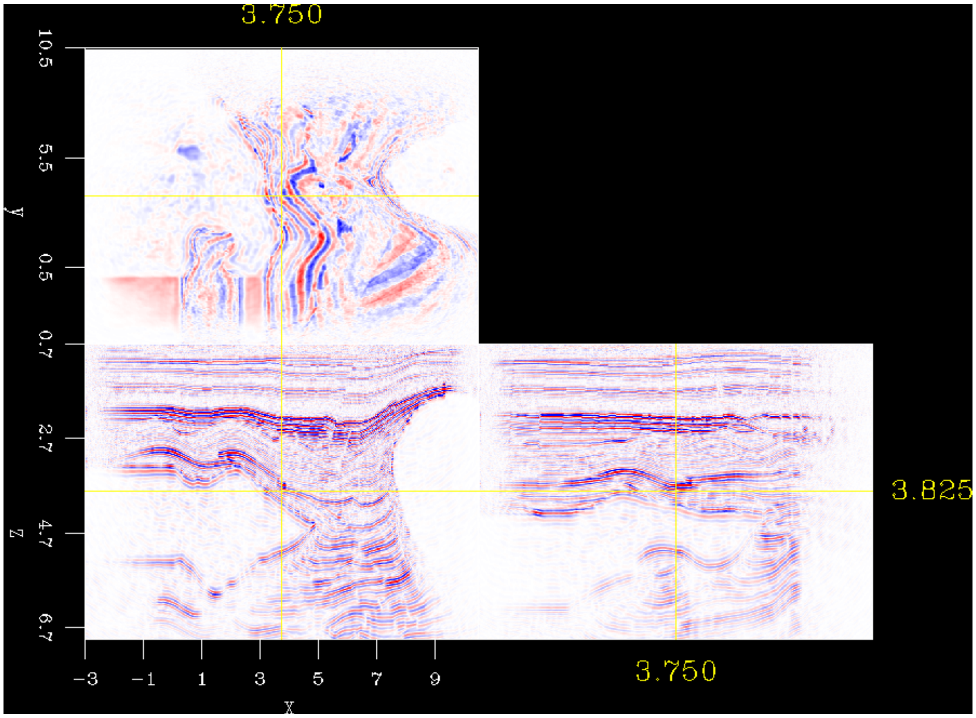
Seismic imaging is a tool for constructing subsurface images using sound waves. These techniques are the industry standard for understanding subsurface geology. Such an analysis may help geologists identify oil and gas reservoirs hidden beneath the earth's surface. Thousands of ultrasensitive receivers are laid out on the earth to record the sound waves as they echo within the earth. A single seismic experiment involves broadcasting sound energy through the subsurface and recording a signal. Some of the energy transmitted into the earth reflects due to differences in subsurface properties (e.g. density, composition) as a function of position. A seismic survey involves conducting thousands of different experiments varying the positions of the source and/or receivers.

Reverse Time Migration (RTM) is one process to create an image of the subsurface in terms of position (x,y,z) from the seismic data which is a function of source and receiver position and time. RTM involves conducting two computational experiments. In both cases, the wave equation is used to propagate a wavefield in time using different forcing functions. In the first phase, the source signature is injected into the earth and propagated forward in time. The second phase uses the data recorded at the receivers as the forcing function and propagates backward in time. An image of the subsurface is updated by multiplying the source and receiver wavefields at every x,y,z.



Instructions for compiling the code

Prerequisites:

A Linux machine with at least gcc-4.8, with C++ and Fortran installed. Intel TBB is also necessary.

1) Clone and install required dependency

```
git clone http://zapad.Stanford.EDU/bob/genericIO.git /opt/genericIO
mkdir -p /opt/genericIO/build
cd /opt/genericIO/build; cmake -DCMAKE_INSTALL_PREFIX=/opt/genericIO  ..
make install
```

2) Clone the code

```
git clone http://bob@zapad.Stanford.EDU/SEP-external/Born.git /opt/Born/src
```

3) Install the code

```
cd /opt/Born
mkdir build
cd build
export CXX=g++
cmake -DgenericIO_DIR=/opt/genericIO/lib -DCMAKE_INSTALL_PREFIX=/opt/Born ..
make
make install
```

Reverse Time Migration (RTM) is one process to create an image of the subsurface in terms of position (x,y,z) from the seismic data which is a function of source and receiver position and time.

Example data is available at https://storage.googleapis.com/sc17-data/send.tgz. In the example, two seismic experiment shots are included. In a real life application, there are thousands of such shots. Each shot will produce a float binary file of the same size and the final result is to take all of these float binaries and to sum them up. An example script example.py, showing the steps of running two shots imaging, is included.

Some details of the tools involved are described below.

a. Using the program "RTM3D" to image a single shot

```
RTM3D json=mig.P
```

b. The “Add” program can add two files together and is designed to run with JSON parameter files. A parameter file to add two files image1.json and image2.json would look like

```
{
"in1": "image1.json",
"in2": "image2.json",
"out": "combo.json"
}
```

Assuming that is saved into the file add.P

```
Add json=add.P
```

would be used to combine the two files.

c. The final image can be visualized using the display tool.

Display tool of the generated datasets

Prerequisites:

A Linux machine with at least gcc-4.8, with C++ and Fortran installed. Intel TBB and qt-devel are necessary. It also depends genericIO that is built in compiling the RTM code.

1) Clone and install required dependency

```
git clone http://zapad.Stanford.EDU/bob/genericIO.git /opt/genericIO
mkdir -p /opt/genericIO/build
cd /opt/genericIO/build; cmake -DCMAKE_INSTALL_PREFIX=/opt/genericIO  ..
make install
```

2) Clone and install the display tool

```
git clone http://bob@zapad.Stanford.EDU/bob/QTCube.git /opt/qtcube/src
mkdir -p /opt/qtcube/build
cd /opt/qtcube/build
cmake  -DgenericIO_DIR=/opt/genericIO/lib -DCMAKE_INSTALL_PREFIX=/opt/qtcube ../src
make install
```

To view datasets on a linux machine

```
Sep_cube out.json json=1
```

where out.json is the file you want to look at.