



Department of Earth and Planetary Sciences  
Northwestern University

Winter 2018

# TsuBox

v2.0

A Toolkit for Ray-Tracing Tsunamis

Amir Salaree

## Contents

<b>1</b>	<b>Foreword</b>	<b>2</b>
<b>2</b>	<b>Introduction</b>	<b>3</b>
<b>3</b>	<b>What Can You Do with TsuBox?</b>	<b>5</b>
3.1	Components of TsuBox . . . . .	6
3.2	Pros and Cons . . . . .	7
<b>4</b>	<b>Getting TsuBox to Work</b>	<b>9</b>
4.1	How to Get TsuBox . . . . .	9
4.2	System Requirements . . . . .	9
4.3	Installation . . . . .	9
<b>5</b>	<b>Using TsuBox</b>	<b>11</b>
5.1	Bathymetry Grids . . . . .	11
5.1.1	Smoothness . . . . .	11
5.1.2	Velocity Grids: depvel . . . . .	12
5.2	Source Types . . . . .	15
5.2.1	Point Sources . . . . .	15
5.2.1.1	Input Parameters . . . . .	16
5.2.1.2	Running the Program . . . . .	16
5.2.2	Finite Sources . . . . .	18
5.2.2.1	Fault Line . . . . .	18
5.2.2.2	Okada Deformation Field . . . . .	24
5.2.3	Ray-Tracing Landslide Tsunamis . . . . .	31
5.2.4	Ray-Tracing Arbitrary Static Sources . . . . .	32
5.3	General Remarks . . . . .	34
<b>6</b>	<b>Making Movies</b>	<b>35</b>
6.1	movray . . . . .	36
6.2	movray.sort . . . . .	37
6.3	movray.most . . . . .	38
6.4	movray.mostfront . . . . .	39
<b>7</b>	<b>Future Plan</b>	<b>41</b>
<b>8</b>	<b>List of Programs</b>	<b>42</b>

# 1 Foreword

The TsuBox package is a modified and streamlined version of the algorithm originally developed by Woods & Okal (1987) in their effort to consider tsunami propagation as a bundle of energy rays. While the main idea is inherited from Fumiko Tajima's algorithm for ray-tracing on a sphere (Tajima & Garmany, 1987), the current program is limited to a box and considers only first passages, stopping rays when they hit the boundaries of the box.

The TsuBox package is designed to accommodate finite as well as point sources. Many of the results and graphics from TsuBox are cool to use and watch, and this is just the primary step in a promising area of research.

In compiling this package and putting together this manual, I am immensely indebted to Emile Okal and Seth Stein at Northwestern University for many thought-provoking chats and their great ideas. I am also grateful to Eddie Brooks and Nooshin Saloor for their invaluable comments on the text.

**Amir Salaree**  
**December 2018**



## 2 Introduction

Propagation of tsunamis in oceans can be examined in some detail by ray-tracing experiments, following the efforts by Woods & Okal (1987) and Satake (1988). The ray-tracing technique is based on Jobert & Jobert's (1983) solution of the 2-D Eikonal equations on a heterogeneous sphere using a field of variable propagation velocities in the shallow-water approximation.

By assuming a radius of  $a$  for the Earth, and considering a Mercator transformation (Cartesian coordinates of  $\phi$  and  $z$  as longitude and latitude), the result is a set of first order differential equations:

$$\begin{aligned} a \partial_S \phi &= \sin i \\ a \partial_S z &= \cos i \\ k a \partial_S i &= -\sin i \partial_z k + \cos i \partial_\phi k \end{aligned} \tag{1}$$

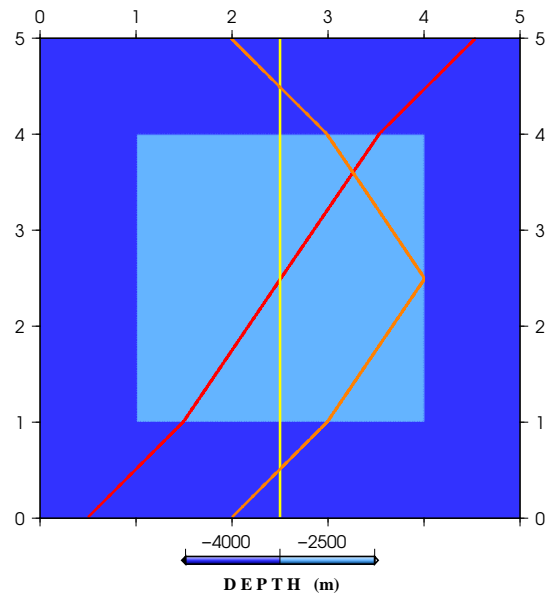
where  $i$  and  $k$  are the incidence angle and the modulus of the local wave vector.<sup>1</sup> Eq. (1) can be numerically solved for a given velocity field in the form of a regular grid. Such grids are built by calculating the group or phase velocity of tsunamis for a given bathymetry grid (see section 5.1.2).

TsuBox uses a 4<sup>th</sup>-order Runge-Kutta scheme to solve the system of equations in Eq. (1). As a result, at each time step, the program will try to estimate the next location of the emanating energy by iterating over the equations. The final product will be a *ray* which has propagated from a starting point until it is terminated at an arbitrary location. TsuBox ends the rays either upon their arrival at the shoreline or at the boundaries of velocity map.

The process mentioned above is similar to following a light beam while passing through media with different velocities or refractive indices (Fig. 1). The main difference is that here the speed of rays in each medium is dictated by its depth as we recall from the hydrodynamic gravity waves (see section 5.1.2). A direct consequence of this analogy (also evident in Eq. (1)) is *reciprocity* as a result of which the path taken from point **A** to point **B** for a given ray is identical to the path from point **B** to **A**.

---

<sup>1</sup> For more detail see Jobert & Jobert (1983).



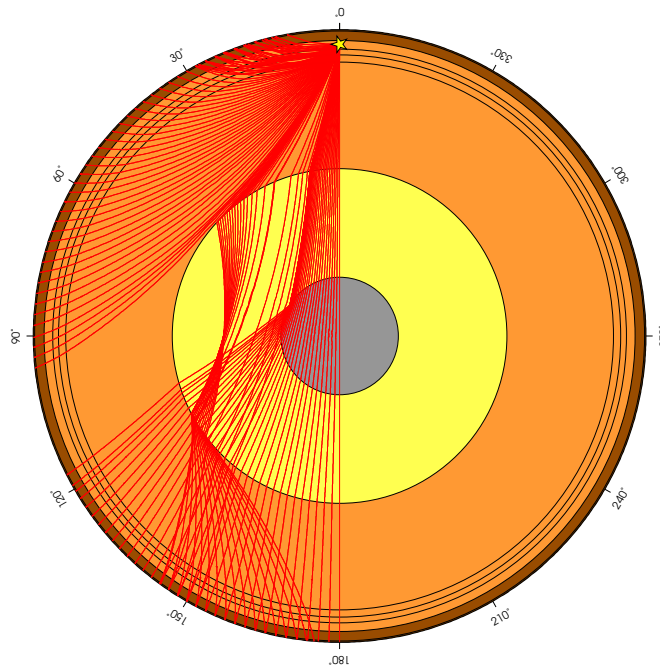
**Figure 1:** Rays traveling through a hypothetical ocean with only two depths. This is very similar to three light beams passing through a cubic prism.

The ray-tracing method has been successfully applied several cases of tsunamis [see for example, Woods & Okal (1987), Satake (1988), Ward (2001), and Salaree & Okal (2018)].

### 3 What Can You Do with TsuBox?

Ray patterns produced by TsuBox are good representations of tsunami propagation. However, as is common practice in seismology, once using the ray representation of the wavefield, all the information regarding the wave amplitude is discarded. For instance, while Fig. 2 provides invaluable information regarding the propagation of seismic energy from an earthquake through the Earth, it implies *almost* no information about amplitude of the propagated energy.

The “almost” disclaimer above refers to the old idea that the focusing of rays points to a higher concentration of energy and therefore *may* imply higher amplitudes (e.g. Aki & Richards, 2002).

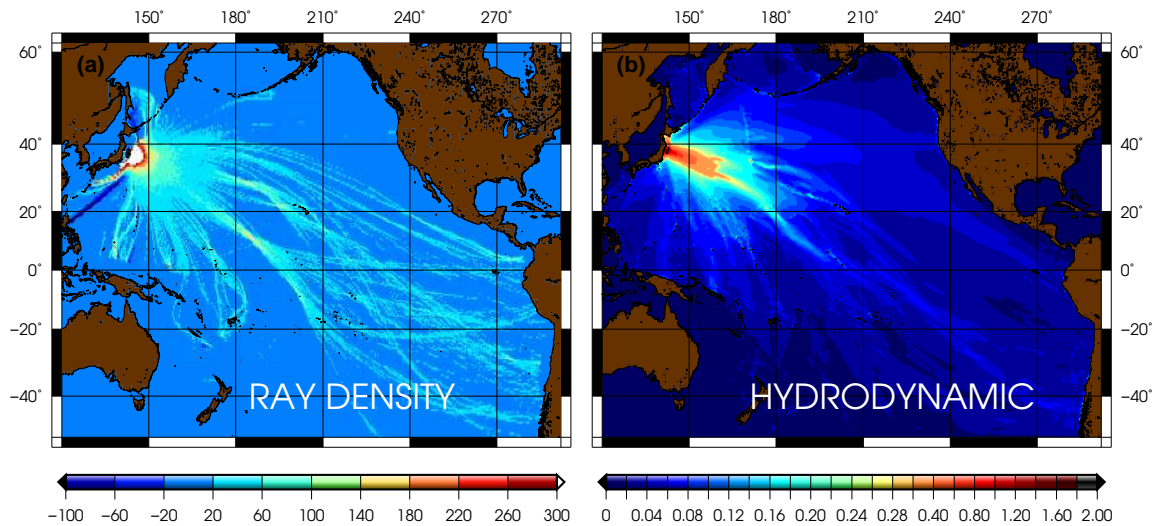


**Figure 2:** Energy rays of phases  $P$ ,  $PKP$ , and  $PKIKP$  from an earthquake traveling through the Earth with PREM structure (Dziewonski & Anderson, 1981) created by the Taup program (Crotwell et al., 1999).

Obtaining amplitude information requires solving the full equations of energy propagation, and in the case of tsunamis the gravity wave version of Navier-Stokes equations describes the behavior of waves. Such equations are, for the most part,

nonlinear and do not have analytical solutions, hence requiring numerical schemes. The computer programs used in doing so are time-consuming, as they often use finite-difference methods to solve the equations for all the bathymetry grid points at each time step.

An interesting workaround to propagating the full energy trail is to propagate *select* pockets of energy in specific azimuths. The latter will have the form of *rays* propagating from a source. Using TsuBox, we can study the two methods as shown in Fig. 3 for the 11 March 2011 Japan tsunami. Fig. 3 compares the energy field (distribution of maximum amplitudes) from the MOST algorithm (Titov et al., 2016) and the highest concentration of ray propagation from a similar source.



**Figure 3:** (a) Ray density computed by TsuBox; color palette corresponds to the number of ray points in  $0.5^\circ \times 0.5^\circ$  grid cells. (b) Distribution of maximum amplitudes from a hydrodynamic simulation of the 2011 Japan tsunami using the MOST algorithm.

### 3.1 Components of TsuBox

Below are brief descriptions for each of the main components of the TsuBox package.

- **tsubox:** The main `tsubox` program can be used to trace the tsunami rays from a single point source in a bathymetry grid for a given ocean.

- **AUTOTSU:** This set of programs is used to ray-trace finite sources. This is done by considering such sources as combinations of points. The program considers finite fault lines and 3D surface deformation fields known as Okada solutions.
- **movray:** Using GMT and FFMPEG, this control script creates movies of ray propagation from single point sources. The final product is shown on a blank background.
- **movray.most:** Similar to **movray**, this control script generates movies, but on a background of snapshots from hydrodynamic simulations.
- **movray.mostfront:** Similar to **movray.most**; adds a *propagation front*.
- **movray.sort:** Similar to **movray.mostfront**; removes the foldings of the wavefront caused by abrupt variations in velocity. No hydrodynamic background is needed.

These features use many other programs in creating the outputs. A complete list of these programs can be found at the end of this document.

## 3.2 Pros and Cons

While Figs. 3a and 3b are not exactly identical, Fig. 3a exhibits the main propagation trends in Fig. 3b. In other words, ray-tracing results such as that shown in Fig. 3a are useful for quick evaluations of energy propagation from tsunami sources. For instance, while the necessary computations to produce Fig. 3a take  $\sim 10$  minutes, the full hydrodynamic solution in Fig. 3b requires a computation time of  $> 2$  hours<sup>2</sup> on a simple desktop computer.

The graphical portion of the program is also great for making educational movies. This is especially useful as the package allows for featuring the results from hydrodynamic simulations in the cartoons and therefore enabling the user to compare the efficiency of these methods.

Interesting propagation features can also be identified in TsuBox outputs. For instance, the folding of the wavefront on itself as a result of changes in velocity is a good demonstration of the *triplication* effect frequently observed in seismology. In this regard, TsuBox demonstrates how bathymetry acts a focusing or defocusing agent (similar to lenses in geometrical optics) and thus exposes some regions to higher tsunami energy.

---

<sup>2</sup> Both are done for an ETOPO5 bathymetry grid.

Such figures, also allow curious eyes to figure out why features such as diffraction cannot be achieved by ray representation of the energy field; a feature that can play a critical role in studying the tsunami propagation in certain areas.

On the down side, we strongly advise the user to avoid interpreting figures such as Fig. 3a literally. In other words, although very tempting to think so, such maps do not provide much information about tsunami amplitude and should only be used as a guide in understanding the tsunami propagation behavior.

## 4 Getting TsuBox to Work

### 4.1 How to Get TsuBox

You can download a copy of the TsuBox package at `www.sites.northwestern.edu/salaree/tsubox`. Alternatively, you can send me an email at

`amir@earth.northwestern.edu`

for a link to the program.

### 4.2 System Requirements

The following system components are required to successfully run TsuBox.

1. **Unix:** This version of the program has been written for Unix distributions. It has been successfully tested on Linux and OSX.
2. **bash:** All the control scripts run under *bash* shell in Unix.
3. **gfortran:** FORTRAN 90 (or above) compiler is used to create executable files.
4. **GMT 4.X:** Used to create plots. If you are using more recent versions of GMT, you may need to modify the plotting schemes accordingly.
5. **ImageMagick:** Used to convert the graphics from GMT.
6. **FFMPEG:** Used to create movies from the snapshot outputs.

### 4.3 Installation

Copy the compressed tar file containing the program to a directory of your choice on your computer. Extract the tar file with something like

```
tar -xvf tsubox.*.tar.gz
```

A new directory named `TSUBOX` will be created. In this directory you will find the following:

- `install`: A control script used to compile the program.
- `AUTOTSU`: The main control script to run the finite source simulation.
- `smylie.in`: An example of the input needed for the Okada filed program.
- `src/`: The directory containing the source files.
- `REPO/`: The directory where all the executables will be compiled into.
- `TsuBox.pdf`: The User manual plus some documentation for TsuBox – this document.
- `README`: Brief documentation file: a few quick heads-ups.

To get the contents of the `examples` directory, see

<https://www.dropbox.com/s/zde49ey2t2o1k4y/examples.tar.gz>

To compile the programs, change directory (`cd`) to `TSUBOX` and run the `install` script:

```
./install
```

The `install` script will look for the necessary ingredients (see section 4.2) and compiles the FORTRAN programs. In case of any errors, feel free contact me.

You will need to run the program in the main `TSUBOX` directory as in this version, all the paths are relative.

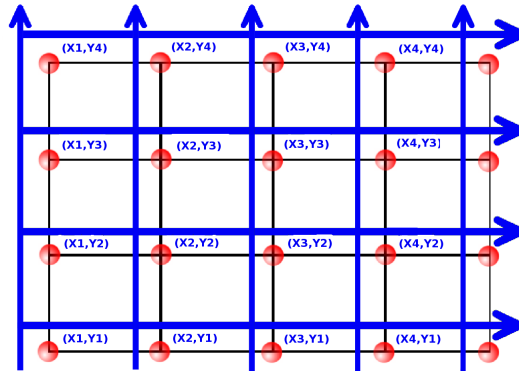


## 5 Using TsuBox

### 5.1 Bathymetry Grids

As in any tsunami simulation tool, a bathymetry grid is necessary while running any component of TsuBox. Such a bathymetry grid can be either extracted from the published datasets such as ETOPO# and GEBCO, or put together by digitizing marine charts. The input bathymetry must be a three column ASCII file in rising XYZ format with longitude, latitude and elevation (or depth) on the X, Y and Z columns (see Fig. 4). An example of such grids for the the Pacific ocean (`pacific.xyz`) is included in the `examples/` directory.

It is important to be careful about this format for the input files especially since many of grid handling programs (such as the GMT package) generate grids with descending latitude. Please note that there is no constraint on the numerical formatting of the values, as long as they are listed in the above order.



**Figure 4:** Structure of the bathymetry grids used by TsuBox. Note the index order as they will be listed in the ASCII file following the  $(i,j)$  indices.

#### 5.1.1 Smoothness

As briefly discussed in section 2, TsuBox uses a Runge-Kutta scheme to solve the raypath equation (Eq. (1)). Despite its efficiency and speed, this method requires a well-behaved function and will therefore fail in the case of bathymetry grids with

very abrupt changes. *TsuBox is very sensitive to sharp variations in bathymetry.* Therefore, it is often a good idea to *smooth* the bathymetry grid before feeding it to the TsuBox.

For this purpose, **grdsmooth** is included in the REPO directory. **grdsmooth** smoothes the grid by applying a moving average smoothing agent throughout its span. To use this program, simply type:

```
./REPO/grdsmooth [grid_file]
```

Upon execution, the program will ask for a *smoothing percentage* which is simply the ratio of the number of points in the smoothing window to the entire grid points. Usually, the level of smoothness depends on what you expect to get out of the simulation and is best found through trial and error. In general, 0.02% is a good starting point.

The smoothing percentage in **grdsmooth** does NOT mean how much the grid is smoothed, but is simply a measure of the number of grid points around each point that will attribute to its “smoothed” value. For instance, with a smoothing percentage of 10%, a large bathymetry grid may lose *all* its details. *The moving average process used in **grdsmooth** is not linear.*

As examples of the smoothing effect, ray-tracings of the 2011 Japan tsunami on the smoothed ETOPO5 bathymetry grids are shown in Fig. 5. Note that the although TsuBox has managed to simulate the rays, the output is not conclusive.

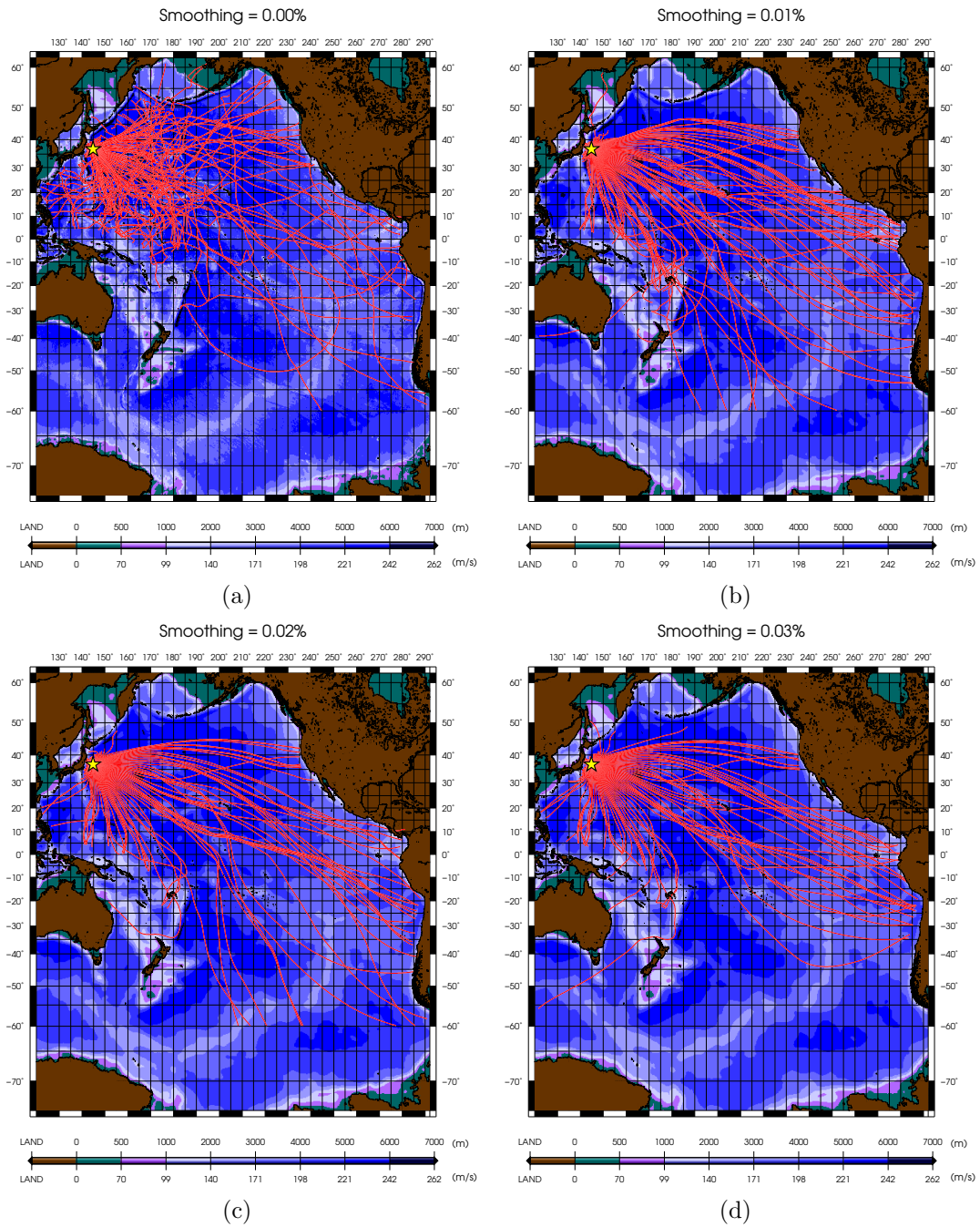
### 5.1.2 Velocity Grids: depvel

In order to calculate raypaths, TsuBox can used either group or phase velocity. While group velocity is usually the best choice in most cases, for dispersive tsunami propagation, one can consider phase velocity. ASCII grids for both phase and group velocities are calculated by a program called **depvel**. Upon execution, **depvel** will ask for coastline altitude as well as fault length. Coastline altitude will help TsuBox scrape lands off the general topography grid. Fault length is a measure of the frequency in question while using phase velocity (see, for instance, Rabinovich, 1997).

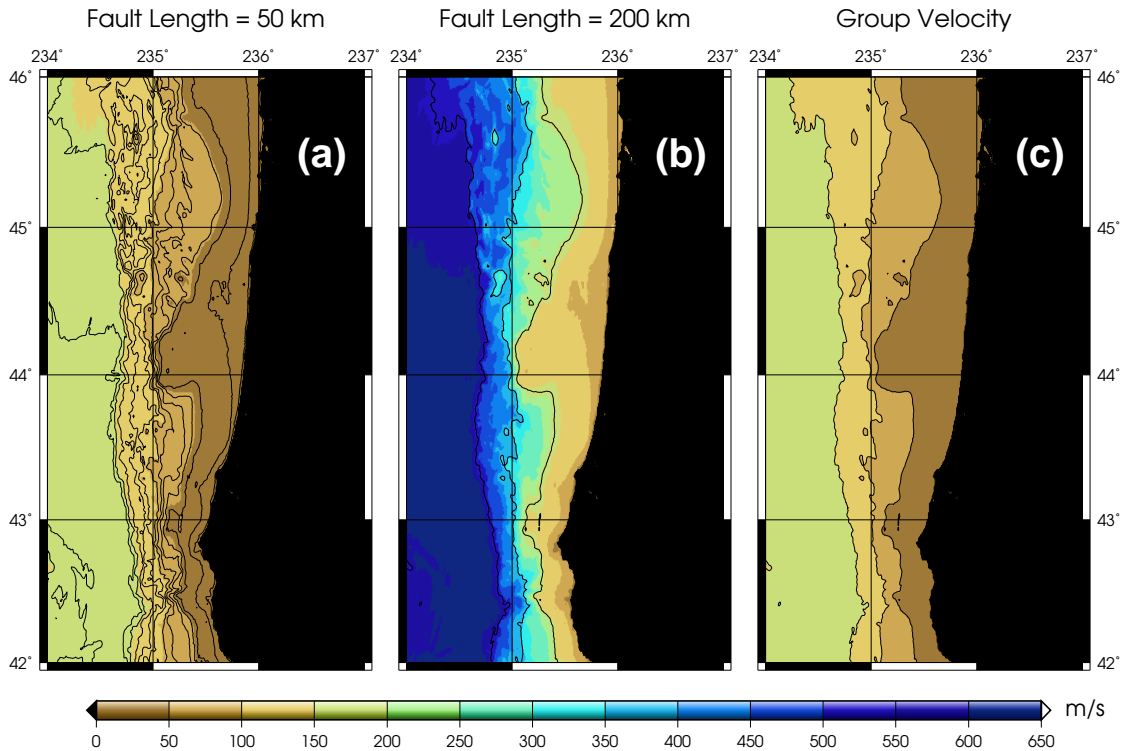
To use the **depvel** program in the TsuBox directory, simply type:

```
./REPO/depvel [bathymetry_file]
```

This will produce three files: `gvel.out` which is a grid file of group velocities, `pvel.out` which is a grid file of phase velocities, and `grid.dim` that is a file with information regarding the grid dimensions. While `grid.dim` is necessary for TsuBox to run, you can use either of `gvel.out` or `pvel.out` depending on your choice of ray-tracing program. Fig. 6 compares the distribution of group velocities as well as two phase velocity maps. Note that at shorter fault lengths (higher frequencies) the velocity distribution approaches that of group velocities as predicted from the shallow water approximation (see, for example, Sandanbata et al., in press).



**Figure 5:** Ray-tracing of the 2011 Japan tsunami in the Pacific Ocean for (a) original ETOPO5, (b) 0.01% smoothed, (c) 0.02% smoothed, and (d) 0.03% smoothed grids. The bathymetry is color-coded according to depth ( $\equiv$ tsunami speed).



**Figure 6:** Off the coast of Oregon; maps of (a) *group* velocity as well as (b) *phase* tsunami velocity for a 50 km fault, and (c) *phase* velocity for a 200 km fault.

## 5.2 Source Types

TsuBox can ray-trace two types of sources as described below.

### 5.2.1 Point Sources

This component is appropriate for smaller sources, or when source dimension is negligible compared to the geometry of important features. It is also useful for a quick evaluation of the tsunami propagation. This is done by the program `tsubox` for non-dispersive propagation and `tsuboxp` for dispersive cases. To use this feature, you will need the following components:

- `grid.dim`: a file from `depvel` containing some information regarding the grid.
- A velocity grid: `gvel.out` for `tsubox` and `pvel.out` for `tsuboxp`.
- `tsubox.in`: a file containing input parameters.

**5.2.1.1 Input Parameters** The file `tsubox.in` (any arbitrary name will do) has the following structure:

```

37.297 144.800      Source lat. , Source lon. ==> Epicenter
-60.000 62.000      Lat. min., Lat. max. of map
120.000 292.000     Lon. min., Lon. max. of map
0                  Azimuth of the central ray
180                Number of rays to be traced
2                  Increment in azimuth between rays
1                  Time step for actual computation (seconds)
600                Time step for redemption & plotting (should be a multiple of step)
3600               Time step for tick mark (seconds; should be a multiple of delt)
JAPAN              Arbitrary text for book-keeping

```

### *Computation Time Steps:*

Choosing an appropriate time step for TsuBox (line 7 of the input file) is a tricky business. In principle, it can be calculated following the CFL principle in finite-difference schemes. However, it is also going to depend on how well-behaved the bathymetry grid is. A program called **CFL** is included in the package for this purpose. **CFL** suggests several time steps (in percentage of the largest possible value) to be used as input to TsuBox. To run **CFL** in the TsuBox directory, type

```
./REPO/CFL [XYZ_bathymetry_file]
```

While larger values of  $\delta t$  may lead to wrong calculations<sup>3</sup>, very small values may cause the program to crash or take much longer to run. For finite sources (see below), you would need to set this value in the `tsubox.in` file placed in the `REPO` directory.

**5.2.1.2 Running the Program** To run TsuBox for single sources, create a separate directory in the root TsuBox folder and `cd` to it. Then, simply type:

```
../REPO/tsubox [parameters_file]
```

---

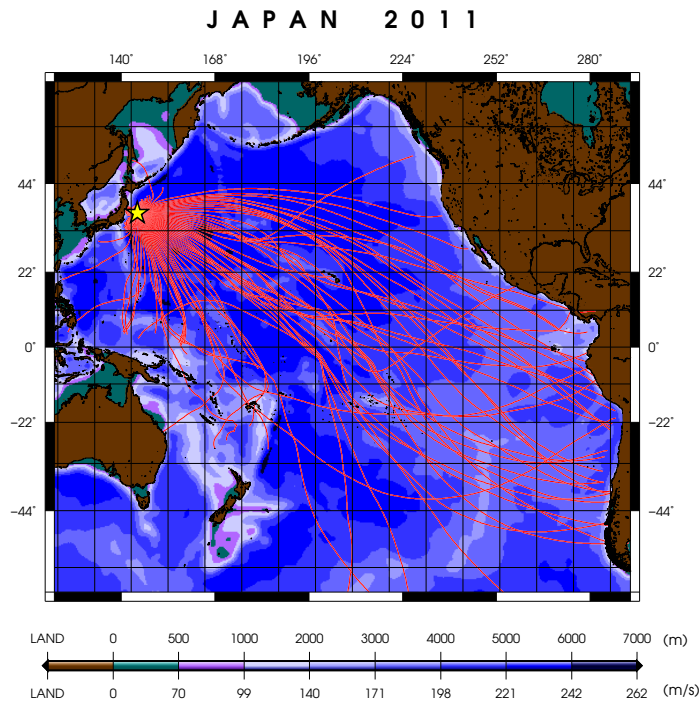
<sup>3</sup>The main idea behind this constraint is that it is simply illogical to predict the future before having data from past and present. One cannot move faster than the propagation of information in a grid to compute the propagation position.

TsuBox will process the individual rays in verbose mode. After a successful run, you can plot the rays on a background of velocity field using the `mapvel` script. To do this, type:

```
../REPO/mapvel [plot_title]
```

Remember that no spaces are allowed in the plot title unless you put them within **single quotes**.

To make a background of phase (instead of group) velocity, change the `vfile` variable in the first line of `mapvel` to “pvel.out” (without quotations). Fig. 7 is an example of TsuBox output map using the input parameters listed above and a 0.02% smoothed ETOPO5 bathymetry map of the Pacific (pacific.xyz in the examples directory).



**Figure 7:** Ray-tracing of the 2011 Japan tsunami as a point source using the input parameters (`tsubox.in`) listed above.

Although `tsubox` and `tsuboxp` are very quick tools in ray-tracing tsunamis,<sup>4</sup> they may be inadequate especially for large sources where one cannot ignore the faulting/deformation size.

### 5.2.2 Finite Sources

The `AUTOTSU` control script can be used to ray-trace finite sources. As discussed before, this is done by considering such sources as sets of points. TsuBox provides two methods to depict finite sources:

1. Fault line
2. Okada deformation field

In both cases, arbitrary numbers of points are used to approximate the deformation. While the in first method, the deformation is simply considered as a linear array of points, the second method uses an approach known as Okada solution to calculate the static ocean bottom deformation as the result of the earthquake. The latter process will create two poles of displacement with opposite polarities.

TsuBox then runs the ray-tracing process for each individual point, and finally calculates the “density” of ray points (plus count for “positive” poles and minus count for “negative” poles) in arbitrary geographic cells (e.g.  $0.5^\circ \times 0.5^\circ$ ), hence providing a measure for amplitude (see section 3).

#### 5.2.2.1 Fault Line

TsuBox will need a “fault line file” which should contain coordinate pairs (lon,lat) of points in the fault trace. The user can choose a handful of points and the program will fill in the gaps between them if the user deems it necessary (Fig. 13a).

#### *Notes on the Number of Points:*

It goes without saying that using more points will result in more realistic simulations. However, the user must keep a few points in mind: First, using more points will result in longer computation time: with the *luxury of time*, people may choose to use the time-consuming hydrodynamic simulation codes such as MOST. Secondly,

---

<sup>4</sup> They take about 10 s to run on a desktop computer.



adding more points does not always improve your results since after a certain limit, the rays will simply start to overlap – especially in the case of smaller earthquakes.

In the TsuBox root directory, type:

```
./AUTOTSU [bathymetry_file]
```

You will be prompted for a “zero” ocean level. For free oceans, the free surface is at  $z = 0$  m. You will then be able choose between an *earthquake* (e) or a *landslide* (l) sources.<sup>5</sup> For the scope of our interest, let’s choose *earthquake*. Then for a fault line, choose (1). You will now need a fault file which is a two-column ASCII file listing fault points (lon,lat). This file can be only a coarse sampling of the real fault; the program will ask you whether you would like to fill more point into the trace through linear interpolation. It is usually a good idea to use a more detailed fault (instead of only a handful of points). However, keep in mind that a large number of points is not always great (see the Notes above).

### Note on Fault Line Sorting

TsuBox automatically sorts the fault line as it were a grid (rising lat and lon). Although the sorting process might do fine, it is may create unwanted artifacts by changing the order of fault points. This issue is more prominent while trying to create a movie of the rays from a moving point source on a fault line (i.e. propagating rupture). To overcome any such issues, feel free to edit the **faultseg** program to change/remove the sorting subroutine.

Using simple scripts, you can make interesting movies of rays from a propagating ruptures out of the created snapshots in the **Fxxxx** directories. An example of such scripts **getpic** is included in the **REPO** directory. A video created using this script for the 2004 Sumatra tsunami is available at <https://youtu.be/FRmsxIdd7oY>.

The program will then need to create a velocity grid by using **depvel**. It will then go through the fault points and ray-trace each of them. The rays from individual points will be stored in **Fxxxx** directories where *xxxx* is simply a 4 digit number corresponding to the fault points used in the simulation.

---

<sup>5</sup>The landslide is approximated as a dipole with static upward or downward motions, based on Synolakis et al. (2002). For more information, see Salaree & Okal (2018).

After successfully simulating the rays, the program will prompt for a cell dimension (in degrees) to calculate the *ray density*. This is done automatically through the **denflt** program which will produce an XYZ file named **density.out** with longitude, latitude and the number of rays in each cell.

### Note on Choosing Density Cell Size

Choosing a very small cell will – although not significantly – increase calculation time. It will also result in underestimating the ray density throughout the grid since some of the propagation features may fall well within the chosen precision. At the same time, choosing too large a cell, will result in coarse sampling of trends. Therefore, one should always come up with an optimized value. Unfortunately, there is not a unified approach to automatically define a threshold which would depend on grid size and resolution, source dimensions, and what you would like to do with the grid.

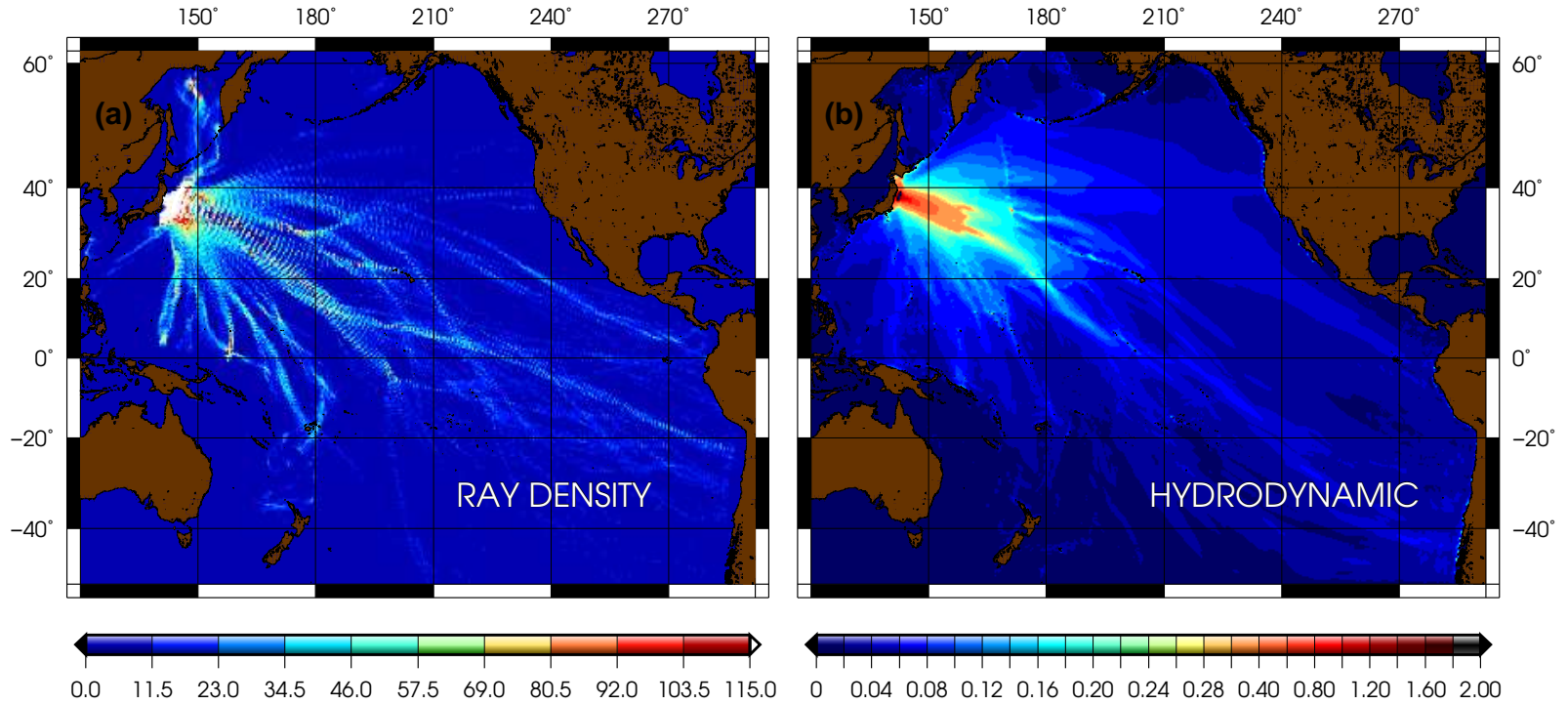
For instance, while for the large fault line chosen in the examples below,  $0.5^\circ \times 0.5^\circ$  seems to be appropriate, a smaller fault will require larger density cells as such a source creates a smaller number of rays. While choosing an appropriate cell size needs some experience, you should not worry too much about it. Although by decreasing cell size you will inherently decrease the number of rays, at the same time you are breaking artificially large concentrations into smaller chunks (i.e. cells). You can later adjust your maps concentration by tweaking the color palette at line #61 of the **plot.sh** script in REPO.<sup>6</sup>

Finally, you will be asked if you would like to make maps to visualize your outputs. The latter is automatically done by the **plot.sh** script in the REPO directory. As an optional input, you can include a raster file named **max.grd** from your hydrodynamic simulations to be used as background in your maps. As a result, five output maps (in three formats: ps, eps, and jpg) are created, as shown in Figs. 8 to 12.

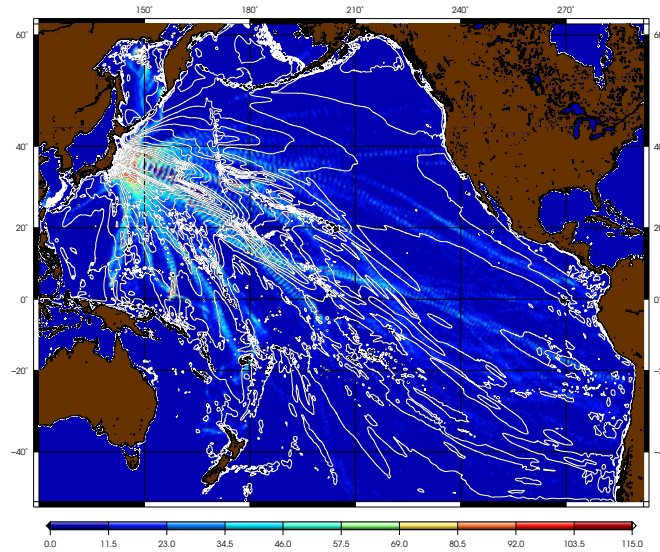
In general, the fault line representation of the tsunami source is faster to run compared to the deformation field. However, the the Okada solutions produce results more similar to the full hydrodynamic simulations as it considers the source polarity. The latter reproduces an effect similar to the source directivity which is dominantly observed in tsunami simulations (see Fig. 10).

---

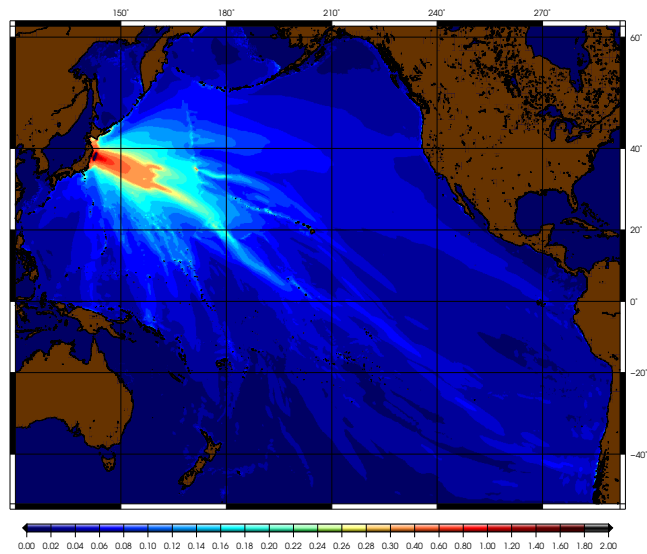
<sup>6</sup> Alternatively, you can flip the comment sign between the two *makecpt* commands in the script and manually set a threshold.



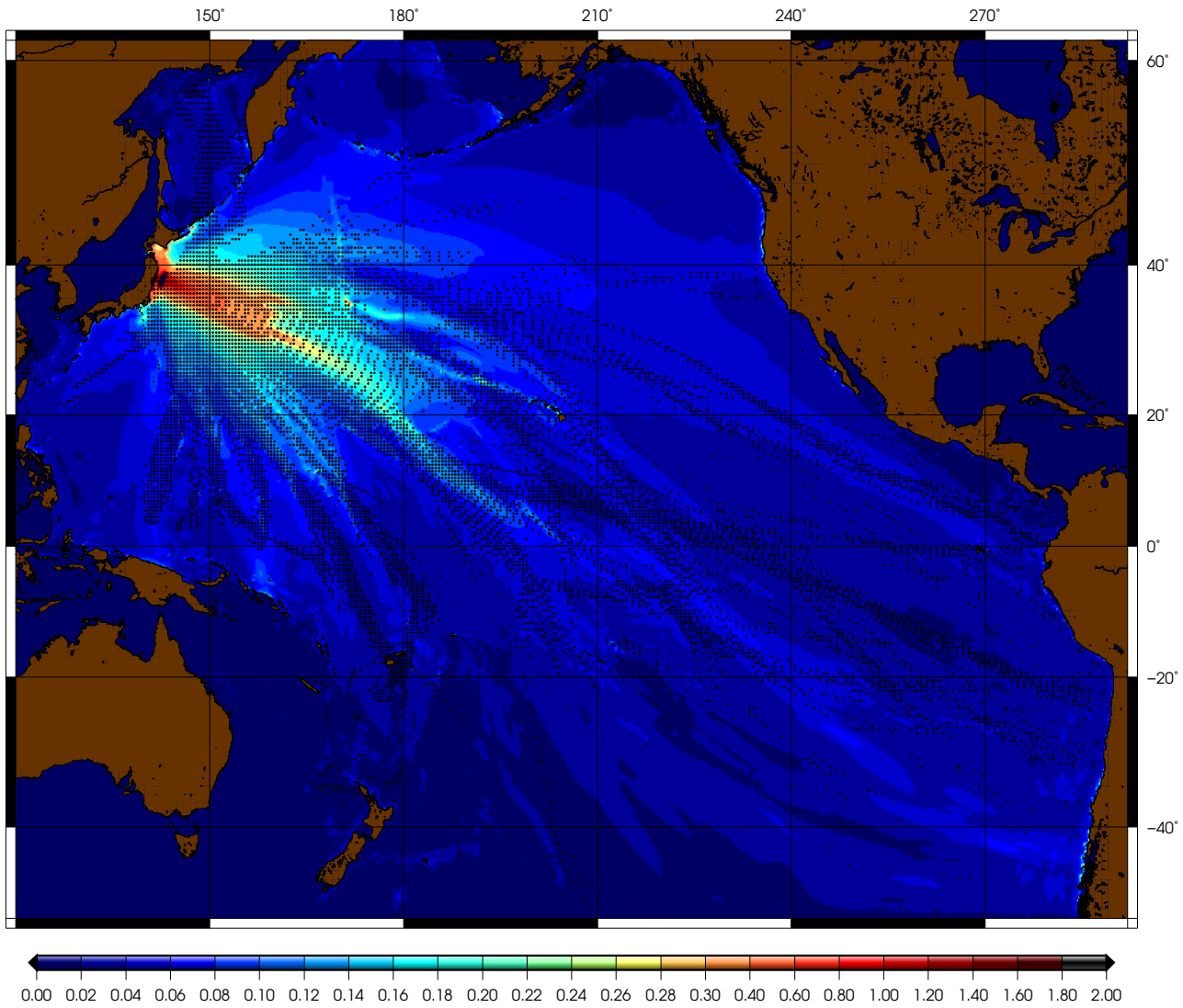
**Figure 8:** (a) Calculated ray density color coded according to the number of rays in each  $0.5^\circ \times 0.5^\circ$  cell. (b) The field of maximum amplitudes from a hydrodynamic simulation for a source with similar dimensions using the MOST algorithm color coded with wave height in meters. This image is created as *den.eps*. The right-hand side panel is only produced is a *max.grd* file exists.



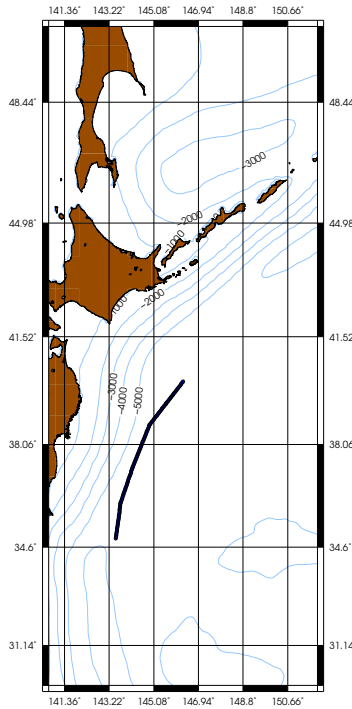
**Figure 9:** Calculated ray density color coded according to the number of rays in each  $0.5^\circ \times 0.5^\circ$  cell with contours of maximum amplitude distribution from the hydrodynamic simulation. This image is named *comp1.eps* and is only created if a *max.grd* file exists.



**Figure 10:** The right panel from Fig. 8 plotted separately. This image is created as *comp3.eps* and is only made if a *emphmax.grd* file exists.



**Figure 11:** Calculated ray density for the cells with a number of rays larger than the median, superimposed as black dots on the hydrodynamic simulation result as background. In the absence of a *max.grd* file, the background will be left as blank. This image is created as *comp2.eps*.

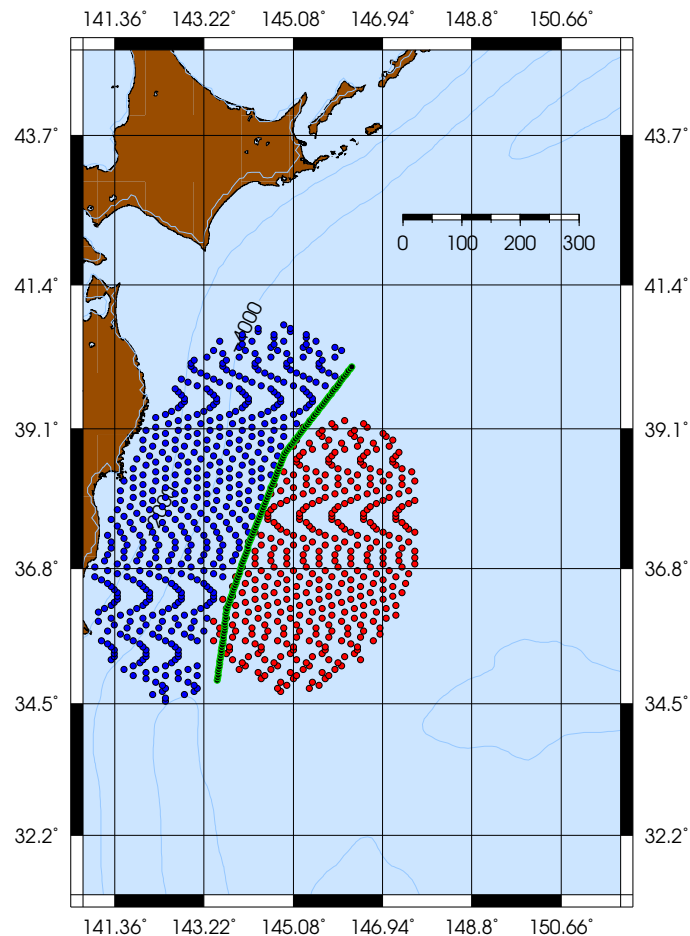


**Figure 12:** A zoomed view of the fault points (dark blue) used in the simulation on a background of bathymetry contours. This image is named *bathy.eps*.

In the end, you will be asked if you want to clean up the working directory by removing the temporary files. This is a good idea if you are not trying another run, since some of the time-consuming steps (such as generating the velocity grid) will be avoided by using some of the files from previous runs.

### 5.2.2.2 Okada Deformation Field

TsuBox calculates the ocean bottom static deformation field using an algorithm developed by Mansinha & Smylie (1971). This process will usually output a large number of points for both positive (“hump”) and negative (“trough”) deformation poles. The user will have the option to decimate the number of points in each (Fig. 13b).



**Figure 13:** The 2011 Japan tsunami source: the decimated Okada deformation field with upward (positive) motions in red and downward (negative) motions in blue based on the CMT solution and the fault line (green line) inferred from the Okada field.

Similar to the case of the fault line, invoke **AUTOTSU** by typing:

```
./AUTOTSU [bathymetry_file]
```

and choose *earthquake* after setting a zero level. Upon the prompt, choose (0) to create an Okada field using the **smylie** program. Remember that for this part you will need a file named **smylie.in** containing input parameters for the deformation program. If this file does not exist, you will have to manually input these parameter.

`smylie.in` has the following structure.

```

7.25 4.05 3.0   P-wave velocity, S-wave velocity, density of rocks
001           Arbitrary number for your model
20           Source depth
1           Hypocenter (1) or Centroid (0)
530          Seismic Moment in units of 10**27 dyn-cm
12 420 210    Dislocation (m), fault length(km), fault width (km)
203 10 88    Strike, dip, rake
-900 900 201  Min. dist (km) west of the epicenter, Max. dist east, leave 201
-900 900 201  Min. dist (km) south of the epicenter, Max. dist north, leave 201

```

You can avoid the hassle of assigning fault dimensions on line 6, by replacing the three values by a single slash (/). In this way, the program will calculate fault dimensions using seismic scaling laws (Geller, 1976).

The last two lines determine the geographic extent of deformation. As the vertical dislocation will drop by moving away from the epicenter, you can use smaller windows to smaller earthquakes (e.g. `-300 300`).

After creating deformation field, you will now have to superimpose it on your bathymetry grid. This is automatically done by the **superdisloc** program.

After creating a velocity grid, the program will try to generate a reasonable number of points. This is done by sifting through the deformation grid and removing very small and zero deformation amplitudes. Therefore, you are asked for *cut-off amplitudes*.<sup>7</sup> You can select these so-called cut-off thresholds according to the maximum and minimum values displaced.

Good values for the CMT solution of Japan earthquake (used in the `smylie.in` file above) with `min.= -2.62 m` and `max.= 4.95 m` are `-0.2 m` and `0.2 m`. You can choose any other values within reason. As a result of these choices for the Japan earthquake, 3198 positive points (hump) and 3205 negative points (trough) will be created. To further decrease the number of points, you can decimate them by choosing (1) at the prompt. While for faster computation, you can choose something like 32 points for each of the hump and trough, you can choose larger numbers such as 320 points for each if you are willing to wait longer.

After this point, TsuBox will ray-trace the individual points in the hump and trough stacks and finally will calculate a ray density for cells of arbitrary sizes using

---

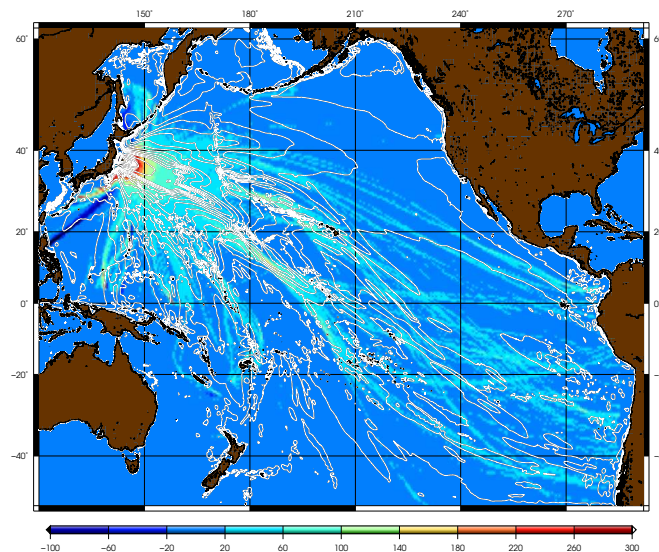
<sup>7</sup> Please choose (1), otherwise the program will run forever without actually achieving anything!



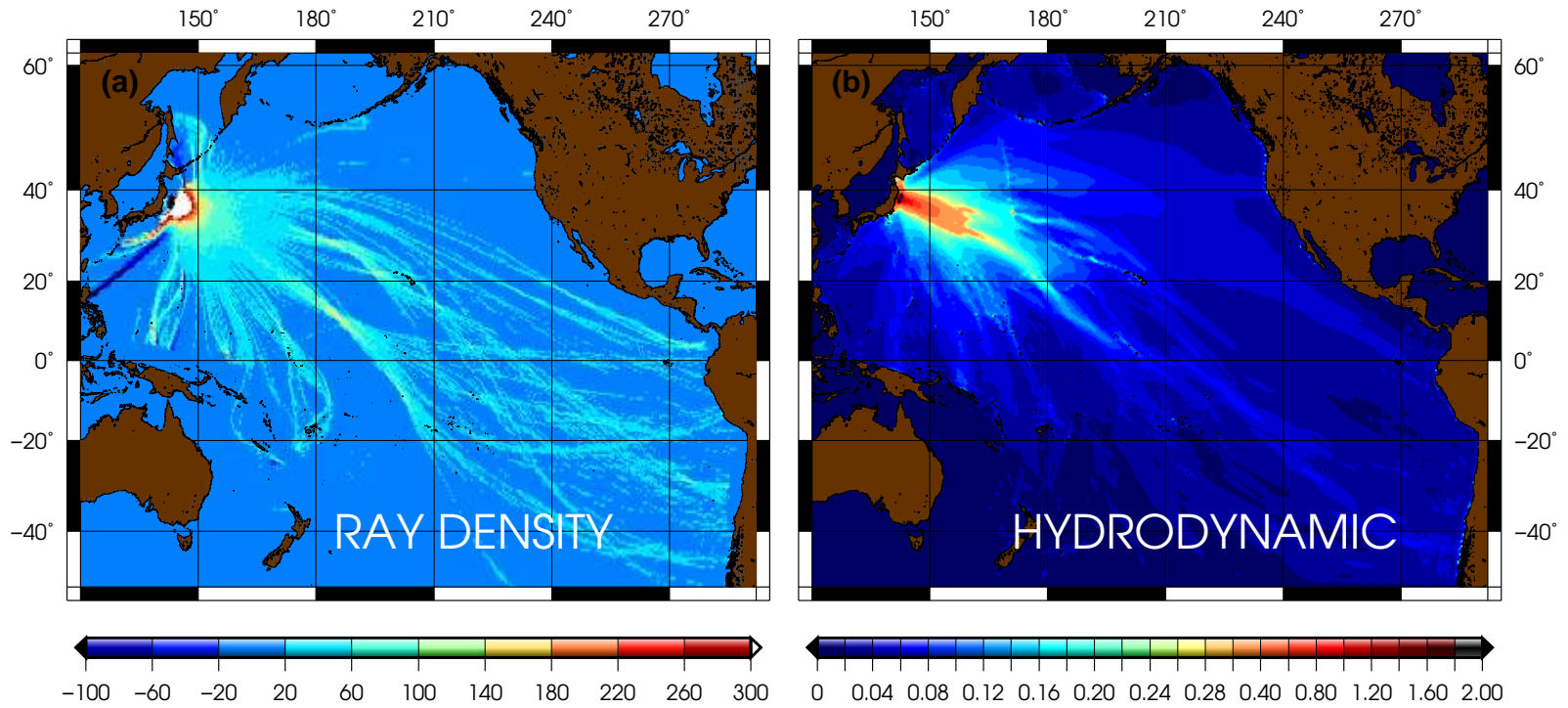
the **dencomp** program. While calculating ray density for each cell, **dencomp** will add the points from hump rays and subtracts the points from trough rays from the total calculated number.

Finally, you will be asked if you would like to make maps to visualize your outputs. The latter is automatically done by **plot.sh** in the **REPO** directory. As an optional input, you can include a raster file named **max.grd** from your hydrodynamic simulations to be used as background in your maps. As a result, five output maps (in three formats: ps, eps, and jpg) are created, as shown in Figs. 14 to 18.

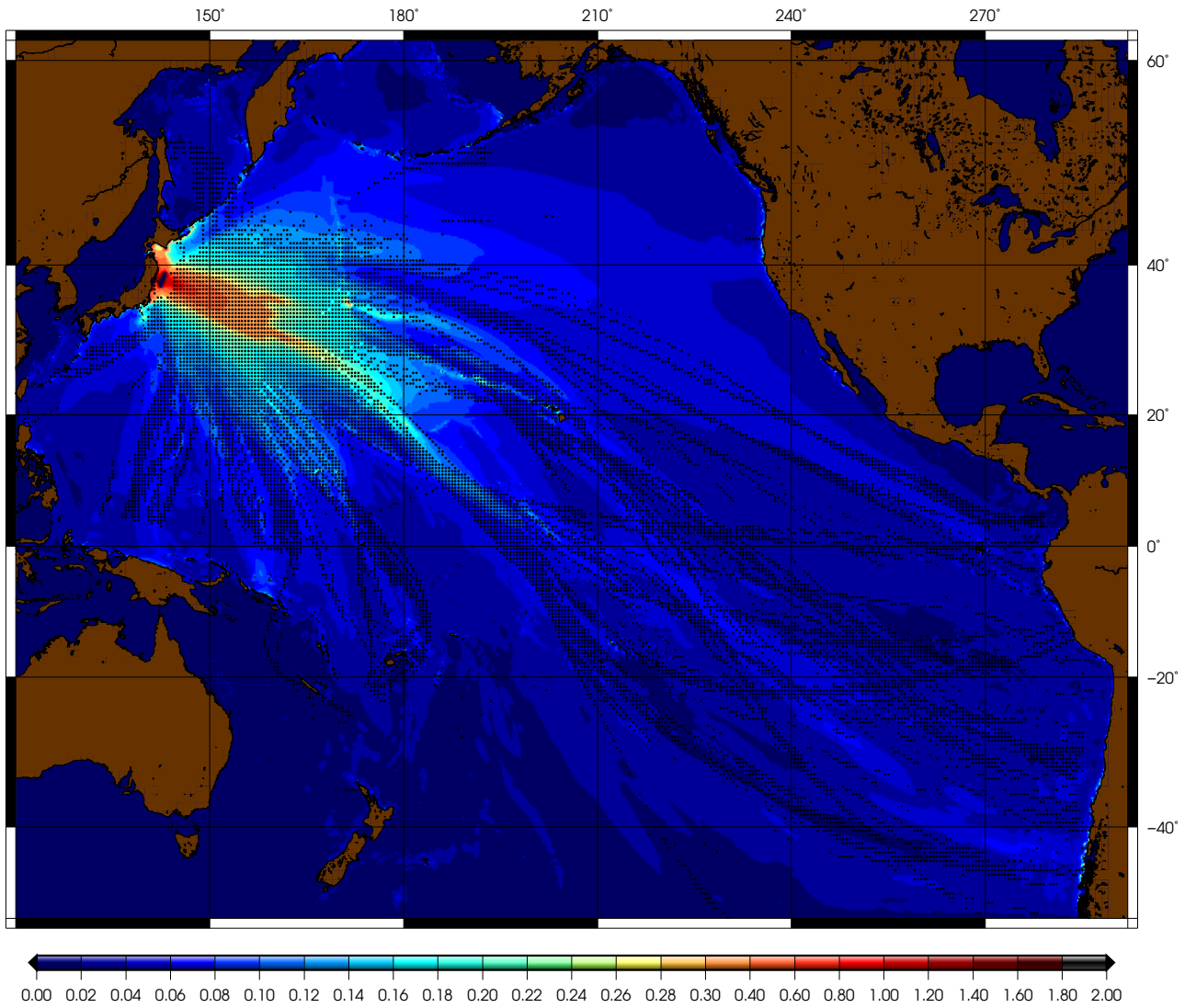
Note that similar to the fault line option, the rays from individual points will be stored in **Hxxxx** (for hump) and **Txxxx** (for trough) where *xxxx* is simply a 4 digit number corresponding to the fault points used in the simulation.



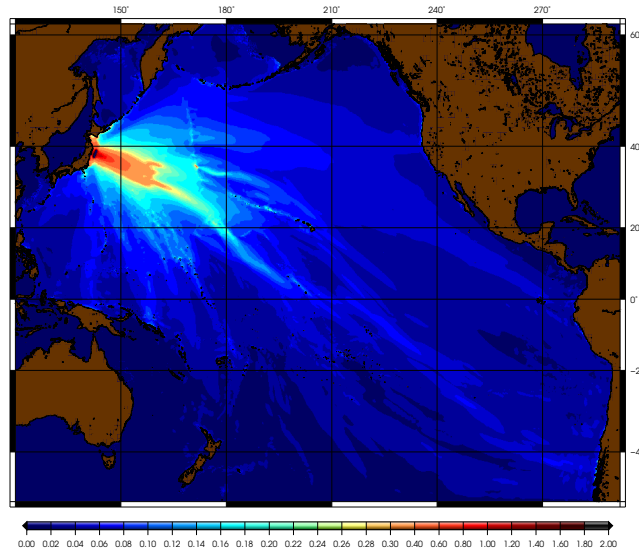
**Figure 15:** Calculated ray density color coded according to the number of rays in each  $0.5^\circ \times 0.5^\circ$  cell with contours of maximum amplitude distribution from the hydrodynamic simulation. This image is named *comp1.eps* and is only created if a *max.grd* file exists.



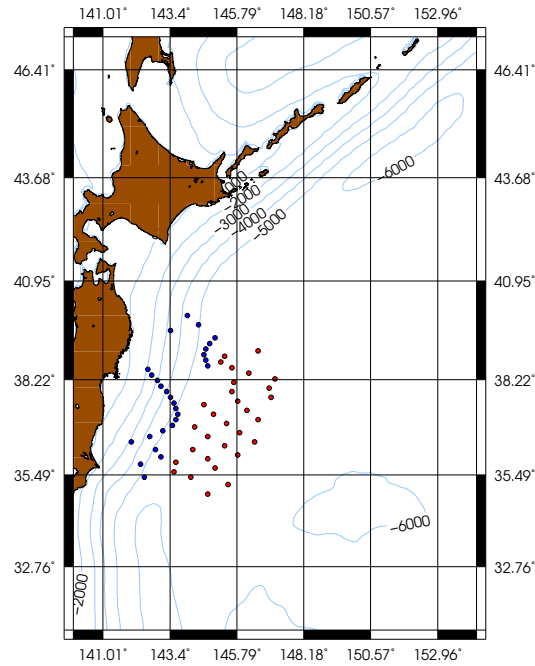
**Figure 14:** (a) Calculated ray density color coded according to the number of rays in each  $0.5^\circ \times 0.5^\circ$  cell. (b) The field of maximum amplitudes from a hydrodynamic simulation for a source with similar dimensions using the MOST algorithm color coded with wave height in meters. This image is created as *den.eps*. The right-hand side panel is only produced is a *max.grd* file exists.



**Figure 16:** Calculated ray density for the cells with a number of rays larger than the median, superimposed as black dots on the hydrodynamic simulation result as background. In the absence of a *max.grd* file, the background will be left as blank. This image is created as *comp2.eps*.



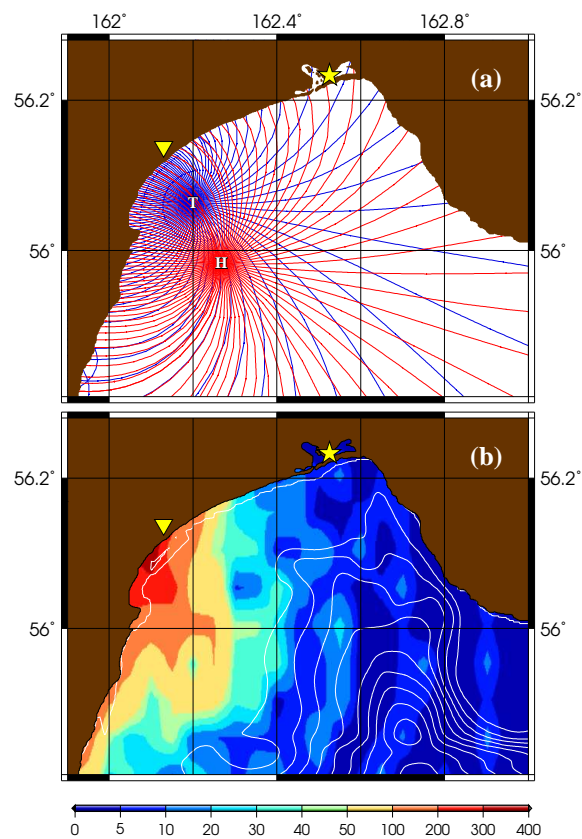
**Figure 17:** The right panel from Fig. 14 plotted separately. This image is created as *comp3.eps* and is only made if a *max.grd* file exists.



**Figure 18:** A zoomed view of the deformation field with red dots representing the upward motion and blue dots showing the downward deformation. This image is named *bathy.eps*.

### 5.2.3 Ray-Tracing Landslide Tsunamis

Landslide tsunamis are poorly understood compare to their tectonic counterparts. However, several studies have tried to simulate their hydrodynamic behaviors using various techniques. To ray-trace landslides using the TsuBox package, we consider them as simultaneous hydrodynamic dipoles (a hump and a trough) which propagate rays as instantaneous sources. Fig. 19 shows ray-tracing of the tsunami from a landslide dipole in the Gulf of Kamchatka (Salaree & Okal, 2018).



**Figure 19:** (a) Ray-tracing of the slide dipole. Blue and red lines show rays propagating respectively from the trough (T) and the hump (H) of the dipole; (b) number of ray points per  $0.05^\circ \times 0.05^\circ$  area calculated from (a). The yellow stars and inverted triangles represent points with maximum reported run-up.

To ray-trace landslide tsunamis, you will need a file named `ellipse.in` with the following structure in the TsuBox directory.

56.064	Latitude of trough
162.200	Longitude of trough
155	Azimuth of slide
10	Lever (km)
25	Depth of trough (m)
17	Height of hump (m)
0.08	Coeff. alpha_ for trough
0.25	Coeff. alpha+ for hump
0.8	Coeff. gamma_ for trough
0.30	Coeff. gamma+ for hump

An example `ellipse.in` with the above information exists in the `REPO` directory. For information on these parameters see Salaree & Okal (2015). This file is used by the `superellipse` program to create surface deformation similar to those made by `superdisloc`. The rest is identical to the Okada deformation process.

#### 5.2.4 Ray-Tracing Arbitrary Static Sources

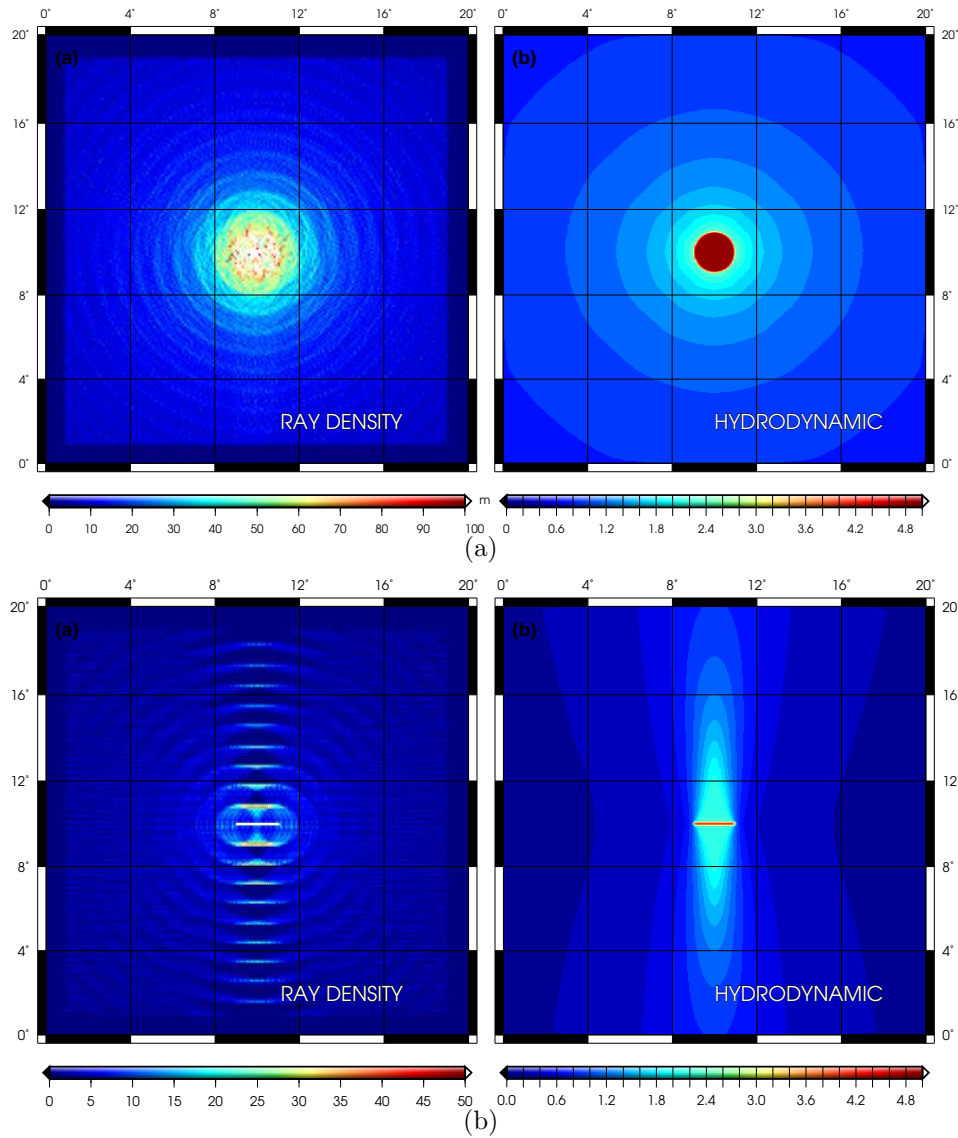
TsuBox can be used to ray-trace pre-designed sources with arbitrary geometry. Such sources can include any configuration of initial vertical displacements in the grid points – considering smoothness. Fig. 20 are outputs of TsuBox comparing ray-tracing and hydrodynamic simulation results for a cylindrical 20(a) and a linear 20(b) static initial deformations.

This is achieved by selecting (o) upon selecting the source type in the program. The program will then look for a deformation file called `burn.dat` containing the initial deformation with the following structure:

```

z(x1,y1) z(x2,y1) z(x3,y1) z(x4,y1) ...
z(x1,y2) z(x2,y2) z(x3,y2) z(x4,y2) ...
z(x1,y3) z(x2,y3) z(x3,y3) z(x4,y3) ...
z(x1,y4) z(x2,y4) z(x3,y4) z(x4,y4) ...
.         .         .         .
.         .         .         .
.         .         .         .

```



**Figure 20:** Comparison of ray-tracing and hydrodynamic simulation of (a) cylindrical, and (b) linear initial deformations.

Please note that you can create sources with only one type of polarity (up or down). Simple programs to create cylindrical and rectangular initial deformations (**supercyl** and **supercube**) are included in the REPO directory. You would need a bathymetry in the form of MOST input (**BATH.dat**) to use these programs.

### 5.3 General Remarks

After running TsuBox (in all cases), ray information for each source point will be stored in a separate directory. You can use this information to look at the data for individual points or to troubleshoot any code crash. Sometimes TsuBox crashes on points at or around abrupt changes in bathymetry. You can make maps of the result for individual points using the **mapvel** script described in section 5.2.1.2.

While the codes and scripts used in TsuBox are designed to work on many systems and cover various applications, they are by no means bulletproof. There may still be many bugs and errors that will only show up while using the package. Therefore a fair understanding of shell scripting, GMT and FORTRAN may come in useful when using TsuBox.



## 6 Making Movies

As was briefly discussed in sections 3.1 and 3.2, TsuBox can be used to create movies of ray propagation. Four types of movie can be made:

1. Rays and a propagation front on a blank background (**movray**).
2. Rays on a background of hydrodynamic simulations from MOST (**movray.most**).
3. Rays and the propagation front on the MOST background (**movray.mostfront**).
4. Rays and a smoothed propagation front on a blank background (**movray.sort**).

These scripts are placed in the `movie/` directory. All these scripts work ONLY for point sources. This is because superimposing all the rays from individual points in finite sources would only clutter the map without any visible pattern. Also, ray-density plots are not time-variant as they only consider the FINAL number of points in each geographic cell. The best way to use ray density maps is to look at them along with hydrodynamic maximum amplitude distributions as they are also accumulation of wave maxima over time.

To run these scripts, the user must make a directory (with an arbitrary name) in the TsuBox directory and `cd` to it, otherwise the relative path will not work. The following parameters can be set in all four scripts.

- **land**: `land= 1` plots continents. To avoid superimposing the continents, set `land= 0`.
- **FR**: sets frame rate. The default is 64. This determines how fast your snapshots will progress in the movie.

The scripts require the `gmtray.xxx` files from the **tsubox** output as well as `tsubox.in` to run. These files should reside in your movie making directory. You will also need the bathymetry file, but TsuBox will ask you for its location if it is not specified in the syntax (see section 6.1).

The `movray*` scripts create snapshots of propagating rays and will splice them together using FFmpeg libraries. While joining the snapshots is usually done in no time, creating the individual snapshots is time-consuming, especially when using hydrodynamic simulations as background. The outputs of these scripts are

- `vel.mp4`: The movie file.
- `vel.xxxx.ps`: Movie snapshots in Postscript (PS) format.

- `vel.xxxx.eps`: Movie snapshots in Encapsulated Postscript (EPS) format.
- `vel.xxxx.jpg`: Movie snapshots in JPG format.

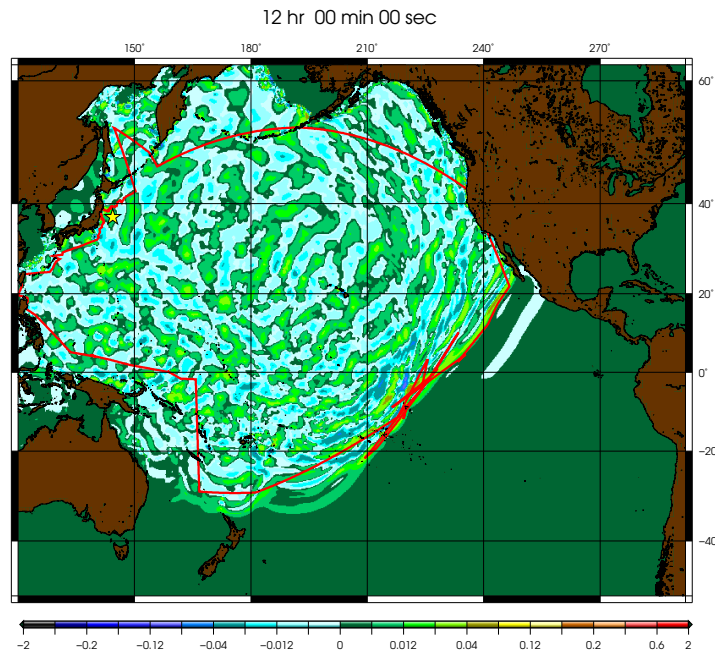
## 6.1 movray

**movray** is a script to create a movie of rays propagating in the bathymetry along with a *propagation front*. To run, type:

```
../REPO/movray [path_to_the_bathymetry_file_directory]
```

If the path is missing from this syntax, the you will be prompted for it. The default is current working directory. A sample snapshot is shown in Fig. 21.

**movray** uses the **-W** flag in GMT's *psxy* to connect the final ray points in each snapshot to create a propagation front. This feature only connects the rays in the order they were first generated at the source, and as a result, does not take into account the distortions caused by velocity changes. Although this may cause distortions and foldings in the wavefront and make the video “look” bad, it is actually a **real** property known as triplication. To remove this property from your movies, use **movray.sort**.



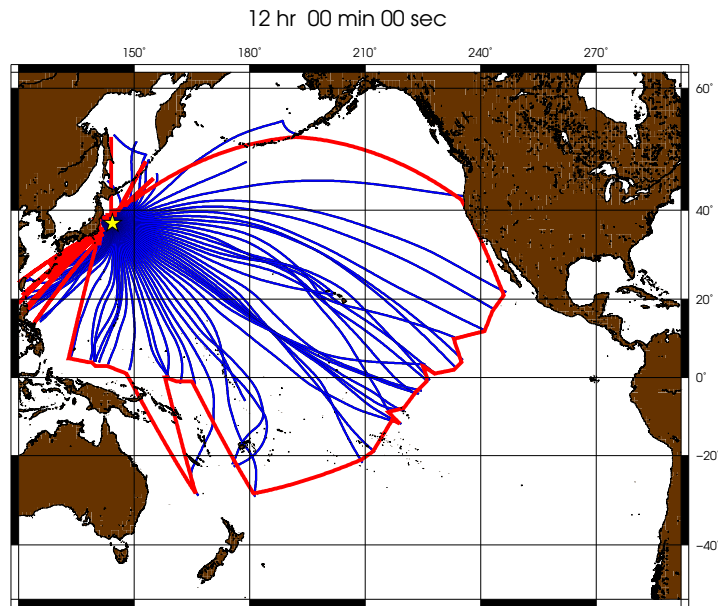
**Figure 21:** The computed rays and *front* for the 2011 Japan tsunami. A triplication of rays can easily be seen at the front.

## 6.2 movray.sort

**movray.sort** creates a movie of smoothed ray propagation front in the bathymetry. The smoothing process is done by sorting the end points with respect to azimuth which is performed by **delaz** and **azsort** programs. To run, type:

```
../REPO/movray.sort [path_to_the_bathymetry_file_directory]
```

You can change the jagged edges of the front curve by adjusting the *psxy* properties. A sample snapshot is shown in Fig. 22.



**Figure 22:** The computed rays and a smoothed front for the 2011 Japan tsunami. A triplication of rays can easily be seen at the front.

### 6.3 movray.most

**movray.most** is a script to create a movie of rays propagating in the bathymetry over a background of MOST simulation. To run, type:

```
../REPO/movray.most [path_to_the_S_files_and_bathymetry_directory]
```

If the path is missing from this syntax, the you will be prompted for it. The default is current working directory. Remember that the MOST S-files and bathymetry need to be in the same directory. A sample snapshot is shown in Fig. 23.

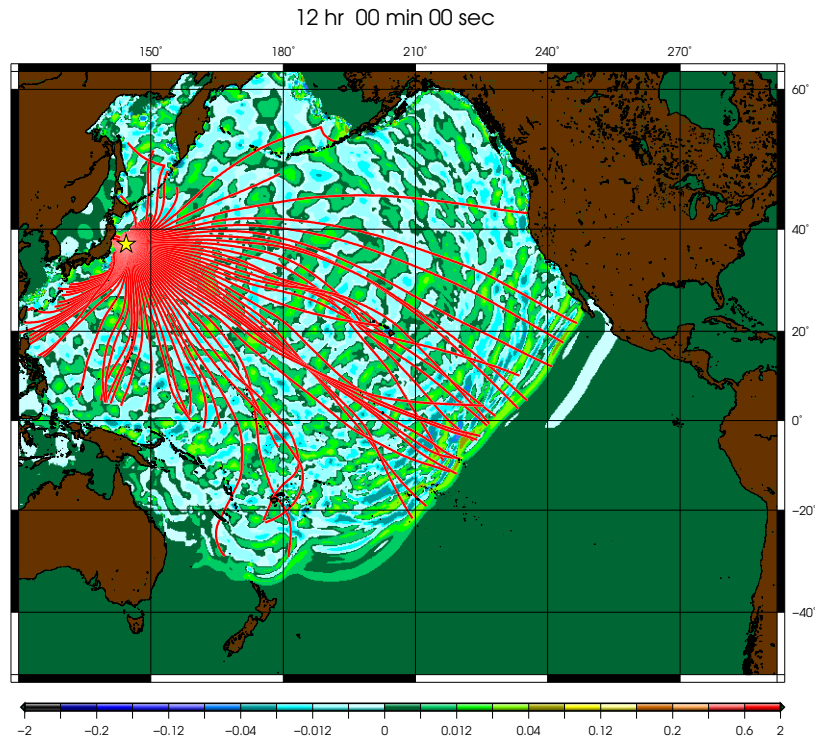


Figure 23: Sample snapshot of `movray.most`.

## 6.4 `movray.mostfront`

`movray.most` is a script to create a movie of rays propagating in the bathymetry over a background of MOST simulation. To run, type:

```
../REPO/movray.most [path_to_the_S_files_and_bathymetry_directory]
```

A sample snapshot is shown in Fig. 24.

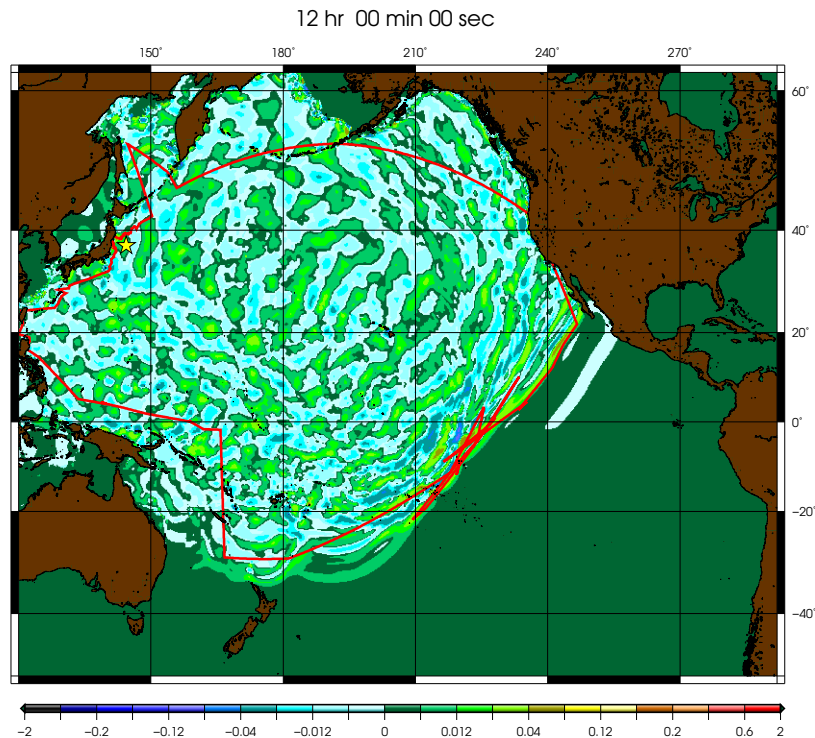


Figure 24: Sample snapshot of `movray.mostfront`.

If the path is missing from this syntax, the you will be prompted for it. The default is current working directory. Remember that the MOST S-files and bathymetry need to be in the same directory.

***Important:***

As you can see in the four movie making scripts, some of the rays have been excluded from the propagation front. These are the “discontinued” upon hitting an island or shallow waters. The exclusion statements which are for the example of Japan earthquake although left in the script, are commented out. You may need to exclude a few rays in other cases.

## 7 Future Plan

Although useful in ray-tracing point and finite tsunami sources, there is still much room for improvement, as some aspects of tsunami behavior are still missing from the package.

An important component to add to TsuBox is the study of temporal evolution of tsunami sources. While this function can be approximated to some degree by looping over point sources over a fault region (see 5.2.2.1), this process can be automated considering a rupture velocity (with size and direction).

A necessary future function of TsuBox would be considering the amplitude of static displacement field while calculating density fields. Seemingly trivial to do, this is actually challenging due to possible violations of shallow water approximation which can result in non-linear superposition of rays.

An interesting consequence of considering the above components in calculations of tsunami displacement fields is the ability to calculate tsunami time-series. This would result in robust and fast approximation energy propagation.

There are also a few known bugs in the package including *rogue* regions cutting the rays by acting like hard boundaries.<sup>8</sup> These bugs must be resolved in the future versions of TsuBox.

---

<sup>8</sup>This is actually resulted in variable handling in the regionalizing subroutine Dziewonski & Steim (1982).

## 8 List of Programs

Below is an alphabetically sorted list of the programs in the package, along with brief description of their purpose.

- **azsort.f** Sorts the rays based on their azimuth.
- **BATH2bath.f** Converts MOST's input format to XYZ.
- **bathprep.f** Converts XYZ bathymetry grids to MOST's input format.
- **burn2xyz.f** Converts deformation from MOST's format to XYZ.
- **CFL.f** Suggest appropriate time steps based on the CFL condition.
- **combunf.f** Converts MOST snapshots to XYZ.
- **delaz.f** Calculates the azimuth & backazimuth.
- **dencomp.f** Computes ray density in geographic cells for Okada field source.
- **denflt.f** Computes ray density in geographic cells for fault source.
- **depvel.f** Converts bathymetry to velocity grid.
- **deriv.f** Calculates derivative of an array.
- **faultpol.f** Converts Okada fields to point sources.
- **faultseg.f** Creates segmented faults.
- **grdminmax.f** Finds min. & max. of a 3D grid.
- **grdsmooth.f** Smooths XYZ grids using a moving average window.
- **median.f** Finds the median of an array.
- **minmax1.f** Finds min. & max. of a 1D grid.
- **minmax2.f** Finds min. & max. of a 2D grid.
- **routines.f** Set of FORTRAN routines used in other codes.
- **slice.f** Slices an XYZ grid.
- **smylie.f** Calculates Okada field based on earthquake source geometry.
- **superdisloc.f** Superimposes Okada field on the bathymetry grid.
- **superellipse.f** Superimposes static landslide dipoles on the bathymetry grid.



- **tsubox.f** Ray-traces point sources with group velocity.
- **tsuboxp.f** Ray-traces point sources with phase velocity.
- **tsubox\_series.f** Ray-traces a series of point sources within finite sources.

## References

- Aki, K. & Richards, P. G., 2002. *Quantitative Seismology*, University Science Books.
- Crotwell, H. P., Owens, T. J., & Ritsema, J., 1999. The taup toolkit: Flexible seismic travel-time and ray-path utilities, *Seismological Research Letters*, **70**, 154–160.
- Dziewonski, A. M. & Anderson, D. L., 1981. Preliminary reference earth model, *Physics of the earth and planetary interiors*, **25**(4), 297–356.
- Dziewonski, A. M. & Steim, J. M., 1982. Dispersion and attenuation of mantle waves through waveform inversion, *Geophysical Journal International*, **70**(2), 503–527.
- Geller, R. J., 1976. Scaling relations for earthquake source parameters and magnitudes, *Bulletin of the Seismological Society of America*, **66**(5), 1501–1523.
- Jobert, N. & Jobert, G., 1983. An application of ray theory to the propagation of waves along a laterally heterogeneous spherical surface, *Geophysical Research Letters*, **10**(12), 1148–1151.
- Mansinha, L. & Smylie, D., 1971. The displacement fields of inclined faults, *Bulletin of the Seismological Society of America*, **61**(5), 1433–1440.
- Rabinovich, A. B., 1997. Spectral analysis of tsunami waves: Separation of source and topography effects, *Journal of Geophysical Research: Oceans*, **102**(C6), 12663–12676.
- Salaree, A. & Okal, E. A., 2015. Field survey and modelling of the Caspian Sea tsunami of 1990 June 20, *Geophys. J. Intl.*, **201**(2), 621–639.
- Salaree, A. & Okal, E. A., 2018. The “tsunami earthquake” of 13 April 1923 in Northern Kamchatka: Seismological and hydrodynamic investigations, *Pure Appl. Geophys.*, **175**(4), 1257–1285.
- Sandanbata, O., Watada, S., Satake, K., Fukao, Y., Sugioka, H., Ito, A., & Shiobara, H., in press. Ray tracing for dispersive tsunamis and source amplitude estimation based on greens law: Application to the 2015 volcanic tsunami earthquake near Torishima, south of Japan, *Pure and Applied Geophysics*, pp. 1–15.
- Satake, K., 1988. Effects of bathymetry on tsunami propagation: Application of ray tracing to tsunamis, *Pure and Applied Geophysics*, **126**(1), 27–36.
- Synolakis, C. E., Bardet, J.-P., Borrero, J. C., Davies, H. L., Okal, E. A., Silver, E. A., Sweet, S., & Tappin, D. R., 2002. The slump origin of the 1998 Papua New Guinea tsunami, **458**(2020), 763–789.

- Tajima, F. & Garmany, J. D., 1987. Surface wave amplitude anomalies: is reciprocity valid?, *Physics of the earth and planetary interiors*, **47**, 267–287.
- Titov, V., Kânoğlu, U., & Synolakis, C., 2016. Development of MOST for real-time tsunami forecasting, *Journal of Waterway, Port, Coast and Oceanic Engineering*, **142**, 03116004–1–03116004–16.
- Ward, S. N., 2001. Landslide tsunami, *Journal of Geophysical Research: Solid Earth*, **106**(B6), 11201–11215.
- Woods, M. T. & Okal, E. A., 1987. Effect of variable bathymetry on the amplitude of teleseismic tsunamis: A ray-tracing experiment, *Geophysical Research Letters*, **14**(7), 765–768.