosen using the 'revision' and 'dsf' tags. Also a 'SegyHeader' can be specified.

ReadSegyFast.m is currently optimized only for reading the whole SEG-Y-file, but the options 'jump' and 'trange' can be used (but will currently not result in faster read times).

Since the trace header values are not read, the 'minmax' option is not supported.

3.3. ReadSegyHeader

`ReadSegyHeader.m` reads the Binary Segy Header only. It can be called with the same options as ReadSegy.m

3.3.1. Force using little endian :

```
>> SegyHeader=ReadSegyHeader(filename,'endian','l');
```

3.4. ReadSu

`ReadSu.m` works similar to ReadSegy.m and the same input parameters can be used. A `SuHeader` can optionally be returned, that simple is a mostly empty SEG-Y file header.

```
>> [Data,SuTraceHeaders,SuHeader]=ReadSu(filename);
```

Chapter 4. Writing a SEG Y file

WriteSegy.m is available to write a Matrix to disk as a SEG-Y file.

WriteSegyStructure.m is available to write A Segy File using a specified set of headers. This is of special use if one wants to load a seismic data set, work with the data in Matlab, and the write the data to disk using the same header values.

4.1. WriteSegy

`WriteSegy` can be used to save a matrix of data, in Matlab as a SEG-Y file.

4.1.1. Specify values for the SEGY Header

Here `dt` is a scalar and `Inline`, `Crossline`, `X` and `Y` are arrays of valies of size size(data,2)

```
>> WriteSegy('datacube.segy',data,
    'dt',.004,'Inline3D',Inline,'Crossline3D',Crossline,
    'cdpX',X,'cdpY',Y);
```

4.1.2. Specify SEG-Y revision

```
>> WriteSegy('test.segy',seisdata,'revision',0); % SEG-Y Revision 0
>> WriteSegy('test.segy',seisdata,'revision',1); % SEG-Y Revision 1
```

4.1.3. Specify data sample format

See Section 2.3.3 for a list of valid and supported values for the datasample format `dsf`.

```
>> % Force Revision 1 and IEEE Floating point :
>> WriteSegy('test.segy',seisdata,'dsf',5,'revision',1);
>>
>> % Force Revision 0 and IBM Floating point :
>> WriteSegy('test.segy',seisdata,'dsf',1,'revision',0);
```

4.2. WriteSegyStructure

`WriteSegyStructure` can be used to write a seismic data to disk gived that both `SegyHeader`, `SegyTraceheaders` and the data `Data` are known. They can be obtained using ReadSegy.m like ;

```
>> [Data,TraceHeaderInfo,SegyTraceHeaders,SegyHeader]=ReadSegy('data.segy');
```

To write the data using `WriteSegyStructure` simply do

```
>> WriteSegyStructure('datacube.segy',SegyHeader,SegyTraceHeaders,Data);
```

4.2.1. Force SEG-Y revision

```
>> % Revision 0
>> WriteSegyStructure('datacube.segy',SegyHeader,
                      SegyTraceHeaders,Data,'revision',0);
>> % Revision 1
>> WriteSegyStructure('datacube.segy',SegyHeader,
                      SegyTraceHeaders,Data,'revision',1);
```

4.2.2. Force Data Sample Format

See Section 2.3.3 for a list of valid and supported values for the datasample format `dsf`.

```
>> % To force the use of SEG Y revision 0 and data sampling format IEEE :
>> WriteSegyStructure('datacube.segy',SegyHeader,
                       SegyTraceHeaders,Data,'revision',1,'dsf',5);
```

Chapter 5. Acknowledgement

Thanks to Brian Farrelly, Norsk Hydro Research Centre, Bergen, Norway, for supplying functions to convert between IBM Floating Point format and doubles.

Thanks to *Sourceforge* (http://sourceforge.net/) for hosting the project.