

New MATLAB® functions for reading, writing and modifying SEG-Y files

Heather Lloyd, Kevin Hall, and Gary Margrave



Abstract

The SEG_Y_Toolbox for MATLAB is a new set of tools that allow SEG-Y files to be read, modified, and written. This toolbox has been written using an object oriented approach. This means that each of the parts of a SEG-Y file have been made into an object which allows for simpler programming and better methods of decoding the file. To comply with the SEG-Y Revision 1 standard, this toolbox has used the definitions laid out for the text, binary and trace headers. As most modern computers natively have ASCII and IEEE formats, and the SEG-Y Revision 1 standard allows these formats, CREWES has opted to save the text header in ASCII format and the trace data in big endian IEEE format. The new tools are distributed as part of the CREWES Matlab toolbox, and are in the segy/Segy_Toolbox directory. In order to use these tools, you will need to download the toolbox, install it, and update your Matlab path.

Acknowledgments

The authors would like to thank everyone currently or formerly in CREWES, who have previously worked on the Matlab SEG-Y tools. A large part of the code used in the current project was developed by Chad Hogan. Other contributors include Henry Bland and Carla Osborne.

References

Norris, M. W., Faichney, A. K., 2002, SEG Y rev 1 Data Exchange format, SEG Technical Standards Committee. SEG.

Tools for Reading and Writing

Tools Used for Reading a File

- **SEG_Y_read** – This tool reads in the entire SEG-Y file
- **SEG_Y_readHeader** – This tool only reads in the headers of the data. You can then select which traces you want, and read them into Matlab with SEG_Y_getTraces.
- **SEG_Y_getTraces** – This tool only reads in the trace data and must be given the TraceHeader object that contains the desired traces. This is useful when you want a specific shot or offset, etc.
- **SEG_Y_readMulti** – This tool will read in multiple SEG-Y files at a time and appends the data from each file to the main data. For this to work the data must be from the same job, and is assumed to have the same byte order and file system throughout all files.

Tools Used for Writing a File

- **SEG_Y_write** – This tool will write an entire SEG-Y file given a text header, binary header, trace header, and trace data information. These can be in the form of SEG-Y objects or as basic Matlab matrices as described in the help file.
- **SEG_Y_writeHeaders** – This tool will only write the binary, text, and extended headers to the file. Use SEG_Y_writeTraces to write the trace headers and trace data to the file.
- **SEG_Y_writeTraces** – This tool will only write the trace header and trace data to a file and appends to the end of a file so that multiple shot records can be written to the same file.

Tools for Modifying and Decoding

Tools Used for Modifying Data

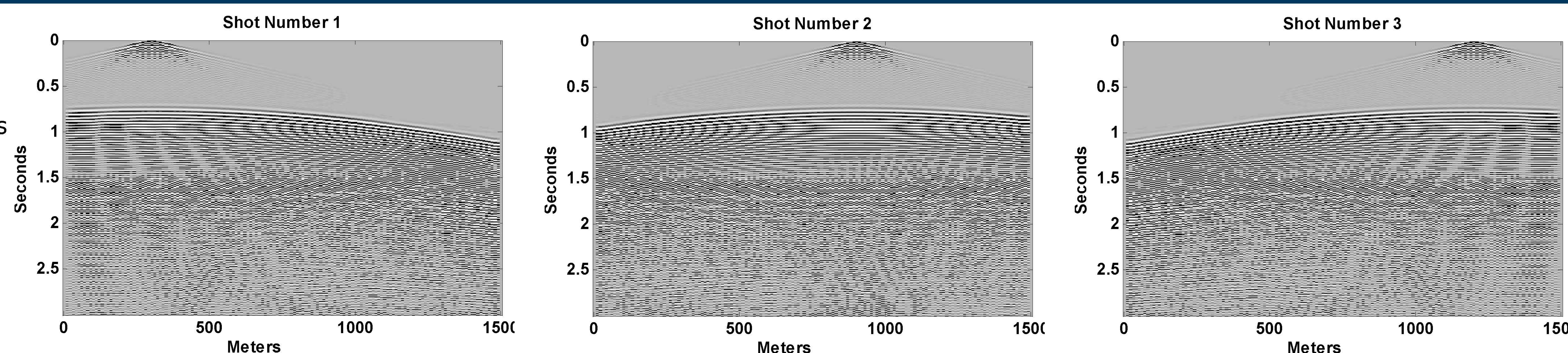
- **SEG_Y_EditTextHeader** – This tool allows the user to create or modify a TextHeader Object through a graphical interface. This function will output a 40 x 80 character array that can be used with the write functions in this toolbox.
- **SEG_Y_getData** – This tool retrieves the trace data from a Trace Object and puts it into a Matlab matrix that the user can then interrogate.
- **SEG_Y_getHeader** – This tool is used to retrieve header information from a TextHeader, BinaryHeader, TraceHeader, or Trace object. For the numerical headers an additional word is required that is listed in the definitions file for that object.
- **SEG_Y_setHeader** – This tool is used to save header information in a TextHeader, BinaryHeader, TraceHeader, or Trace object.

Tools Used for Decoding a Non-Standard SEG-Y File

- **SEG_Y_StandardizeHeader** – This tool allows the user to create a set of definitions that will decode the binary or trace header in a SEG-Y file. This is especially useful if the SEG-Y file does not follow the SEG-Y Revision 1 Standard.
- **SEG_Y_endianSwap** – This tool will swap the bytes of header data or trace data that have been read in with the wrong byte order.

Example

In the CREWES toolbox there are tools that use finite differencing algorithms to create shot records. It would be useful to make several shot records and write them to a SEG-Y file.



Each of these figures represents a shot record that was created using the Finite Differencing Toolbox in CREWES tools. These figures were created as described in step 5 of this example.

1. Create the variables for the shot records.
 - `velocity=1500*ones(101,151); velocity(51:end,:)=4500;`
 - `dx=10;`
 - `dtstep=(dx*(1/sqrt(2))/max(velocity))-0.001;%For stability`
 - `dt=0.002;`
 - `tmax=3;`
 - `dx=10;`
 - `xrec=0:dx:1500;`
 - `zrec=0*xrec;`
 - `[wavelet,twave]=ricker(dt,40);`
2. Plan the seismic experiment. There will be 3 shots at 300m, 900m and 1200m. Each shot will have a receiver every 10 meters along the line for a total of 151 receivers per shot.
 - `shotpos=[300,900,1200];`
 - `ns=length(0:dt:tmax);`
 - `numtraces=length(xrec)*3; % number of receivers * number of shots`
3. Create the binaryheader and textheader using AFD_makeSEG_Yheaders, a utility that will return values that can be used with the SEG_Y_Toolbox.
 - `[texthead,binaryhead]=AFD_makeSEG_Yheaders(dt,ns,numtraces,1,3)`
4. Write the binaryheader and textheader to a SEG-Y file using SEG_Y_writeHeaders.
 - `fileout='seismicexp1.sgy';`
 - `SEG_Y_writeHeaders(fileout,texthead,binaryhead)`
5. Loop over the number of shots to create the synthetic finite difference shot records, create traceheaders using AFD_makeSEG_Ytraceheaders, and write the traces and the traceheaders to the SEG-Y file;
 - `for k=1:length(shotpos)`
 - `snap1=zeros(151,1);snap1((shotpos(k)/dx)+1)=1; snap2=snap1;`
 - `[seismogram,seis,t]=afd_shotrec(dx,dtstep,dt,tmax,velocity,snap1,snap2,... xrec,zrec,wavelet,twave,1);`
 - `tracehead=AFD_makeSEG_Ytraceheaders(xrec,ones(length(xrec),dt,...`
 - `shotpos(k),1,k);`
 - `SEG_Y_writeTraces(fileout,{tracehead,seismogram},length(xrec)*(k1,`
 - `k-1);`
 - `end`
6. The SEG-Y file has now been created.

