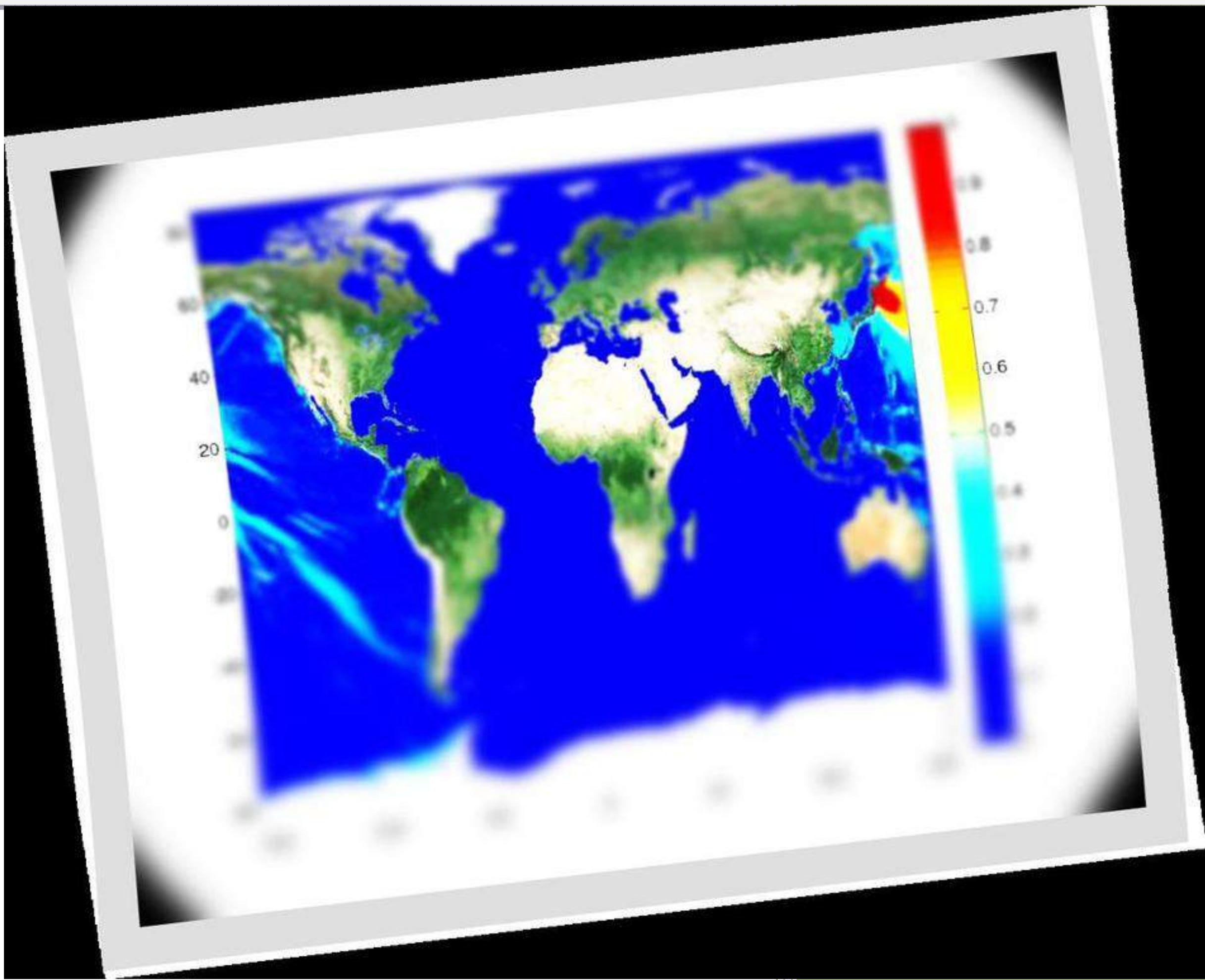
2011

TUNAMI FF User Manual

Numerical simulations of Far-Field tsunamis



LGPL v3 Free Software

EWSP-SIMTF

Version 2011

CUDA-SIMTF

11/11/2011

Preface

This manual of TUNAMI – FF is intended for usage of international users around the globe. The manual drafted[**] and prepared as part of support to Stand-by Inundation Model for Tsunami Forecast (SIMTF). The TUNAMI FF Codes developed as part of Contribution towards open source 23 April 2011. Dedicated to Victims of 2011 Great East Japan earthquake (東日本大震災 Higashi nihon daishinsai); Also this code made openly available to Standby Inundation Model for Tsunami forecast (SIMTF) - Early Warning System Project (EWSP) under GNU LGPL V 3.0 (LESSER GENERAL PUBLIC LICENSE Version 3, 29 June 2007)

License Notice

[**]This manual can be read-well with font face "Courier 10 Pitch"

Contents

1. Introduction

Numerical simulation of Far-Field tsunamis using TUNAMI-FF (***T**ohoku **U**niversity's **N**umerical **A**nalysis **M**odel for **I**nvestigation of **F**ar-**F**ield tsunamis*) was developed in framework of NVIDIA CUDA kernels. The numerical Model TUNAMI - FF is based on "tsunami numerical simulation with the staggered leap-frog scheme (Numerical code of TUNAMI-F1)" of Dr. Fumihiko Imamura, professor of tsunami engineering school of civil engineering, Asian inst. tech. and disaster control research center, Tohoku University prepared in June, 1995 for TIME (Tsunami Inundation Modeling Exchange) project which was a joint effort of IUGG and IOC/UNESCO . In 1997, the TIME project manual was published by UNESCO as IOC Manuals and Guides No.35 "*IUGG/IOC Time Project: Numerical method of tsunami simulation with the leap-frog scheme*".

TUNAMI-FF simulates all stages of a tsunami from the origin and the propagation in the ocean to the arrival at the coast and wave amplitudes at beach (~ 1 m water depth) by application of Greens' law. It solves the linear shallow water equations on a structured regular grid with given resolution representing a coarse grid in the deep ocean. The implementation started in 2010 and many scientists contributed substantially.

2. Background Physics and Numerical Implementation of Model

Tohoku University's Numerical Analysis Model for Investigation of Far-field Tsunamis - TUNAMI FF computes wave propagation in the linear approximation of the long-wave theory in spherical coordinates. The applied numerical scheme follows the well-known TUNAMI-F1 algorithm [TIME Project, 1997], and is leap-frog explicit time-stepping on a staggered finite difference grid. Boundary conditions presume full reflection along shorelines. The following are assumptions made during numerical implementation of model:

1. The astronomical tides do not vary with respect to time throughout the tsunami simulation. The Still water Level in the computation is set equal to the water level at the beginning of the simulation
2. Both temporal and spatial grid lengths vary only at the ratio of 1:3:9 and so on, if the change of them is necessary
3. In the linear computation, no run up can be included, and therefore the computation is not carried out for the water depth shallower than 0.1 cm, and vertical walls are set in place of the actual slope.

Numerical simulations of far-filed tsunamis, representing transoceanic propagation requires large area of computation. Such numerical simulations of far-filed tsunamis which travels more than 1000 km over ocean should be computed in polar-coordinates by considering earth as sphere of radius R, covered by the latitude and longitude (theta, lambda). Far-filed tsunami simulations covering wide areas of computation, in turn long travel distance may yield dispersion of wave components. Therefore in order to include physical dispersion term the equations of higher order approximation are used. But long travel time yields an inevitable accumulation of numerical error, for which the computation programme should be carefully designed.

In the method of simulation, the linear long wave theory is expressed in latitude-longitude coordinates with different formulation of equations. When the liner theory is used, it is very easy to attain a high rate of vectorization in terms of programming. The current TUNAMI FF program for transoceanic propagation is composed to fully utilize the vectorizaion of parallel programming. The rate of vectorization of higher than 99% is a result of elimination of both the IF-sentences in DO-groups and the division operation.

Solving the linear long wave equations is computationally very efficient, but, this approximation is no longer valid in coastal regions of water depths shallower less than ~30 m [Shuto, 1991], where the nonlinear bottom friction and advection terms play major role. In order to overcome this limitation of non-linearity and to get the estimated wave arrival as well wave amplitudes at beach (~ 1 m water depth) the approach of Green's Law adopted [Kamigaichi, 2009]. In this approach, Deep Ocean Tsunami Amplitudes (DOTA) are first computed at selected deep water positions (typically, in 30-50 m water depth) and then extrapolated to get the Wave Amplitudes At Beach (WAAB) using the Green's law. According to Kamigaichi [2009], estimated WAABs are agree well with those averaged tsunami amplitudes computed directly at beach locations with inundation grid of finer resolution. So the WAABs can be considered as 'reliable' estimates of tsunami impact on coastal locations without going for such complex inundation computations with higher resolution model setups.

Governing Equations:

In computation of a far-field tsunami, the dispersion term becomes important because the long travel distance acts to disperse wave components. The linear Boussinesq equation which includes the physical dispersion term is considered appropriate to express this effect. The physical dispersion term can be replaced by the numerical dispersion term using a simple method as explained in TUNAMI-F1 algorithm [TIME Project, 1997]), which is inevitably resulted as the truncation error of a numerical scheme. But this replacement of physical dispersion term is possible only when the grid length is appropriately selected. Then the liner long wave theory approximation of lower order becomes almost equivalent to the liner Boussinesq equation approximation of higher order. Also, with this replacement much CPU time and computer memory are saved in computations.

The liner long wave theory is given by the following expression in the latitude-longitude co-ordinates.

$$\frac{\partial \eta}{\partial t} + \frac{1}{R \cos \theta} \begin{bmatrix} \frac{\partial M}{\partial \lambda} + \frac{\partial}{\partial \theta}(N \cos \theta)) \end{bmatrix} = 0$$

$$\frac{\partial M}{\partial t} + \frac{gh}{R \cos \theta} \frac{\partial \eta}{\partial \lambda} = fN$$

$$\frac{\partial N}{\partial t} + \frac{gh}{R} \frac{\partial \eta}{\partial \theta} = -fM$$

Equation1: \frac{\partial \eta}{\partial t} + \frac{1}{R \cos \theta } \begin{bmatrix} \frac{\partial M}{\partial \lambda } + \frac{\partial}{\partial \theta}(N \cos \theta))\end{bmatrix} = 0

Equation2: \frac{\partial M}{\partial t} + \frac{gh}{R \cos \theta } \frac{\partial \eta}{\partial \lambda } = fN

Equation3: \frac{\partial N}{\partial t} + \frac{gh}{R} \frac{\partial \eta}{\partial \theta } = -fM

URL: https://www.codecogs.com/latex/eqneditor.php

Where η is the water level, M and N are discharge fluxes in the λ (along a parallel of latitude) and θ (along a circle of longitude) directions, g is the gravitational acceleration, and f ($2\omega \sin\theta$) is the Coriolis coefficient. It should be kept in mind that values of h take negative sign on land and positive sign in ocean.

The leap-frog scheme is applied to obtain difference expressions of above three governing equations, then we have

$$\frac{\eta_{j,m}^{n+1/2} - \eta_{j,m}^{n-1/2}}{\Delta t} + \frac{1}{R\cos\theta_m}\begin{bmatrix}\frac{M_{j+1/2,m}^{n} - M_{j-1/2,m}^{n}}{\Delta\lambda} + \frac{N_{j,m+1/2}^{n}\cos\theta_{m+1/2} - N_{j,m-1/2}^{n}\cos\theta_{m-1/2}}{\Delta\lambda}\end{bmatrix} = 0$$

$$\frac{M_{j+1/2,m}^{n} - M_{j-1/2,m}^{n}}{\Delta t} + \frac{gh_{j+1/2,m}}{R\cos\theta_m}\frac{\eta_{j+1,m}^{n+1/2} - \eta_{j,m-1/2}^{n+1/2}}{\Delta\lambda} = fN^1$$

$$\frac{N_{j+1/2,m}^{n+1} - N_{j+1/2,m}^{n}}{\Delta t} + \frac{gh_{j+1/2,m}}{R\sin\theta_m}\frac{\eta_{j,m+1}^{n+1/2} - \eta_{j,m}^{n+1/2}}{\Delta\theta} = fM^1$$

Equation4: \frac {\eta_{j,m}^{n+1/2} - \eta_{j,m}^{n-1/2}} {\Delta t} + \frac{1}{R \cos \theta_m } \begin{bmatrix} \frac {M_{j+1/2,m}^{n} - M_{j-1/2,m}^{n}} {\Delta \lambda } + \frac {N_{j,m+1/2}^{n} \cos \theta_{m+1/2} - N_{j,m-1/2}^{n} \cos \theta_{m-1/2}} {\Delta \lambda }\end{bmatrix} = 0

Equation 5: \frac {M_{j+1/2,m}^{n} - M_{j-1/2,m}^{n}} {\Delta t } + \frac {gh_{j+1/2,m}}{R \cos \theta_m} \frac { \eta_{j+1,m}^{n+1/2}- \eta_{j,m-1/2}^{n+1/2} } {\Delta \lambda } = fN^1

Equation 6: \frac {N_{j+1/2,m}^{n+1} - N_{j+1/2,m}^{n}} {\Delta t } + \frac {gh_{j+1/2,m}}{R \sin \theta_m} \frac { \eta_{j,m+1}^{n+1/2}- \eta_{j,m}^{n+1/2} } {\Delta \theta } = fM^1

URL: https://www.codecogs.com/latex/eqneditor.php

where M^1, N^1 and the unknowns - η, M and N are given by the equations no 7, 8, 9 and 10 as stated in TUNAMI-F1 algorithm [TIME Project, 1997]). The constant coefficients R1ns in equations 8, 9 and 10 as stated in TUNAMI-F1 algorithm [TIME Project, 1997]) are given as follows:

$$R1 = \Delta t/(R\cos\theta m\Delta s)$$

$$R2 = g\Delta t/(R\cos\theta m\Delta s)$$

$$R3 = 2\Delta t\omega\sin\theta m$$

$$R4 = g\Delta t/(R\Delta s)$$

$$R5 = 2\Delta t\omega\sin\theta m+1/2 .$$

A point of computation is numbered as (j, m, n) in the (θ, λ, t) directions. The grid lengths are($\Delta\theta$, $\Delta\lambda$, Δt). In the present computation grid lengths in the latitude and longitude directions are taken equal; i.e., $\Delta s = \Delta\theta = \Delta\lambda$ .The angular velocity of earth rotation is given by ω.

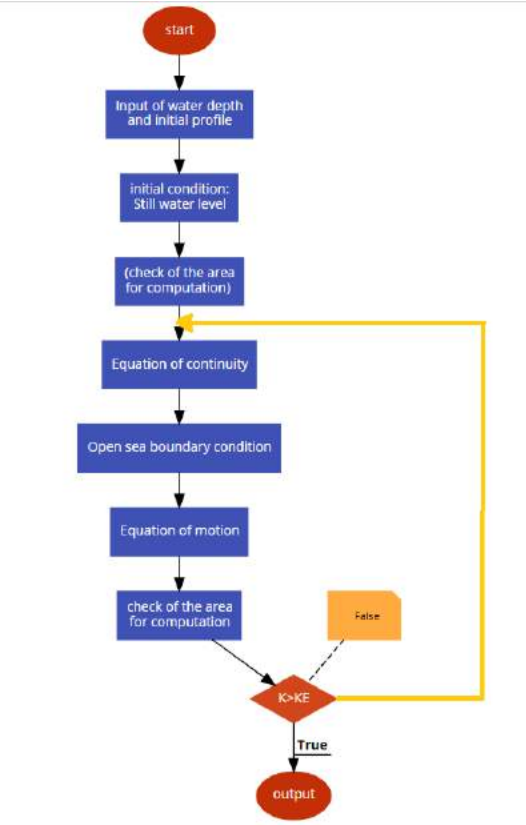
The spatial grid length is taken to be , and the time step interval is taken to be 20 seconds. This spatial grid length is selected to satisfy the condition that the Imamura number Im given below is nearly equal to unity, or in other words the numerical and physical dispersion effects are nearly the same.

$$Im= \Delta x/2h\sqrt{(1-k^2)}$$

where K is the Courant number (= $(gh)^{1/2}$ $\Delta t/\Delta x$) and h is the mean water depth in the ocean under consideration. The time step interval is determined to satisfy Courant–Friedrichs–Lewy (CFL) condition for the spatial grid length thus determined.

Flow of TUNAMI F1 simulation main program

- Input of Water Depth and Initial profile
- Initial condition: Still water level
- Check of the area of computation
- Equation of continuity
- Open Sea boundary condition
- Equation of Motion
- Check of the area of computation
- K>KE
- Output



<u>Variables and constants in TUNAMI F1 program</u>

Variables:

- Water level **Z**
- Discharge flux **M, N**
- Still water depth **H**
- Time history of water level **PZ**
- Co-ordinates of points for output of the history of water level **IP, JP**
- Working arrays for vector operations **V1, V2, V3, V4, V5, V6 and V7**

Coefficients:

■ Highest water level **ZM**
■ Lowest water level **ZN**
■ Coefficients given **R1, R2, R3, R4, R5, R6** AND **R6**=COS (THETA M+1/2)
■ (THETA M+1/2)in radian **C1**
■ (THETA M)in radian **C2**
■ (THETA M-1/2)in radian **C3**
■ Water depth: h **C4**

Constants:

■ Gravitational acceleration **GG**
■ Circular constant **pi** (=**3.1415926**)
■ Radius of the earth **R**

Computation is controlled by following conditions

■ Size of the area for computation in longitude and latitude **IG, JG**
■ Latitude of the southernmost end of the area for computation **FL**
■ Area where the tsunami exists and the computation is carried out **IS, JS, IE, JE**
■ Grid length in minute, and time step length in second **DS, DT**
■ Time steps of beginning and end of computation **KS, KE**
■ Number of spatial points where the time history of water level output **NG**
■ Time step length in outputting the time history of water level **KC**
■ Time step length to output spatial wave profiles **KD**

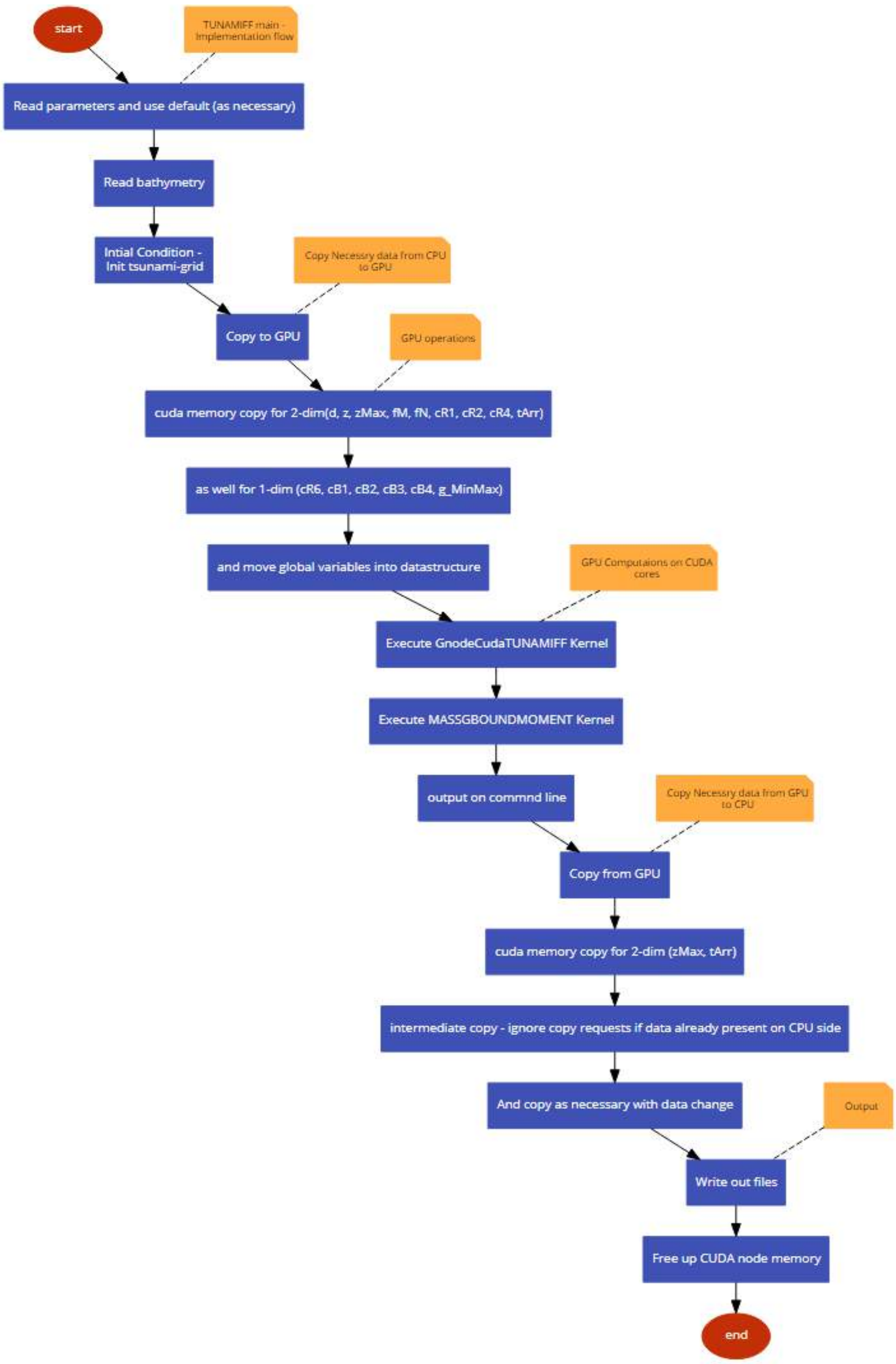
<u>SUBROUTINES (TUNAMI F1) Mapped to CUDA Kernels# (TUNAMI FF)</u>

- Data input of water depth **RDEPTH**
- Setting of parameters required in vectorized computation **PARAME**
- Input of the initial condition and the initial profile **INITIA**
- Making area of computation be within the area under consideration **ALIMIT**
- Enlargement of the area of computation as the tsunami propagates **BLIMINT**
- Output and display of the spatial distribution of water level at an instant **OUTPUT**
- Time form the beginning of the computation, special for a NEC SX-1 **CLOCK**
- Computation of the equation of continuity **MASS**
- Open sea, boundary condition **GBOUND**
- Computation of the equation of motion **MOMENT**
- Check of the highest and lowest water level **MAX**
- Output of the time history of water level at the point (IP, JP) **POINT**

- Output of the tsunami arrival time in hour **PROPA**
- Output of the highest and lowest water level, and the arrival time **OUTDT**
- Output of the water level and the discharge flux **FILEOT**

TUNAMI FF CUDA Kernels Mapping

All subroutines of well-known TUNAMI-F1 algorithm [TIME Project, 1997] are mapped well to CUDA kernels and can be found in the ***Appendix -A*** of this user manual.



//TUNAMI FF main - Implementation flow

start;
Read parameters and use default (as necessary);
Read bathymetry;
Intial Condition -
Init tsunami-grid;
//Copy Necessry data from CPU to GPU
Copy to GPU;
// GPU operations
cuda memory copy for 2-dim(d, z, zMax, fM, fN, cR1, cR2, cR4, tArr);
as well for 1-dim (cR6, cB1, cB2, cB3, cB4, g_MinMax);
and move global variables into datastructure;
//GPU Computaions on CUDA cores
{
Execute GnodeCudaTUNAMIFF Kernel;
Execute MASSGBOUNDMOMENT Kernel;
output on commnd line;
}
//Copy Necessry data from GPU to CPU
Copy from GPU;
cuda memory copy for 2-dim (zMax, tArr);
intermediate copy - ignore copy requests if data already present on CPU side;
And copy as necessary with data change;
// Output
Write out files;
Free up CUDA node memory ;
end;

TUNAMI FF main - Program flow (cuda kernels)

Check of the area of computation
Computation of the equation of continuity (iZ, iM, iN, iR1, iR2)
sea floor topography (mass conservation)
Open sea, boundary condition, open boundaries (iZ, iM, iN, C1, C2, C3, C4)
Computation of the equation of motion (moment conservation)
Equation of Motion (longitudial flux update - moment conservation) (v1, v2, iZ, iM, iN, iR2, iR3)
Open Sea boundary (open bondaries) (iZ, iM, iR2)
Equation of Motion (lattitudial flux update -moment conservation) (v1, v2, iZ, iM, iN, iR4, iR5)
Open Sea boundary (open bondaries) (iZ, iM, iN, iR4)
Check of the area of computation (calculation area for the next step)
Making area of computation be with in the area under consideration and
Enlargement of the area of computation as the tsunami propagates

CUDA Kernels:

MASS Kernel(KernelData data) {
//Computation of the equation of continuity MASS

VARIABLES
Water Level z
Discharge In I-direction fM
Discharge In J-direction fN
Area where the tsunami exists iMin, jMin, iMax, jMax
and the computation is carried out
Highest water level zMax
zoutZT

COEFFICIENTS:
Coefficients given cR1 and cR6=COS (THETA fM+1/2)

__MOMENT Kernel__(KernelData data) {
//Computation of the equation of motion MOMENT
VARIABLES
Water Level z
Discharge In I-direction fM
Discharge In J-direction fN
Area where the tsunami exists iMin, jMin, iMax, jMax
and the computation is carried out

COEFFICIENTS:
Coefficients given cR2 and cR4

__GBOUND Kernel__(KernelData data) {
//Open sea, boundary condition GBOUND
VARIABLES
Water Level z
Discharge In I-direction fM
Discharge In J-direction fN

COEFFICIENTS:
Coefficients given
(THETA M+1/2)in radian = cB1
(THETA M)in radian = cB2
(THETA M-1/2)in radian = cB3
coefficent = cB4

__MOMENTBOUND Kernel__(KernelData data) {
VARIABLES
Water Level z
Discharge In I-direction fM
Discharge In J-direction fN

COEFFICIENTS:

Coefficients given cR2 and cR4

<u>AlimitBlimitGrid Kernel</u>(KernelData data) {

//Making area of computation be with in the area under consideration as wellas

//Enlargement of the area of computation as the tsunami propagates

VARIABLES

Water Level z

Area where the tsunami exists iMin, jMin, iMax, jMax

and the computation is carried out

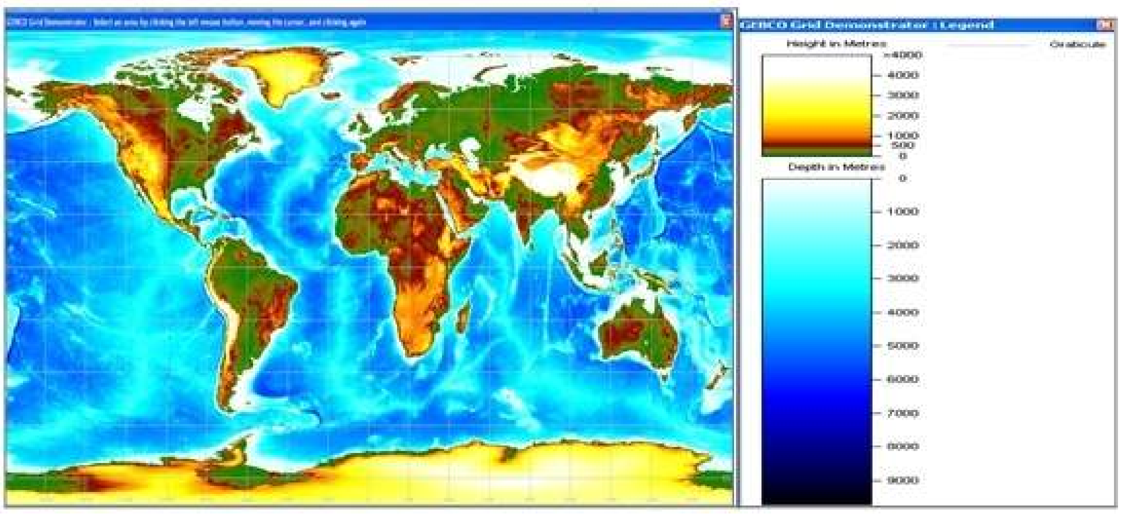
zoutCT, x, y, w

3. Data Preparation

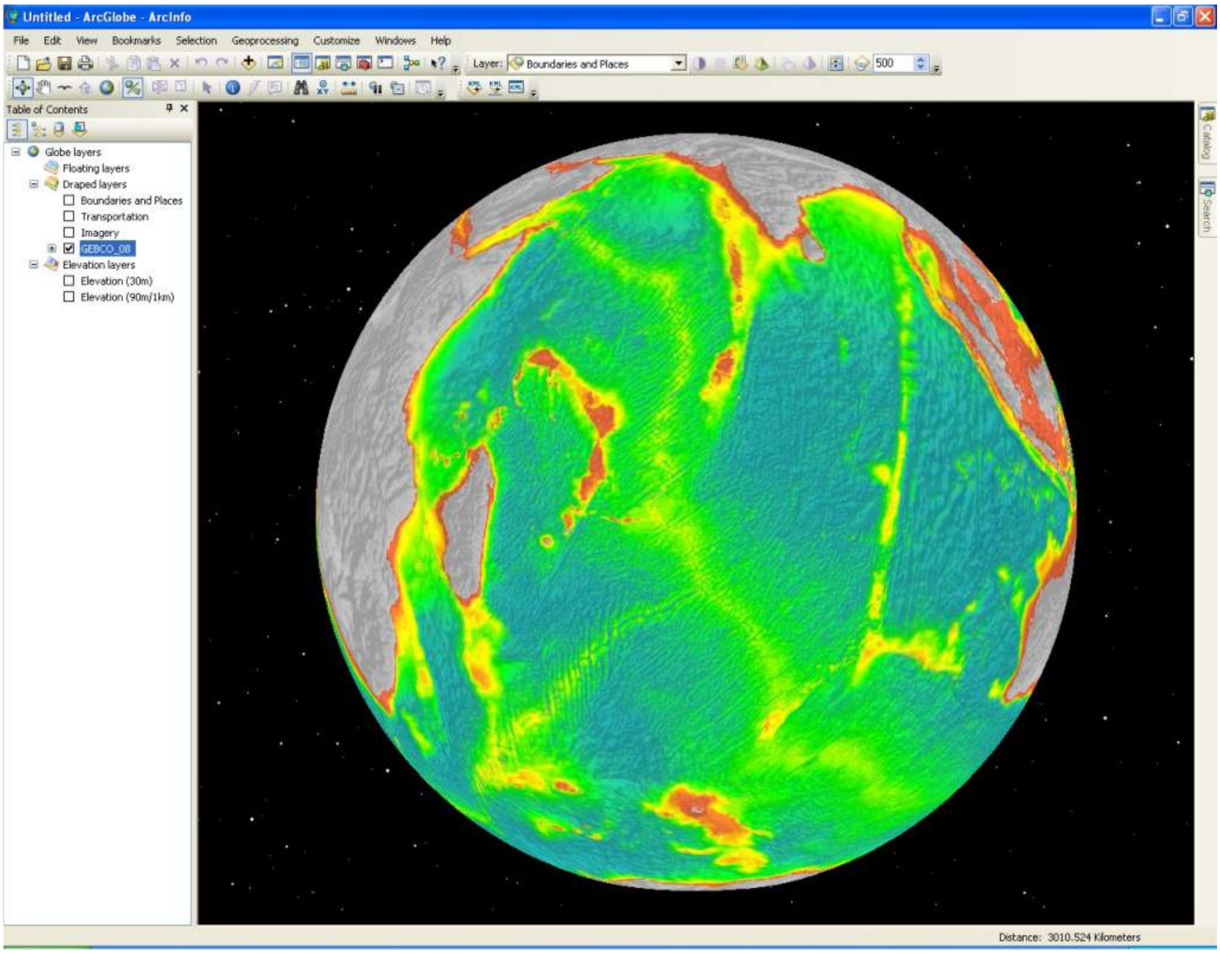
TUNAMI FF program uses the bathymetry of the area as input data. The bathymetry of the area is usually stored as netcdf format (.nc), GMT grid format (.grd) and data files (.asc). For TUNAMI FF program one can use the data from open sources viz., GEBCO, Etopo,..etc. Here in this user manual one example of GEBCO data preparation is given.

GEBCO, the General Bathymetric Chart of the Oceans, is a joint project of the International Hydrographic Organization (IHO) and the Intergovernmental Oceanographic Commission (IOC) bringing together an international group of experts whose aim is to provide the most authoritative publicly available bathymetry of the world's oceans, for scientific and educational use. One of GEBCO's products is a global gridded bathymetry data set. This grid contains depth (and, on land, elevation) estimates on a 30 arc-second mesh, which near the equator is about a one-kilometer spacing. Over the oceans, seafloor depths are from ship soundings, and where there are no soundings depths are interpolated, using estimates from satellite altimetry as a guide. The global bathymetry grid is periodically updated by incorporating additional bathymetric survey data, new compilations, and improved data. The grid data and grid display software can be accessed with user registration and downloaded from GEBCO Website

Grid display software - is free software available to view and access data from GEBCO's gridded bathymetric data sets. It provides the means for displaying the data and accessing the data in netCDF and simple ASCII formats. Please note that the software is designed for use with the complete, global grid files. It will not work with subsets of the full global data sets. The software has been developed to run on a PC running Microsoft Windows 95 or later. It is controlled by a series of drop-down menus and toolbar buttons. Version 2.13 of the software was released in April 2010. It includes the option to export the gridded data in ASCII form for conversion to an ESRI raster file.



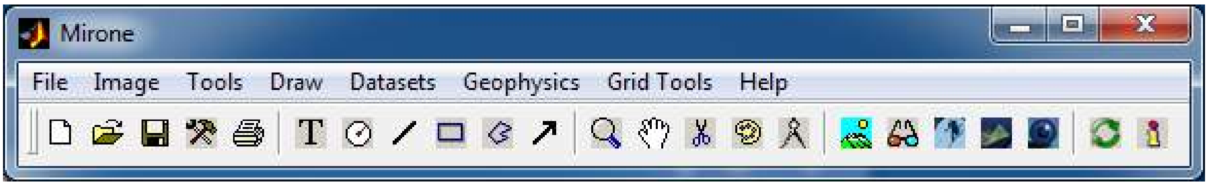
The ESRI raster file generated from ASCII form can viewed in ArcGlobe as shown in following figure.



Displaying GEBCO data in ArcGlobe

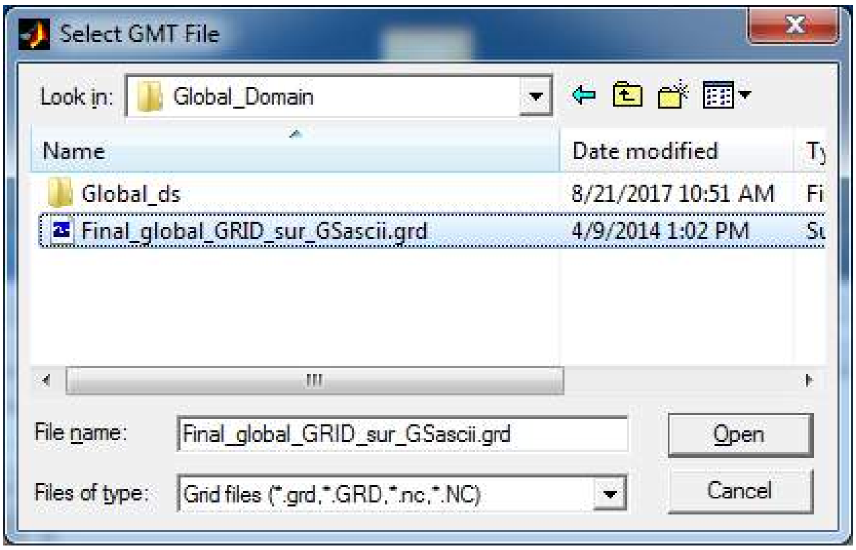
The exported netcdf gridded data from Grid display software can also be viewed in Mirone software for further processing and re-sampling, cropping..., etc grid operations for selection of area of interest (AOI).

Mirone is a MATLAB-based framework tool that allows the display and manipulation of a large number of grid formats through its interface with the GDAL library. Its main purpose is to provide users with an unusual easy-to-use graphical interface to the more commonly used programs of the GMT package. In addition it offers also a large number of tools that are particularly focused to the fields of geophysics and Earth Sciences. Among them, the user can find tools to do multibeam mission planning, elastic deformation studies, tsunami propagation modeling, IGRF computations and magnetic Parker inversions, Euler rotations and Euler poles computations, plate tectonic reconstructions, seismicity analysis and focal mechanism plotting, advanced image processing tools, etc... The high quality mapping and cartographic capabilities that made GMT known worldwide is guaranteed trough the Mirone's capability of automatically generating GMT cshell scripts and dos batch files.

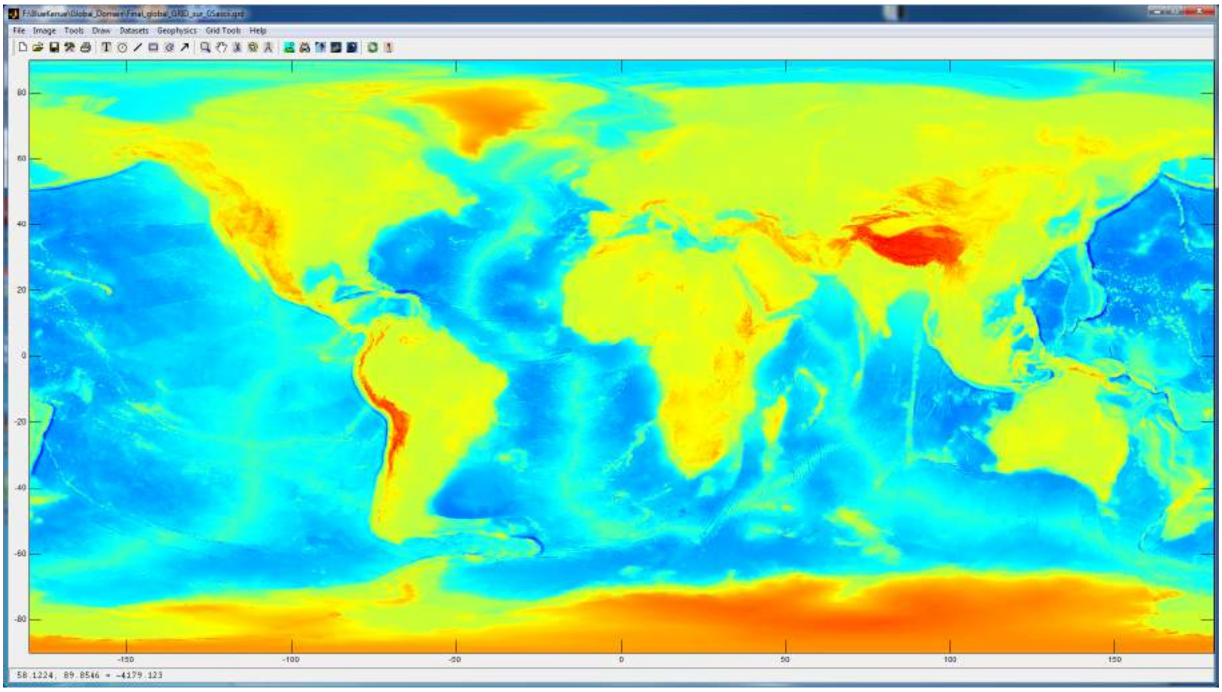


Although Mirone is written in MATLAB, a stand-alone version to run under Windows is also provided. This version was originally a bit less efficient than the native MATLAB code but since it is compiled the Intel compiler it became actually sensibly faster than the MATLAB version (not so much wonder, since the later gets permanently fatter with Java shit). The compiled version has its own advantages. Namely, we can drag-and-drop files onto the Mirone desktop icon to open them. In addition, it even supports file associations but in order that this works the user must set the environmental variable MIRONE_HOME with the path to the root Mirone's installation directory (this should be taken care of by the installer).

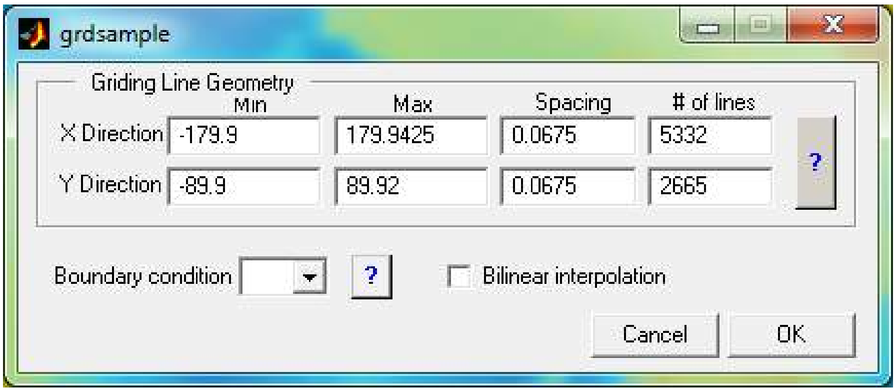
One can load the grid files of different formats (.nc, .NC, .grd, .GRD) netcdf, GMT, Surfer in Mirone tool for further processing with grid tools menu. Even larger size grid data files also can be loaded by setting up Grid Max size in *preferences* of Mirone tool depends on computer's memory appropriately.



Grid Max size - In order to do all grid manipulations Mirone stores the original grid in the computer's memory (regardless of their type they are always stored as single precision). This is the maximum size in Mb that a grid can have and be held on the computer's memory. This does not mean that larger grids/images cannot be processed. It only means that if they are larger than this value, they will be re-read from disk whenever necessity demands. It is up to you (based of course on the computer capabilities) to decide on a reasonable value for this parameter. Grid size is computed in the following way: n_rows x n_columns x 4 / (1024*1024). Note: Mirone does most of its heavy computation with external MEX files that use single precision arithmetic's. However, some operations may still be done with MATLAB operators which imply double precision (in this case it takes twice as much memory). When you are using image files, however, this rule does not apply anymore. In fact there is no obvious rule because you may be using compressed or uncompressed images.



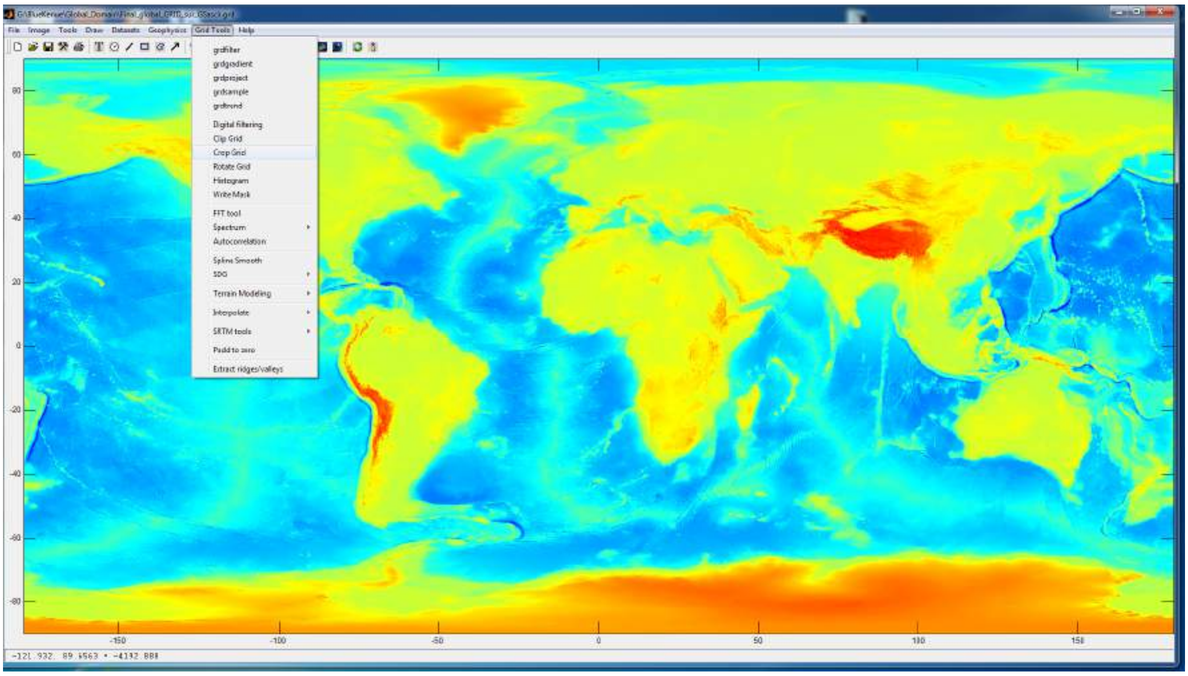
One can view the grid line gemotery of loadede grid data from Grid Tools menu >grdsample



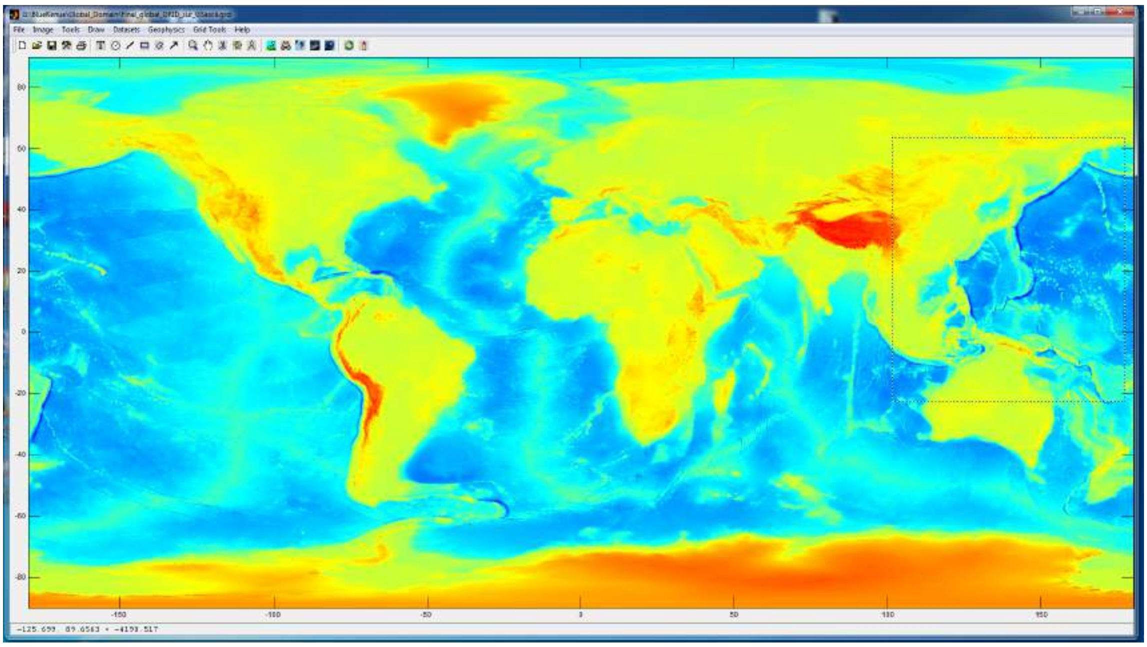
Grid Line Geometry

Grid line geometry defines the grid limits and grid density. Grid limits are the minimum and maximum X and Y coordinates for the grid. Grid density is usually defined by the number of columns and rows of the grid. The # of Lines in the X direction is the number of columns, and the # of Lines in the Y Direction is the number of grid rows. By defining the grid limits and the number of rows and columns as follows, the Spacing values are automatically determined as the distance between adjacent rows and adjacent columns.

	Min	Max	Spacing	No of Lines
X-direction:	-179.9000	179.9425	0.0675	5332
Y-direction:	-89.9000	89.9200	0.0675	2665



One can select area of interest (AOI) either by re-sampling *Grid tools >gridsample* or by cropping *Grid Tools>Crop Grid*

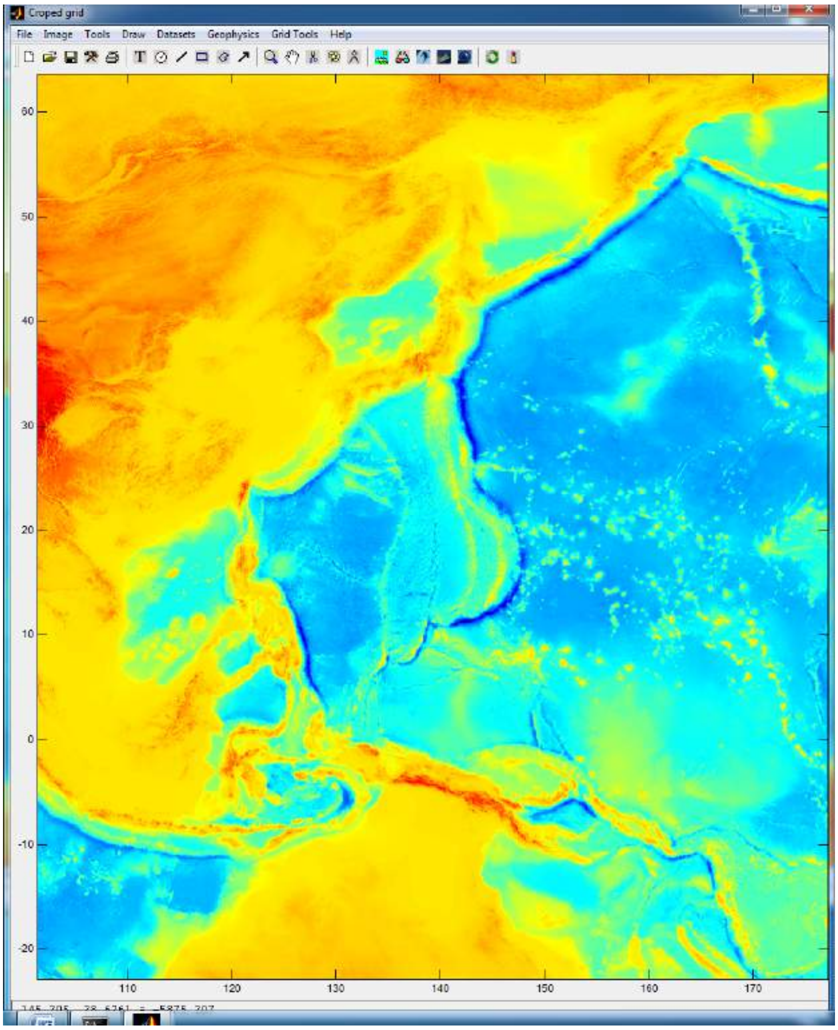


Cropping the AOI

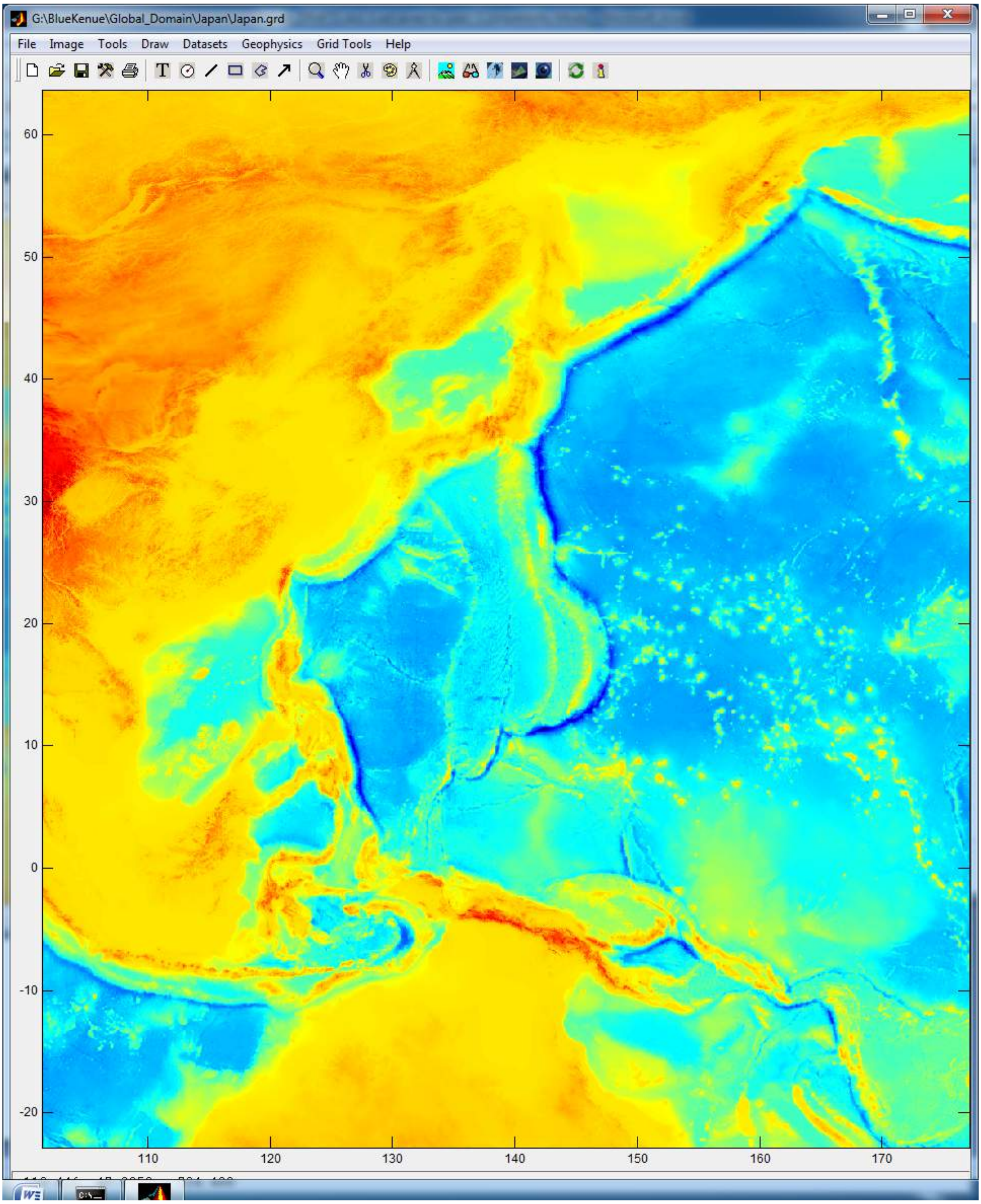
Crops an data grid inside a specified rectangle. After selecting this option the pointer changes into a cross indicating that you are in the drawing mode. Press the left mouse button and drag to define the crop rectangle. Finish by pressing the left mouse button again. The rectangle thus defined is used to extract a subset of the original grid. This corresponds to an interactive version of the grdcut GMT program with the advantage that you will not fall onto the grips of the "Ivan the Terrible" (I mean the despairing message: GMT ERROR: (x_max-x_min) must equal (NX + eps) * x_inc), where NX is an integer and |eps| <= 0.0001). Note, this is a crude and poorly controlled way of doing a cropping operation. The best way is to draw a rectangle on the image. Calmly decide on its size and position (or conversely, on its limits) and than inquire its properties (by a right-click on the rectangle borders) for “Crop Tools -> Crop Grid”.

a. Selection of Area of Interest (AOI):

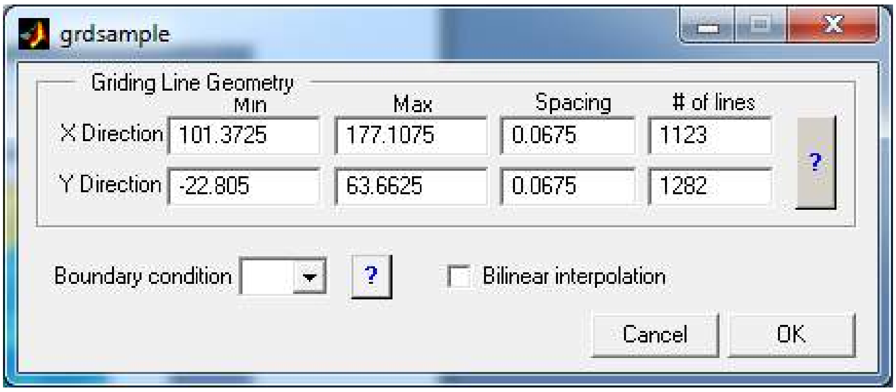
The AOI - area of interest can be cropped and exported as surfer grid bin format (Surfer 6 grid – Golden software grid binary format (ancient Surfer 6 format)). Either way by cropping the AOI or re-sampling through gridsample the exported grid shows following grid geometry with the # of



with the # of Lines in the X direction - number of columns, and the # of Lines in the Y Direction - the number of grid rows with given spacing of re-sampling.



In this example, the Girding Line Geometry is given as



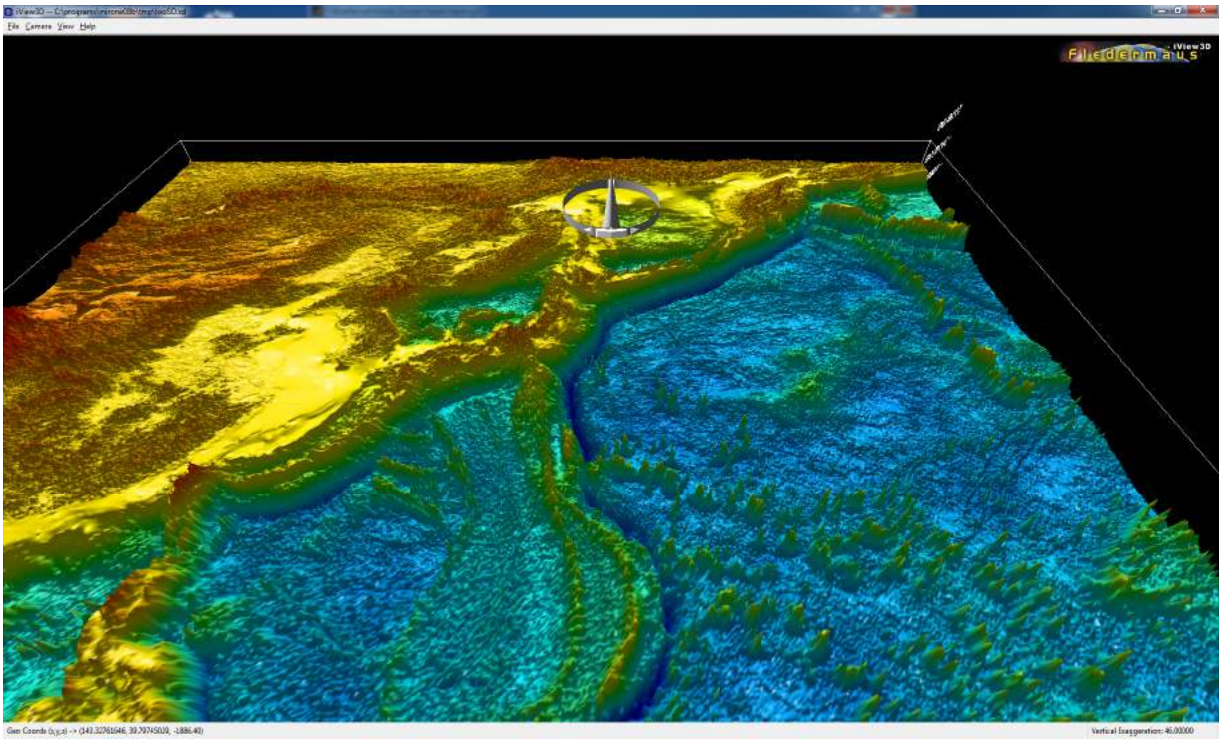
	Min	Max	Spacing	No of Lines
X-direction:	101.3725	177.1075	0.0675	1123
Y-direction:	-22.8050	63.6625	0.0675	1282

Grid sample Interpolates a grid to create a new grid with either: a new grid-spacing, or number of nodes, and perhaps also a new sub-region. Interpolation is bicubic [Default] or bilinear and uses boundary conditions. grdsample safely creates a fine mesh from a coarse one; the converse may suffer aliasing unless the data are filtered using grdfilter.

Finally save grid file Save as surfer grid bin format (Surfer 6 grid – Golden software grid binary format (ancient Surfer 6 format)) with naming convention " ***Ogrid.grd*** "

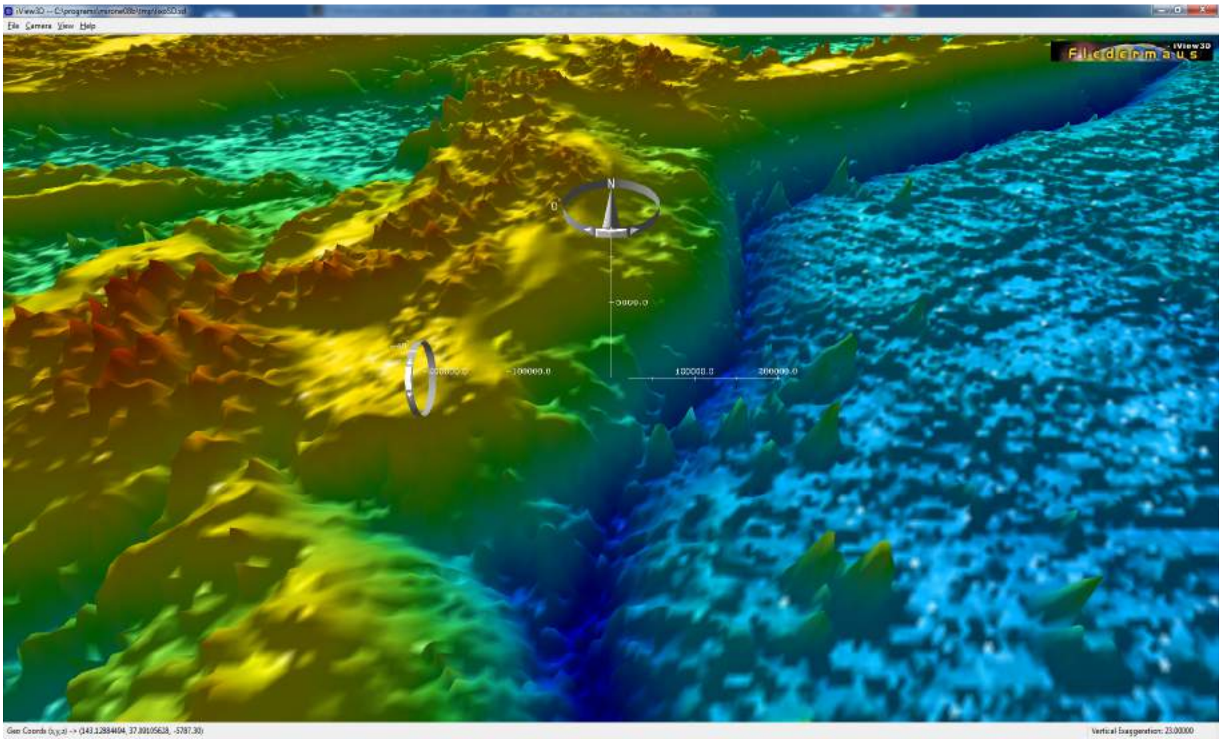
b. Visualization in 3- Dimensional view:

Grids (with whatever they contain) may be saved as .sd IVS objects. These objects may be



viewed with the iview3d free Fledermaus viewer. Fledermaus is a fantastic 3D software (probably the best) and has a free viewer that can be downloaded from www.ivs.unb.ca/products/iview3d/. If you have it installed you can either use this option to create a .sd file and separately run it with the viewer or click the icon "eye" in the Tool bar which will do both (but not save the .sd file). Note that large grids produce large .sd files so if you do not have that much RAM (or you want to save the .sd for later use) it may be better to use this save option.

There are two types of Fleder files and their labels are quite obvious for what they mean: Spherical Fledermaus Obj and Planar Fledermaus Obj



4. Initial conditions:

TUNAMI FF does not use the fault data as it is. The present program is only for tsunamis. No wind waves and tides are included. The still water level is given by tides and is assumed constant during tsunamis are computed. Through mirone one can generate deformation by importing or drawing fault trace data and used to produce the initial wave - deformation by the fault. TUNAMI FF uses this initial wave data to start with the modeling.

Bottom Deformation due to the fault motion: The fault break parameters are

i) The location of the fault coordinates as starting and ending points consequently (length of the fault)
ii) The width of the fault
iii) The direction of the fault axis from North (Clockwise)
iv) The dip angle
v) The slip angle
vi) Vertical displacement of the fault
vii) Focal depth

The data concerning the fault consists of the following parameters. The coordinates of the starting point and the end point of the fault, the width of the fault in kilometers, the dip direction, dip angle, slip angle in degrees, the dislocation and the depth in meters.

To compute elastic deformation, here you can either import a line (or polyline) or draw it over your image grid. That line is than interpreted as a surface trace of a fault. The following is sample format of line file " fault_trace.dat " representing three fault segments:

Start point Segment1 --->	141.559639	34.892694	
Start point Segment2 --->	142.967992	36.882004	***<-- End point Segment1***
Start point Segment3 --->	143.630746	39.119978	***<-- End point Segment2***
	143.630746	41.026401	***<-- End point Segment3***

Right clicking on the line will show (among others) two options to call new windows where you will specify the fault parameters (width, depth, dip) as well as the movement you will simulate. The two available options are called ***Okada*** and ***Mansinha***. The Okada window is more general and oddly nearly twice fast and here you can: model co-seismic displacement - that can be compared with GPS measuring - for an initial condition of the far field tsunami modeling (see below) or to compute synthetic interferograms. The Okada window provides short help in the form of tooltips when the mouse rests a bit over individual entry boxes. For a complete reference you must read the RNGCHN paper (Feigl and Dupre 1999; Computers and Geosciences, 25 (6) pp. 695-704). The paper and the source code used to be available on authors ftp site but it seems to have vanished.

The Mansinha option computes only the vertical component of the deformation produced by an earthquake. The two codes should produce equal results (and they very closely do). On both Okada and Mansinha, a clever guess is made about the type of the grid coordinates. Accepted types are geographical, meters and kilometers. While the guessing works well most of times it may also fail, so please pay attention to the guessed type and corrected if it is not correct. When working with geographical coordinates all distances are computed by previously projecting the geographical coordinates to Transverse Mercator with origin in fault's first point.

It is possible to import variable slip models using the SUBFAULT format (well, more like what I could guess that format is since I didn't find any formal format description) or the "Finite-Source Rupture Model Database" SRCMOD (.fsp) format maintained by Martin Mai at www.seismo.ethz.ch/srcmod For that, select "Import Model Slip" and select a file from the window below. As options show, you can compute the cumulated deformation of all individual patches. Off course that it may take some time if you are computing over a large grid.

Parameters to compute *Vertical elastic deformation* are listed below

Length : fault length (km) *- by default computed using Segment starting and Ending points*

Width : fault width (km)

Strike : dip direction (degrees) - *by default computed using Segment starting and Ending points*

Dip : dip angle (degrees)

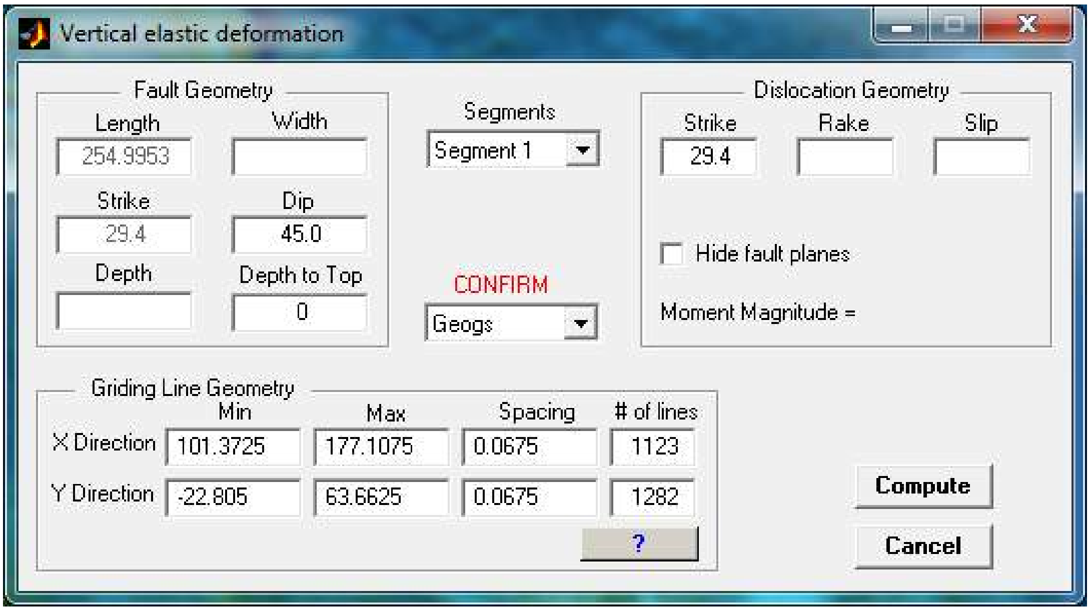
Rake : slip angle (degrees)

Slip : dislocation (m)

Depth : depth (km)

a. *Setting up of source parameters:*

After importing / drawing ploylline of fault trace " fault_trace.dat " representing three fault segments, the source parameters can be set for three segments namely Segment 1, Segment2 and Segment 3 with the following example parameters as shown in below table.

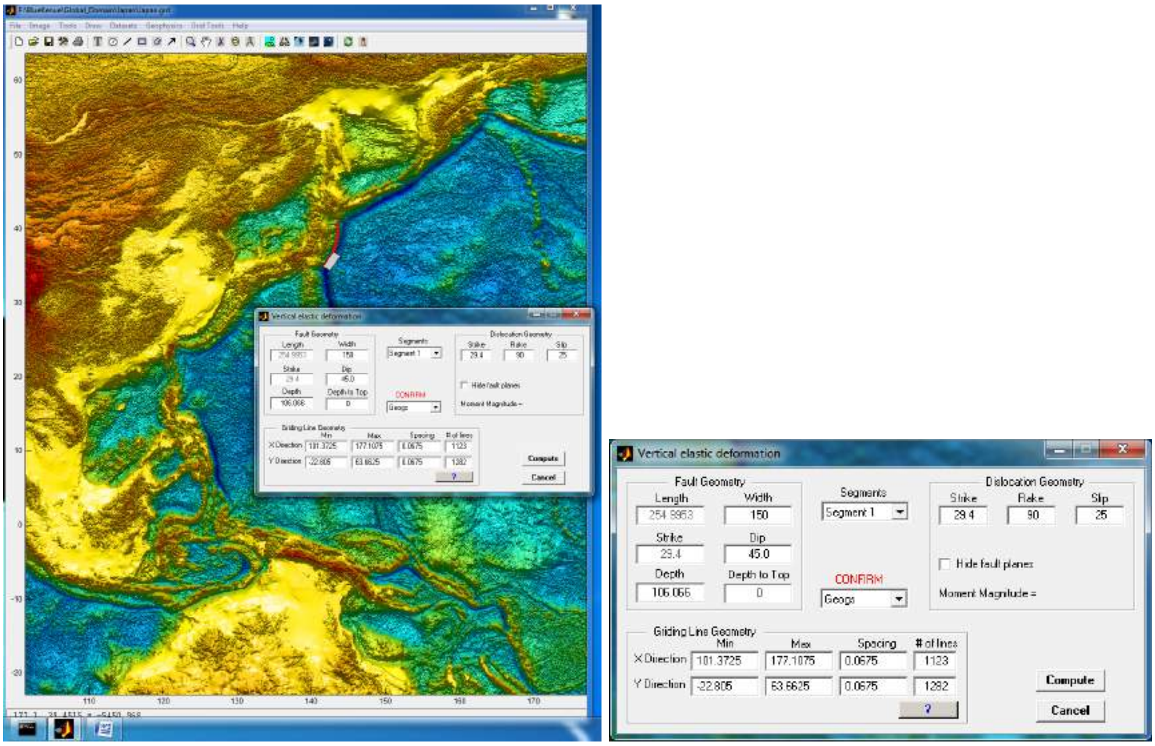


The above figure shows an example of the “Mansinha” use to compute Vertical elastic deformation. Below three figures for Segment 1, 2 and 3 show the one more of the Mirone’s fine details. The horizontal projection of the fault(s) plane(s) are displayed on the main window. Also, if you have the Fledermaus viewer installed you can have a 3D visualization of the fault plane. For time being the fault’s top is placed at “Depth to Top” below the grid’s minimum and not at the fault length’s average depth + “Depth to Top” as it would be correct.

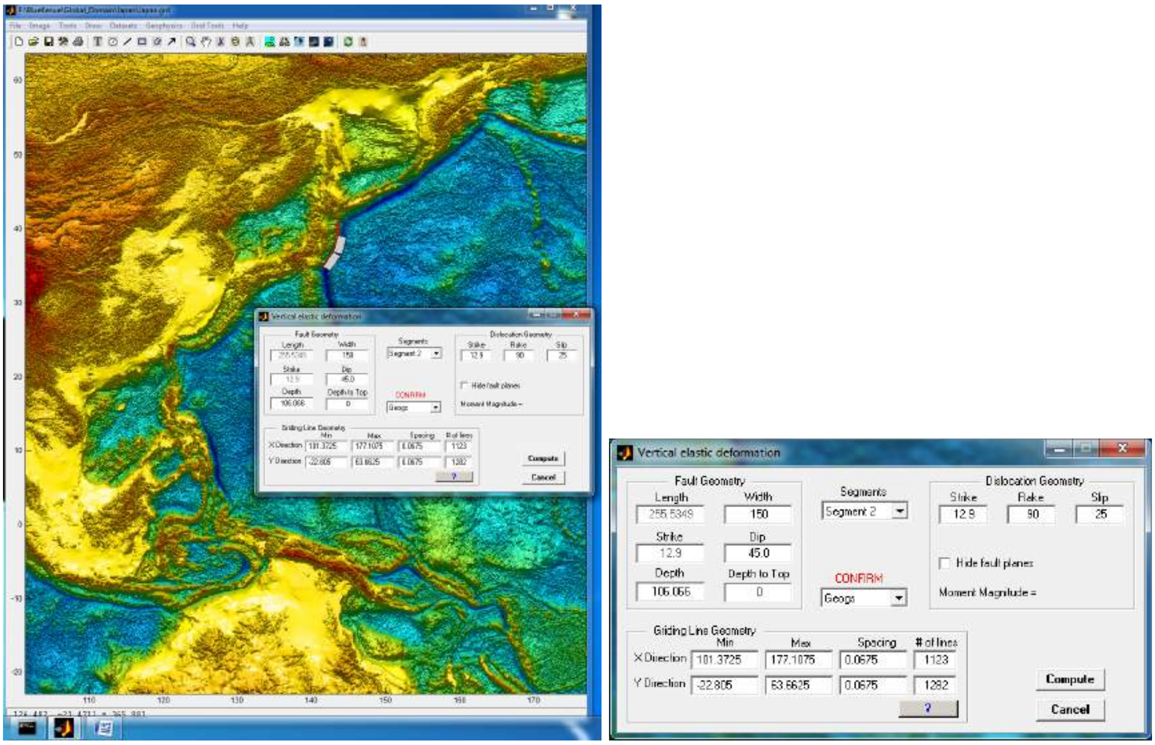
S No	Segments	Length(km)	Width(km)	Depth(km)	Dip (deg)	Strike (deg)	Rake (deg)	Slip (m)
1	Segment 1	254.9953	150	106.066	45	29.4	90	25
2	Segment 2	255.5349	150	106.066	45	12.9	90	25
3	Segment 3	211.9845	150	106.066	45	0.0	90	25

Table with example parameters for three segments for an earthquake of Moment Magnitude 9.2

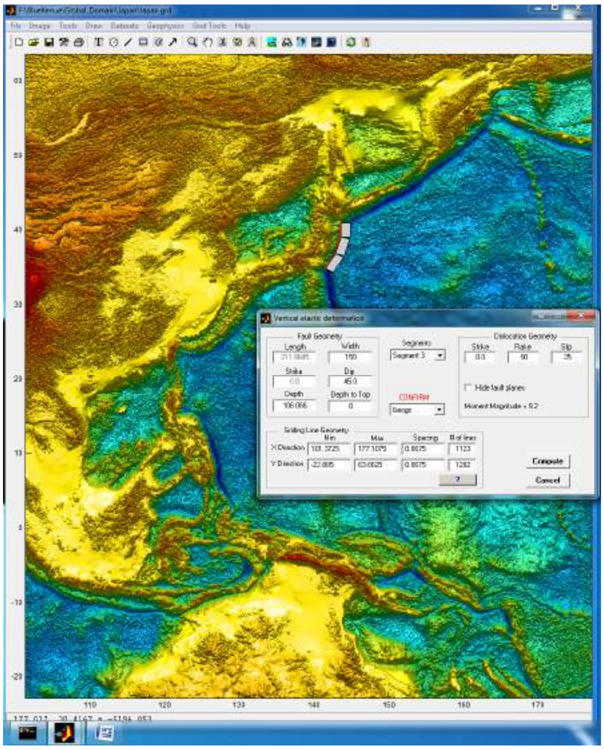
Segment No: 1 - Entry of source parameters

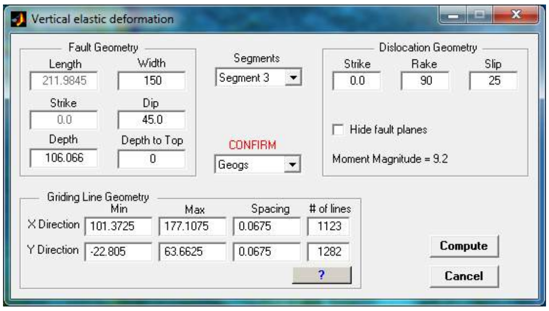


Segment No: 2 - Entry of source parameters



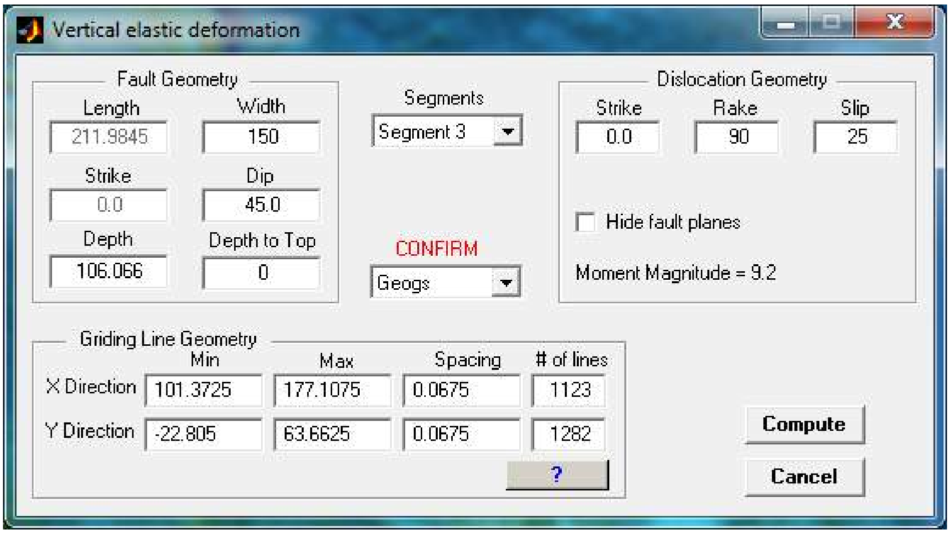
Segment No: 3 - Entry of source parameters



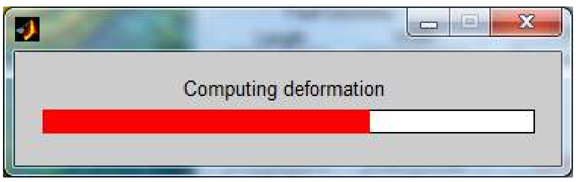


b. *Computing Deformation:*

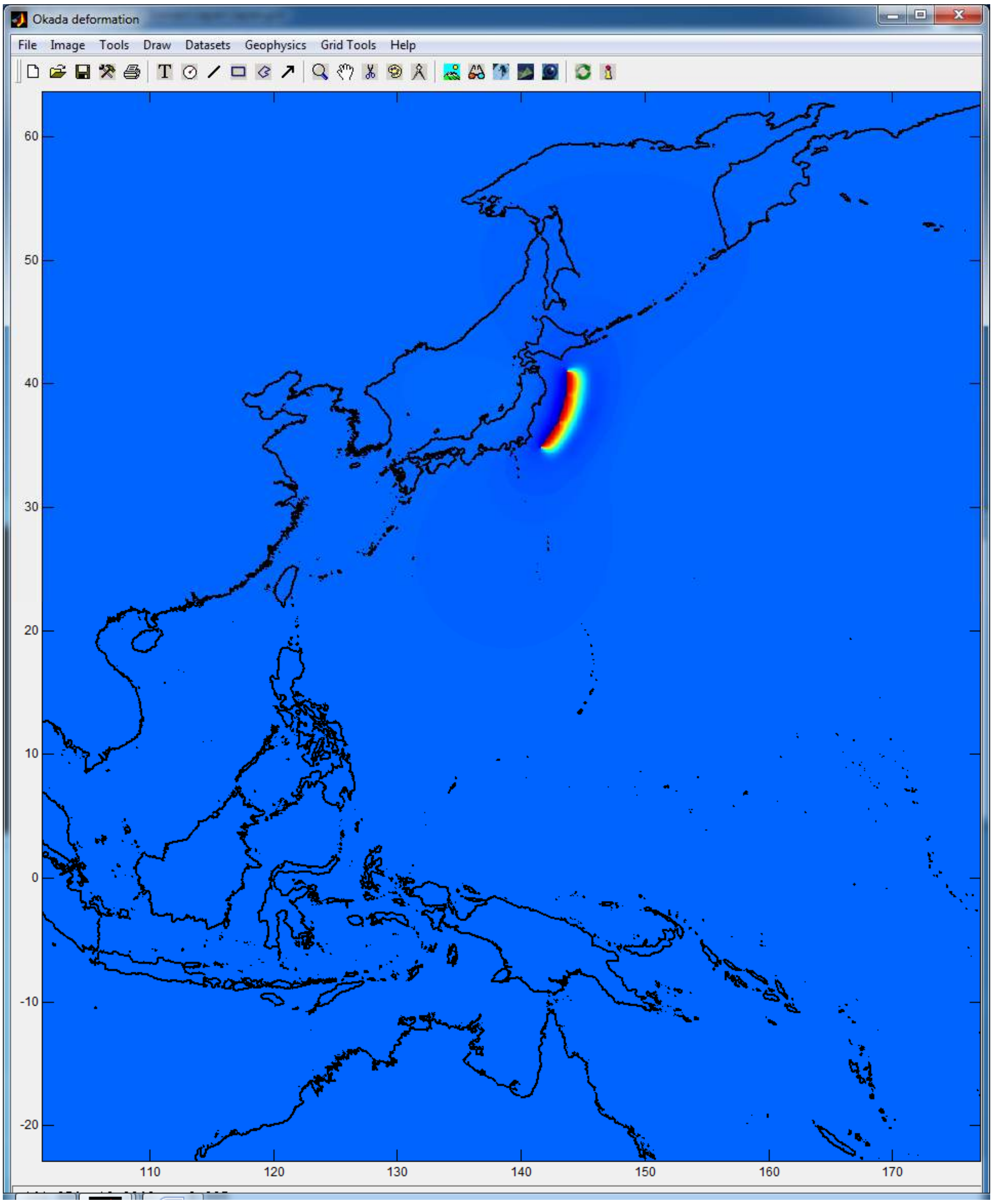
Multi segment Deformation can be computed by proceeding with the Compute option after correct entries of source parameters of all three segments



Then a dialog box appears stating "*Computing deformation*"

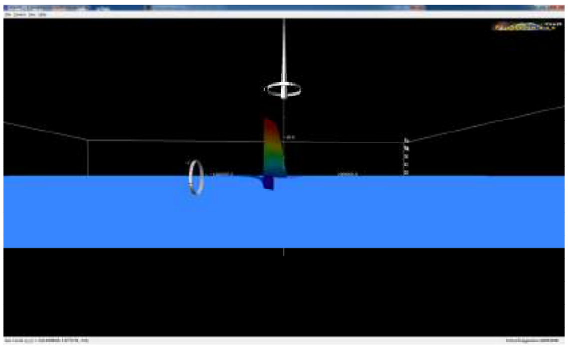
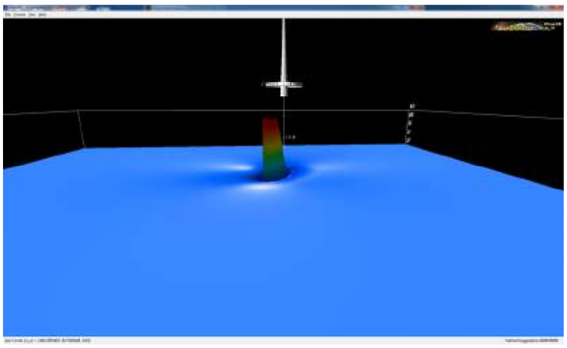
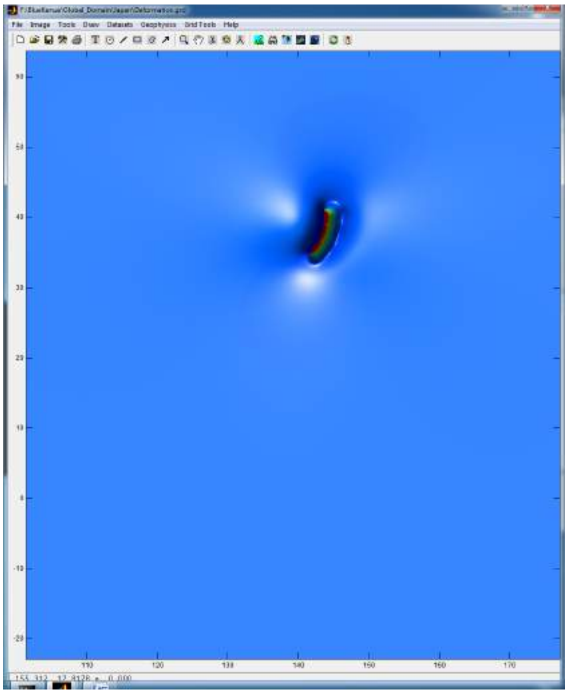
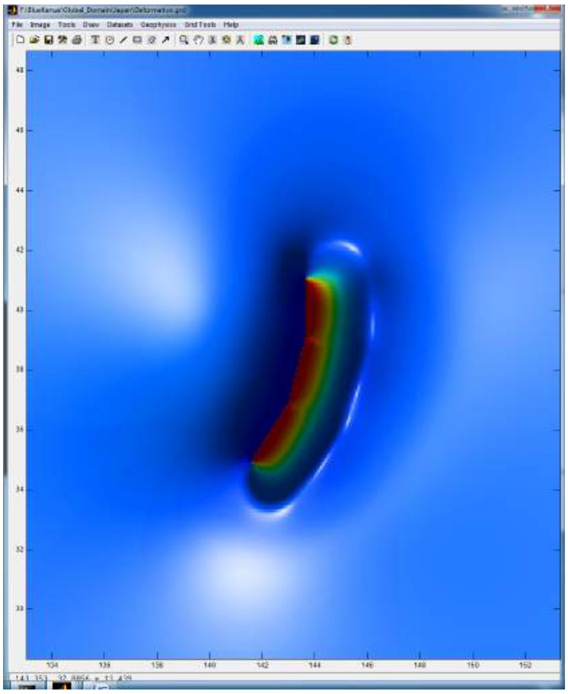
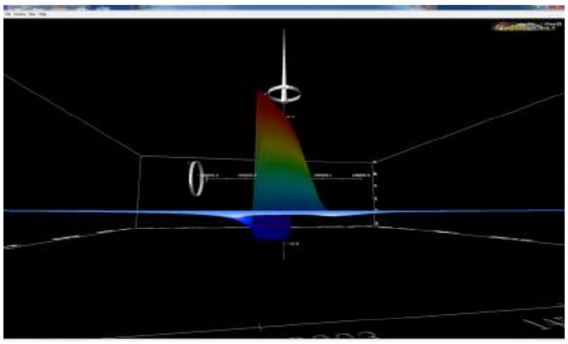
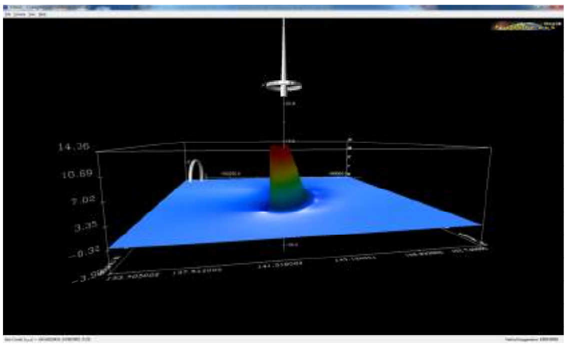


Finally the Vertical Elastic deformation apprears for multisegement fault and save it as sufer bin



grid Save as surfer grid bin format (Surfer 6 grid – Golden software grid binary format (ancient Surfer 6 format)) with naming convention " ***faultdeform.grd*** " .

This deformation grid ca be viewed with the iview3d free Fledermaus viewer by clicking the icon "eye" in the Tool bar

5. Model Setup

List of Files of TUNAMI FF includes following CUDA Kernel files, CUDA Header files, CPP files ad Header files:

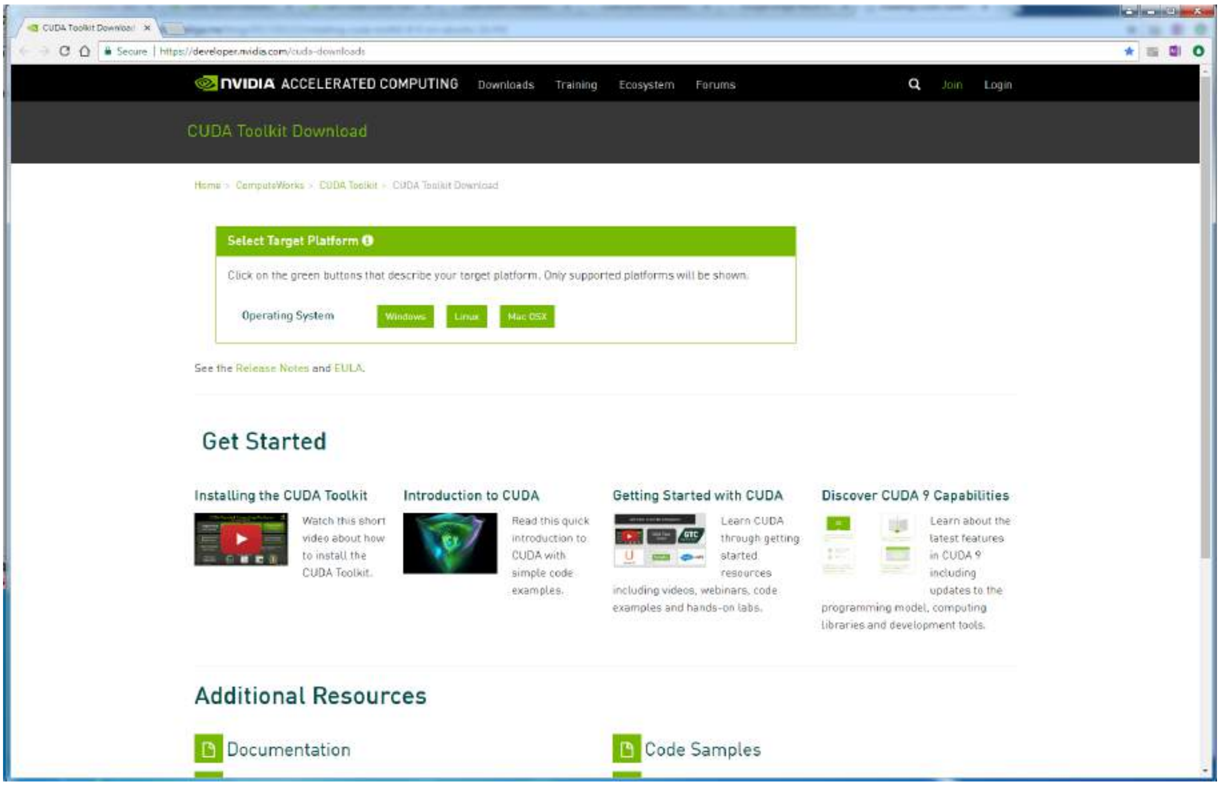
CUDA Kernel files	CUDA Header files	CPP files	Header files
GPUGNODEBOUNCK.cu	*GnodeCudaTUNAMIFF.cuh*	*GRDUTIL.cpp*	*GRDUTIL.h*
TUNAMI_FFGpu.cu	*MASSGBOUNDMOMENT.cuh*	*MASSGBOUNDMOMENT.cpp*	*DEFORMINITIA.h*
		TFFcalutilsub.cpp	*NODEINI.h*
			OFAULTPARA.h
			SPHERECAL.h
			TUNAMI_FF.h
			UTILMDL.h

These files are coded for 100 % compatibility to open source Linux platforms like Ubuntu with gnu c-compliers.

a. Pre-requisites

Latest CUDA toolkit is required along with appropriate GPU hardware with Ubuntu OS Linux.

Download CUDA toolkit from URL: https://developer.nvidia.com/cuda-downloads

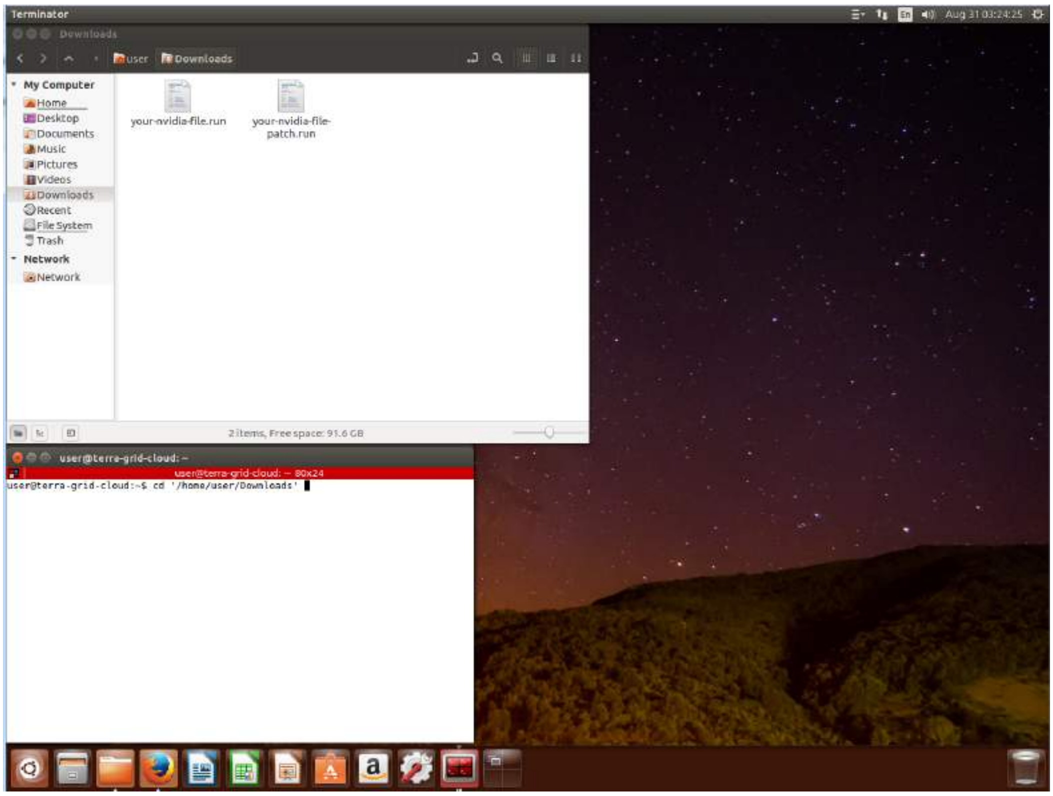


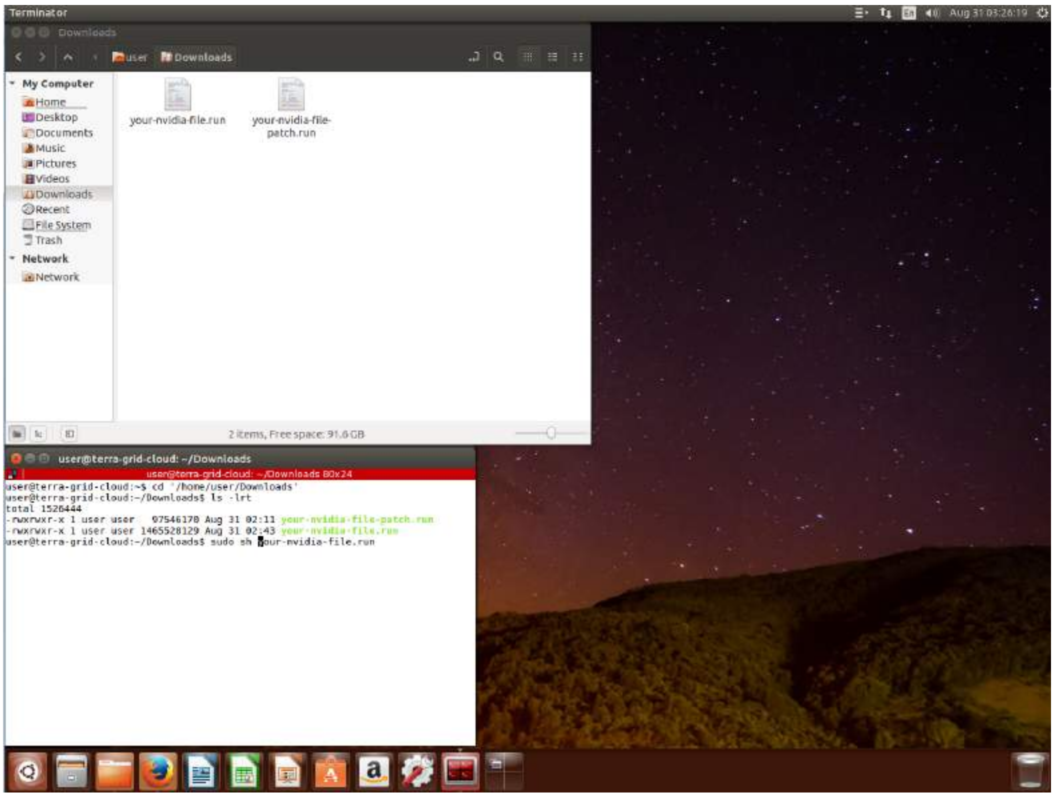
b. CUDA Installation

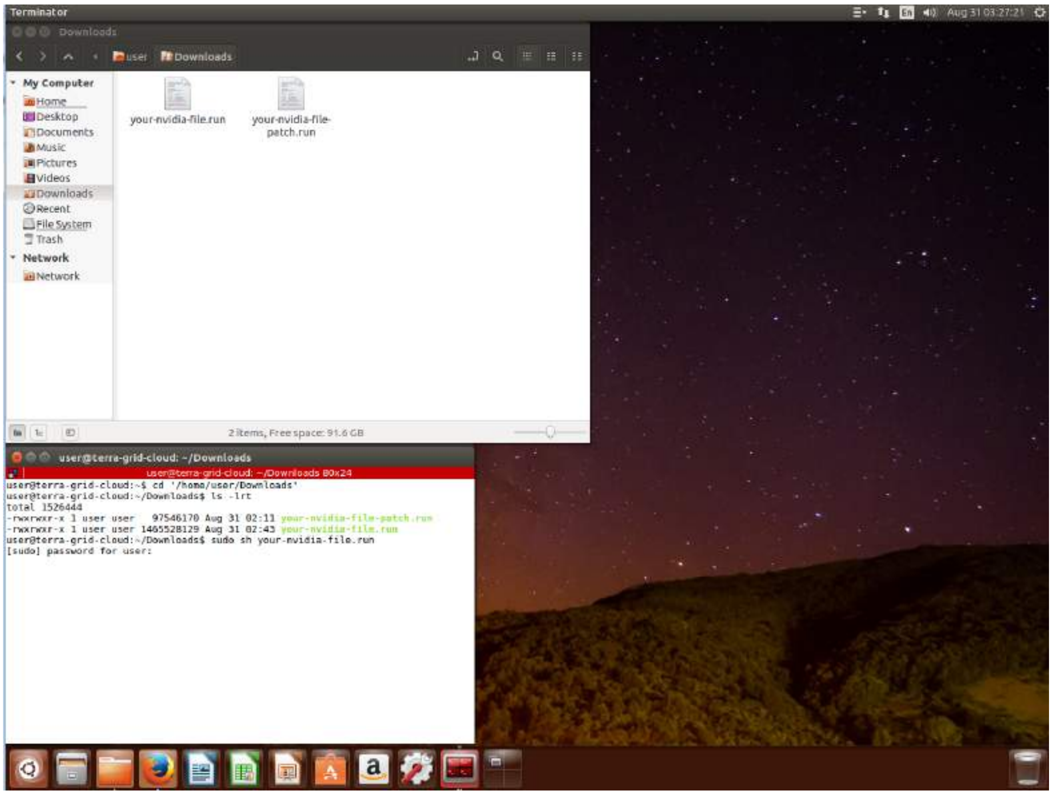
There two ways of installation procedures demonstrated to install CUDA toolkit in UBUNTU in this manual.

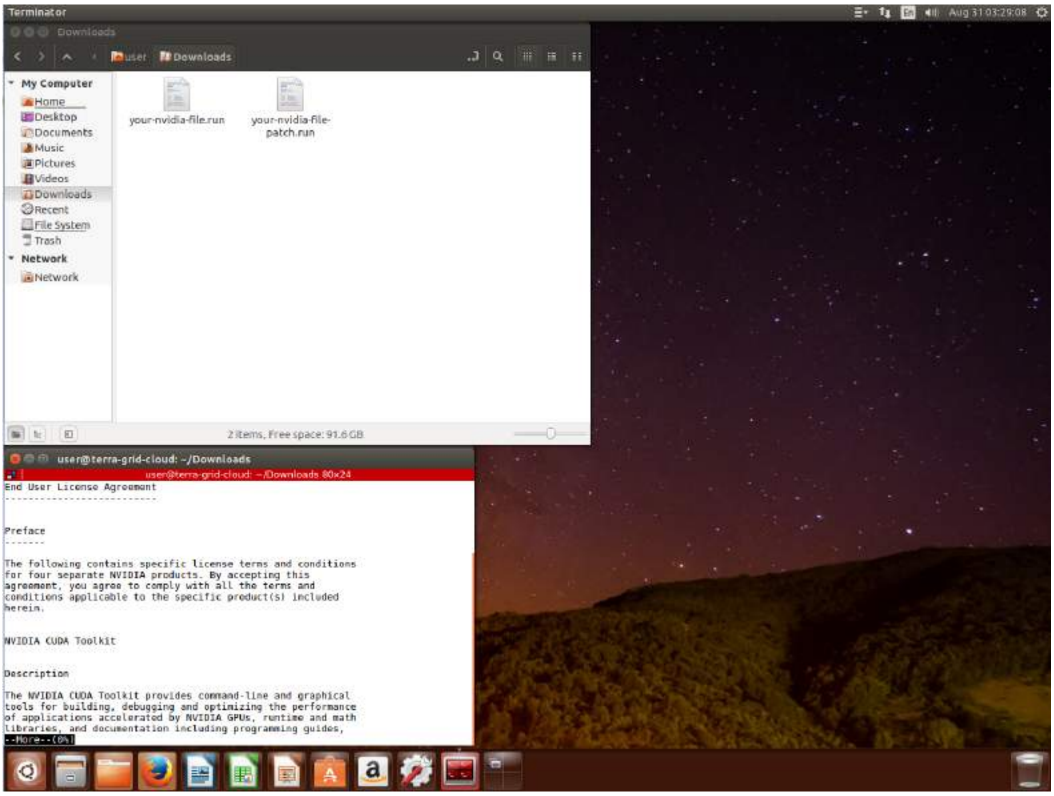
Procedure - 1:

Once the appropriate CUDA tool kit downloaded, one can follow the installation step as described in following pictures.


Terminator
Downloads
user
Downloads
My Computer
Home
Desktop
Documents
Music
Pictures
Videos
Downloads
Recent
File System
Trash
Network
Network
your-nvidia-file.run
your-nvidia-file-patch.run
2 items, Free space: 91.6 GB
user@terra-grid-cloud: ~
user@terra-grid-cloud: ~ 80x24
user@terra-grid-cloud:~$ cd '/home/user/Downloads'


Terminator
Downloads
user
Downloads
My Computer
Home
Desktop
Documents
Music
Pictures
Videos
Downloads
Recent
File System
Trash
Network
Network
your-nvidia-file.run
your-nvidia-file-patch.run
2 items, Free space: 91.6 GB
user@terra-grid-cloud: ~/Downloads
user@terra-grid-cloud: ~/Downloads 80x24
user@terra-grid-cloud:~$ cd '/home/user/Downloads'
user@terra-grid-cloud:~/Downloads$ ls -lrt
total 1526444
-rwxrwxr-x 1 user user 97546170 Aug 31 02:11 your-nvidia-file-patch.run
-rwxrwxr-x 1 user user 1465528129 Aug 31 02:43 your-nvidia-file.run
user@terra-grid-cloud:~/Downloads$ sudo sh your-nvidia-file.run


Terminator
Aug 31 03:27:21
Downloads
user
Downloads
My Computer
Home
Desktop
Documents
Music
Pictures
Videos
Downloads
Recent
File System
Trash
Network
Network
your-nvidia-file.run
your-nvidia-file-patch.run
2 items, Free space: 91.6 GB
user@terra-grid-cloud: ~/Downloads
user@terra-grid-cloud: ~/Downloads 80x24
user@terra-grid-cloud:~$ cd '/home/user/Downloads'
user@terra-grid-cloud:~/Downloads$ ls -lrt
total 1526444
-rwxrwxr-x 1 user user 97546170 Aug 31 02:11 your-nvidia-file-patch.run
-rwxrwxr-x 1 user user 1465528129 Aug 31 02:43 your-nvidia-file.run
user@terra-grid-cloud:~/Downloads$ sudo sh your-nvidia-file.run
[sudo] password for user:


Terminator
Aug 31 03:29:08
Downloads
user
Downloads
My Computer
Home
Desktop
Documents
Music
Pictures
Videos
Downloads
Recent
File System
Trash
Network
Network
your-nvidia-file.run
your-nvidia-file-patch.run
2 items, Free space: 91.6 GB
user@terra-grid-cloud: ~/Downloads
user@terra-grid-cloud: ~/Downloads 80x24
End User License Agreement
Preface
The following contains specific license terms and conditions for four separate NVIDIA products. By accepting this agreement, you agree to comply with all the terms and conditions applicable to the specific product(s) included herein.
NVIDIA CUDA Toolkit
Description
The NVIDIA CUDA Toolkit provides command-line and graphical tools for building, debugging and optimizing the performance of applications accelerated by NVIDIA GPUs, runtime and math libraries, and documentation including programming guides,
--More--(0%)

```
user@terra-grid-cloud: ~/Downloads
user@terra-grid-cloud: ~/Downloads 80x24
End User License Agreement
--------------------------


Preface
-------

The following contains specific license terms and conditions
for four separate NVIDIA products. By accepting this
agreement, you agree to comply with all the terms and
conditions applicable to the specific product(s) included
herein.


NVIDIA CUDA Toolkit


Description

The NVIDIA CUDA Toolkit provides command-line and graphical
tools for building, debugging and optimizing the performance
of applications accelerated by NVIDIA GPUs, runtime and math
libraries, and documentation including programming guides,
--More--(0%)
```

```
user@terra-grid-cloud: ~/Downloads
user@terra-grid-cloud: ~/Downloads 80x24
More information, including licensing information, about the
NVIDIA CUDA Toolkit and the NVIDIA CUDA Samples can be found
at: http://www.nvidia.com/getcuda

More information, including licensing information, about the
NVIDIA DirectX SDK can be found at:
http://developer.nvidia.com/object/sdk_home.html


6. NVIDIA CUDA General Terms
----------------------------

The Software, on the Windows platform, may collect
non-personally identifiable information for the purposes of
customizing information delivered to you and improving future
versions of the Software. Such information, including IP
address and system configuration, will only be collected on an
anonymous basis and cannot be linked to any personally
identifiable information. Personally identifiable information
such as your username or hostname is not collected.

-------------------------------------------------------------
 Do you accept the previously read EULA?
accept/decline/quit:
```

user@terra-grid-cloud: ~/Downloads

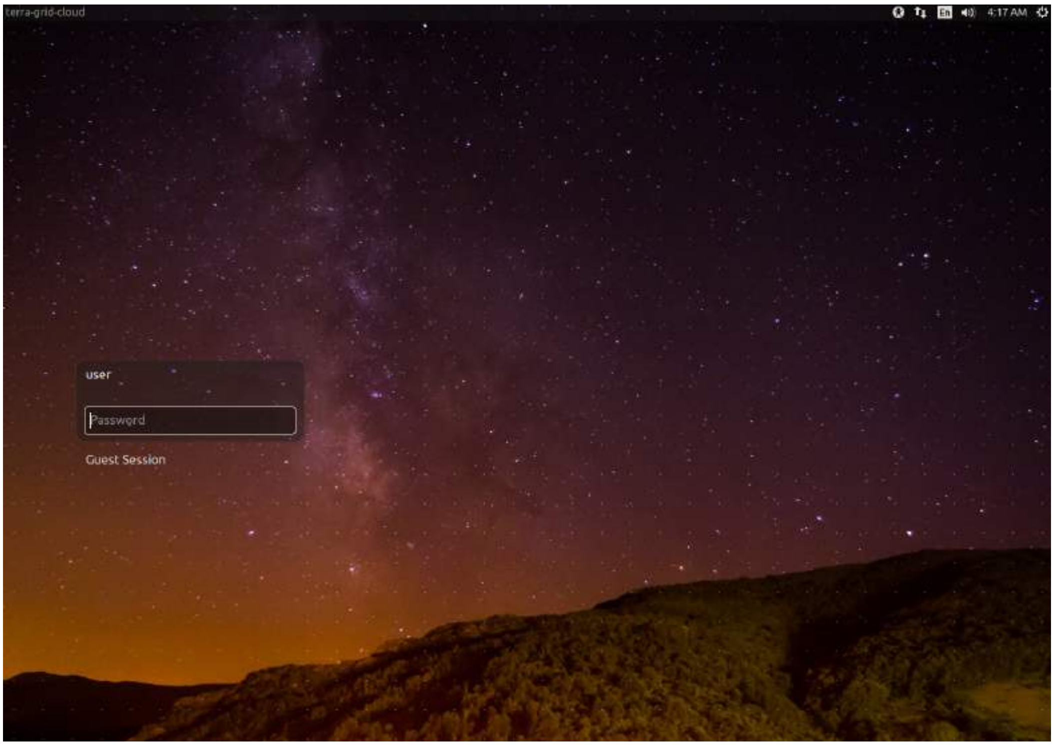
user@terra-grid-cloud: ~/Downloads 80x24

```
non-personally identifiable information for the purposes of
customizing information delivered to you and improving future
versions of the Software. Such information, including IP
address and system configuration, will only be collected on an
anonymous basis and cannot be linked to any personally
identifiable information. Personally identifiable information
such as your username or hostname is not collected.

------------------------------------------------------------
 Do you accept the previously read EULA?
accept/decline/quit:accept

Install NVIDIA Accelerated Graphics Driver for Linux-x86_64 375.26?
(y)es/(n)o/(q)uit: y

Do you want to install the OpenGL libraries?
(y)es/(n)o/(q)uit [ default is yes ]: y

Do you want to run nvidia-xconfig?
This will update the system X configuration file so that the NVIDIA X driver
is used. The pre-existing X configuration file will be backed up.
This option should not be used on systems that require a custom
X configuration, such as systems with multiple GPU vendors.
(y)es/(n)o/(q)uit [ default is no ]: y
```

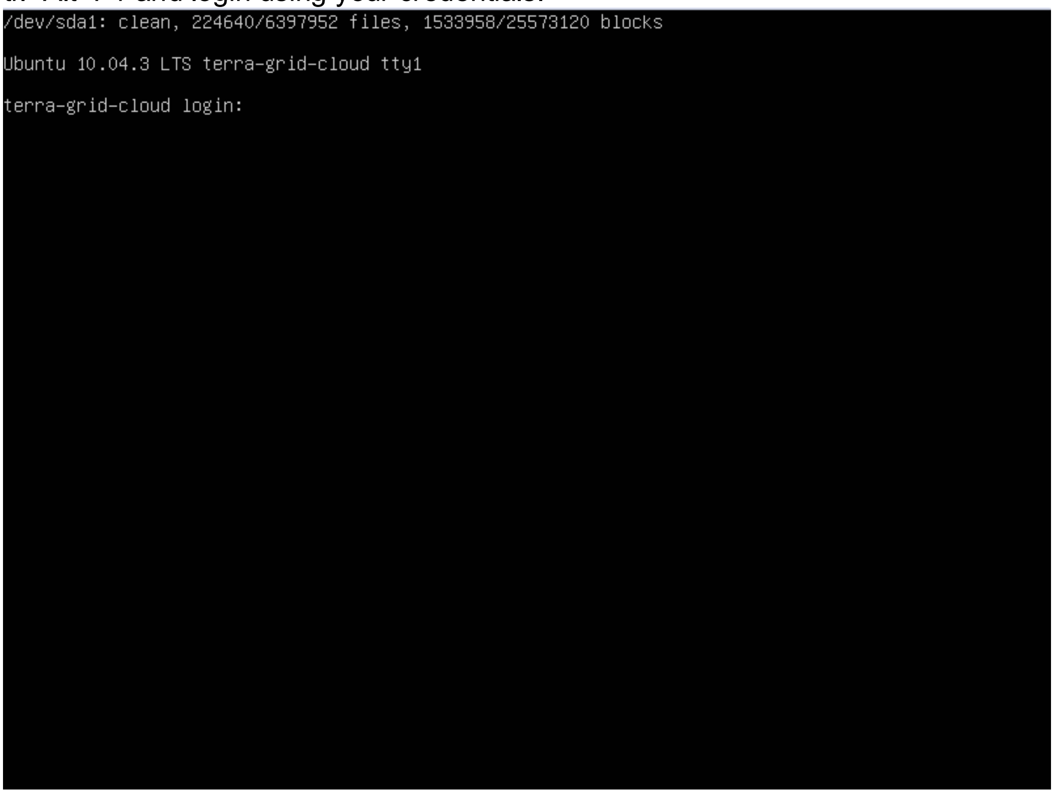
```
user@terra-grid-cloud: ~/Downloads
user@terra-grid-cloud: ~/Downloads 80x46

Install NVIDIA Accelerated Graphics Driver for Linux-x86_64 375.26?
(y)es/(n)o/(q)uit: y

Do you want to install the OpenGL libraries?
(y)es/(n)o/(q)uit [ default is yes ]: y

Do you want to run nvidia-xconfig?
This will update the system X configuration file so that the NVIDIA X driver
is used. The pre-existing X configuration file will be backed up.
This option should not be used on systems that require a custom
X configuration, such as systems with multiple GPU vendors.
(y)es/(n)o/(q)uit [ default is no ]: y

Install the CUDA 4.0 Toolkit?
(y)es/(n)o/(q)uit: y

Enter Toolkit Location
 [ default is /usr/local/cuda-4.0 ]:

Do you want to install a symbolic link at /usr/local/cuda?
(y)es/(n)o/(q)uit: y

Install the CUDA 4.0 Samples?
(y)es/(n)o/(q)uit: y

Enter CUDA Samples Location
 [ default is /home/user ]:

Installing the NVIDIA display driver...
It appears that an X server is running. Please exit X before installation. If yo
u're sure that X is not running, but are getting this error, please delete any X
 lock files in /tmp.

===========
= Summary =
===========

Driver:   Installation Failed
Toolkit:  Installation skipped
Samples:  Installation skipped


Logfile is /tmp/cuda_install_3898.log
user@terra-grid-cloud:~/Downloads$
```

As the error states, you are still running an X server. This error occurs when you try to install the Nvidia .run files while logged in.

Make sure you are logged out.



1. Hit Ctrl+Alt+F1 and login using your credentials.



```
/dev/sda1: clean, 224640/6397952 files, 1533958/25573120 blocks

Ubuntu 10.04.3 LTS terra-grid-cloud tty1

terra-grid-cloud login:
```

```
/dev/sda1: clean, 224640/6397952 files, 1533958/25573120 blocks

Ubuntu 10.04.3 LTS terra-grid-cloud tty1

terra-grid-cloud login: user
Password:
Last login: Thu Aug 24 20:27:51 EDT 2011 on tty1
Welcome to Ubuntu 10.04.3 LTS (GNU/Linux 4.10.0-33-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

20 packages can be updated.
0 updates are security updates.

user@terra-grid-cloud:~$ cd Downloads/
user@terra-grid-cloud:~/Downloads$ ls -lrt
total 1526444
-rwxrwxr-x 1 user user   97546170 Aug 31 02:11 your-nvidia-file-patch.run
-rwxrwxr-x 1 user user 1465528129 Aug 31 02:43 your-nvidia-file.run
user@terra-grid-cloud:~/Downloads$
```

2. Kill your current X server session by typing sudo service lightdm stop or sudo lightdm stop

```
user@terra-grid-cloud:~$ cd Downloads/
user@terra-grid-cloud:~/Downloads$ ls -lrt
total 1526444
-rwxrwxr-x 1 user user   97546170 Aug 31 02:11 your-nvidia-file-patch.run
-rwxrwxr-x 1 user user 1465528129 Aug 31 02:43 your-nvidia-file.run
user@terra-grid-cloud:~/Downloads$ sudo service lightdm stop
[sudo] password for user:
user@terra-grid-cloud:~/Downloads$
```

3. Enter runlevel 3 by typing sudo init 3

```
user@terra-grid-cloud:~$ cd Downloads/
user@terra-grid-cloud:~/Downloads$ ls -lrt
total 1526444
-rwxrwxr-x 1 user user   97546170 Aug 31 02:11 your-nvidia-file-patch.run
-rwxrwxr-x 1 user user 1465528129 Aug 31 02:43 your-nvidia-file.run
user@terra-grid-cloud:~/Downloads$ sudo service lightdm stop
[sudo] password for user:
user@terra-grid-cloud:~/Downloads$ sudo init 3
user@terra-grid-cloud:~/Downloads$
```

4. Install your *.run file.

a. you change to the directory where you have downloaded the file by typing for instance cd Downloads. If it is in another directory, go there. Check if you see the file when you type ls NVIDIA*

b. Make the file executable with chmod +x ./your-nvidia-file.run

```
user@terra-grid-cloud:~/Downloads$ sudo init 3
user@terra-grid-cloud:~/Downloads$ chmod 775 *nvi*
user@terra-grid-cloud:~/Downloads$ _
```

c. Execute the file with sudo ./your-nvidia-file.run

```
user@terra-grid-cloud:~$ cd Downloads/
user@terra-grid-cloud:~/Downloads$ ls -lrt
total 1526444
-rwxrwxr-x 1 user user   97546170 Aug 31 02:11 your-nvidia-file-patch.run
-rwxrwxr-x 1 user user 1465528129 Aug 31 02:43 your-nvidia-file.run
user@terra-grid-cloud:~/Downloads$ sudo service lightdm stop
[sudo] password for user:
user@terra-grid-cloud:~/Downloads$ sudo init 3
user@terra-grid-cloud:~/Downloads$ chmod 775 *nvi*
user@terra-grid-cloud:~/Downloads$ sudo sh your-nvidia-file.run
```

```
End User License Agreement
--------------------------


Preface
-------

The following contains specific license terms and conditions
for four separate NVIDIA products. By accepting this
agreement, you agree to comply with all the terms and
conditions applicable to the specific product(s) included
herein.


NVIDIA CUDA Toolkit


Description

The NVIDIA CUDA Toolkit provides command-line and graphical
tools for building, debugging and optimizing the performance
of applications accelerated by NVIDIA GPUs, runtime and math
libraries, and documentation including programming guides,
user manuals, and API references. The NVIDIA CUDA Toolkit
License Agreement is available in Chapter 1.


Default Install Location of CUDA Toolkit

Windows platform:

%ProgramFiles%\NVIDIA GPU Computing Toolkit\CUDA\v#.#

Linux platform:

/usr/local/cuda-#.#
--More--(0%)_
```

```
you because your country may not allow the exclusion or limitation of
incidental, consequential or other damages.

The Software contains components, as listed below that are
licensed to Licensee pursuant to the terms and conditions of
their respective End User License Agreements:

  * NVIDIA CUDA Samples

  * NVIDIA CUDA Toolkit

  * NVIDIA DirectX SDK

More information, including licensing information, about the
NVIDIA CUDA Toolkit and the NVIDIA CUDA Samples can be found
at: http://www.nvidia.com/getcuda

More information, including licensing information, about the
NVIDIA DirectX SDK can be found at:
http://developer.nvidia.com/object/sdk_home.html


6. NVIDIA CUDA General Terms
----------------------------

The Software, on the Windows platform, may collect
non-personally identifiable information for the purposes of
customizing information delivered to you and improving future
versions of the Software. Such information, including IP
address and system configuration, will only be collected on an
anonymous basis and cannot be linked to any personally
identifiable information. Personally identifiable information
such as your username or hostname is not collected.

-------------------------------------------------------------
 Do you accept the previously read EULA?
accept/decline/quit:_
```

```
More information, including licensing information, about the
NVIDIA CUDA Toolkit and the NVIDIA CUDA Samples can be found
at: http://www.nvidia.com/getcuda

More information, including licensing information, about the
NVIDIA DirectX SDK can be found at:
http://developer.nvidia.com/object/sdk_home.html


6. NVIDIA CUDA General Terms
----------------------------

The Software, on the Windows platform, may collect
non-personally identifiable information for the purposes of
customizing information delivered to you and improving future
versions of the Software. Such information, including IP
address and system configuration, will only be collected on an
anonymous basis and cannot be linked to any personally
identifiable information. Personally identifiable information
such as your username or hostname is not collected.

-------------------------------------------------------------
 Do you accept the previously read EULA?
accept/decline/quit:accept

Install NVIDIA Accelerated Graphics Driver for Linux-x86_64 375.26?
(y)es/(n)o/(q)uit: y

Do you want to install the OpenGL libraries?
(y)es/(n)o/(q)uit [ default is yes ]: y

Do you want to run nvidia-xconfig?
This will update the system X configuration file so that the NVIDIA X driver
is used. The pre-existing X configuration file will be backed up.
This option should not be used on systems that require a custom
X configuration, such as systems with multiple GPU vendors.
(y)es/(n)o/(q)uit [ default is no ]: y_
```

```
identifiable information. Personally identifiable information
such as your username or hostname is not collected.

-------------------------------------------------------------
 Do you accept the previously read EULA?
accept/decline/quit:accept

Install NVIDIA Accelerated Graphics Driver for Linux-x86_64 375.26?
(y)es/(n)o/(q)uit: y

Do you want to install the OpenGL libraries?
(y)es/(n)o/(q)uit [ default is yes ]: y

Do you want to run nvidia-xconfig?
This will update the system X configuration file so that the NVIDIA X driver
is used. The pre-existing X configuration file will be backed up.
This option should not be used on systems that require a custom
X configuration, such as systems with multiple GPU vendors.
(y)es/(n)o/(q)uit [ default is no ]: y

Install the CUDA 4.0 Toolkit?
(y)es/(n)o/(q)uit: y

Enter Toolkit Location
 [ default is /usr/local/cuda-4.0 ]:

Do you want to install a symbolic link at /usr/local/cuda?
(y)es/(n)o/(q)uit: y

Install the CUDA 4.0 Samples?
(y)es/(n)o/(q)uit: y

Enter CUDA Samples Location
 [ default is /home/user ]:

Installing the NVIDIA display driver...
_
```

```
Enter CUDA Samples Location
 [ default is /home/user ]:

Installing the NVIDIA display driver...
The driver installation is unable to locate the kernel source. Please make sure that the kernel sour
ce packages are installed and set up correctly.
If you know that the kernel source packages are installed and set up correctly, you may pass the loc
ation of the kernel source with the '--kernel-source-path' flag.

===========
= Summary =
===========

Driver:   Installation Failed
Toolkit:  Installation skipped
Samples:  Installation skipped


Logfile is /tmp/cuda_install_4632.log
user@terra-grid-cloud:~/Downloads$ sudo apt-get install build-essential
Reading package lists... Done
Building dependency tree
Reading state information... Done
build-essential is already the newest version (12.1ubuntu2).
0 upgraded, 0 newly installed, 0 to remove and 23 not upgraded.
user@terra-grid-cloud:~/Downloads$ sudo apt-get install linux-image-extra-virtual_
```

```
===========
= Summary =
===========

Driver:   Installation Failed
Toolkit:  Installation skipped
Samples:  Installation skipped


Logfile is /tmp/cuda_install_4632.log
user@terra-grid-cloud:~/Downloads$ sudo apt-get install build-essential
Reading package lists... Done
Building dependency tree
Reading state information... Done
build-essential is already the newest version (12.1ubuntu2).
0 upgraded, 0 newly installed, 0 to remove and 23 not upgraded.
user@terra-grid-cloud:~/Downloads$ sudo apt-get install linux-image-extra-virtual
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  linux-image-4.4.0-93-generic linux-image-extra-4.4.0-93-generic linux-image-generic
Suggested packages:
  fdutils linux-doc-4.4.0 | linux-source-4.4.0 linux-tools linux-headers-4.4.0-93-generic
The following NEW packages will be installed:
  linux-image-4.4.0-93-generic linux-image-extra-4.4.0-93-generic linux-image-extra-virtual
  linux-image-generic
0 upgraded, 4 newly installed, 0 to remove and 23 not upgraded.
Need to get 57.8 MB of archives.
After this operation, 219 MB of additional disk space will be used.
Do you want to continue? [Y/n] _
```

```
user@terra-grid-cloud:~/Downloads$ sudo apt-get install linux-source
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  linux-source-4.4.0
Suggested packages:
  libncurses-dev | ncurses-dev kernel-package libqt3-dev
The following NEW packages will be installed:
  linux-source linux-source-4.4.0
0 upgraded, 2 newly installed, 0 to remove and 23 not upgraded.
Need to get 112 MB of archives.
After this operation, 130 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

```
user@terra-grid-cloud:~/Downloads$ sudo apt-get source linux-image-$(uname -r)
Reading package lists... Done
Picking 'linux-hwe' as source package instead of 'linux-image-4.10.0-33-generic'
E: Unable to find a source package for linux-hwe
user@terra-grid-cloud:~/Downloads$
```

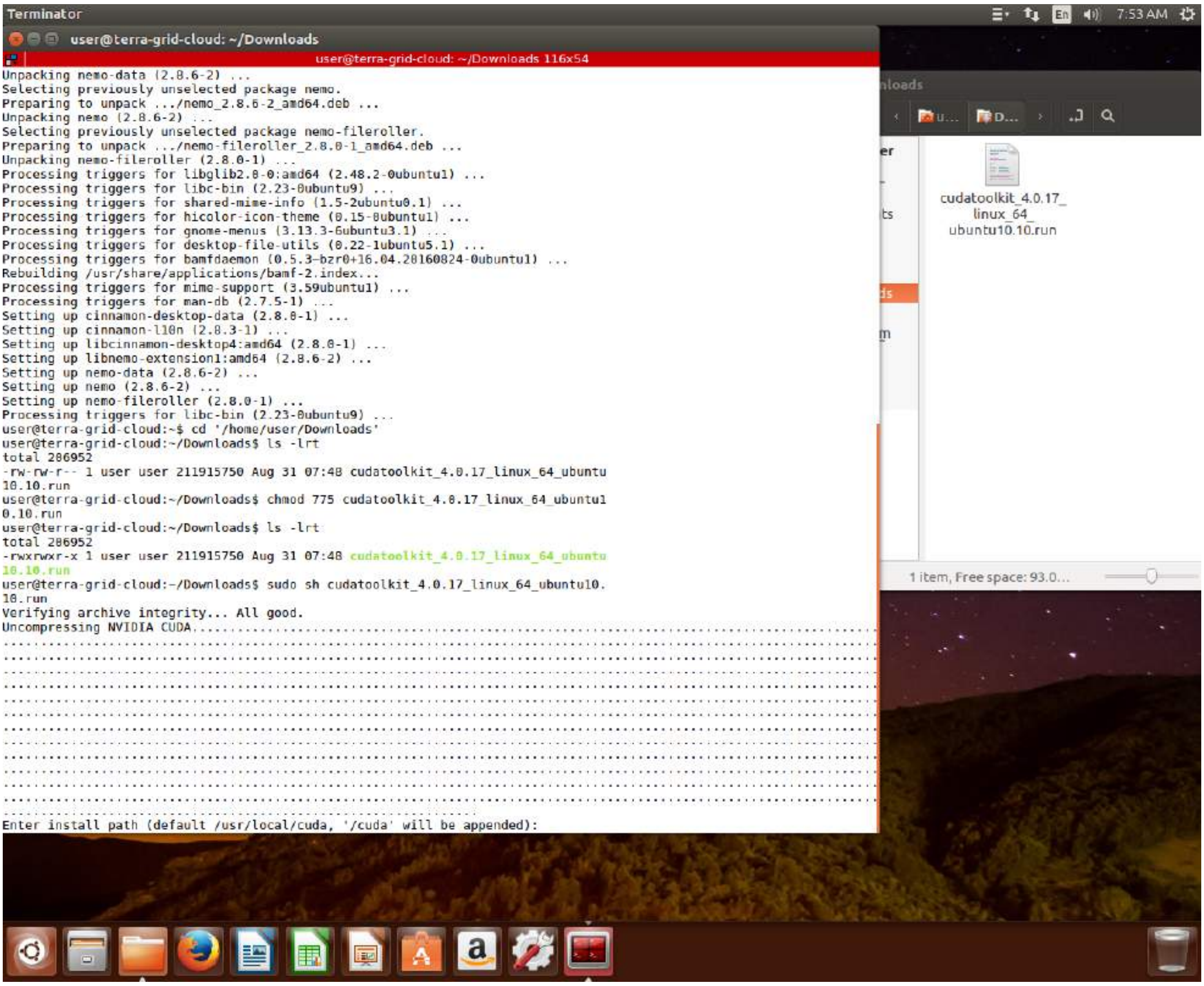
```
user@terra-grid-cloud:~/Downloads$ sudo apt-get install linux-headers-$(uname -r)
Reading package lists... Done
Building dependency tree
Reading state information... Done
linux-headers-4.10.0-33-generic is already the newest version (4.10.0-33.37~16.04.1).
linux-headers-4.10.0-33-generic set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 23 not upgraded.
user@terra-grid-cloud:~/Downloads$
```

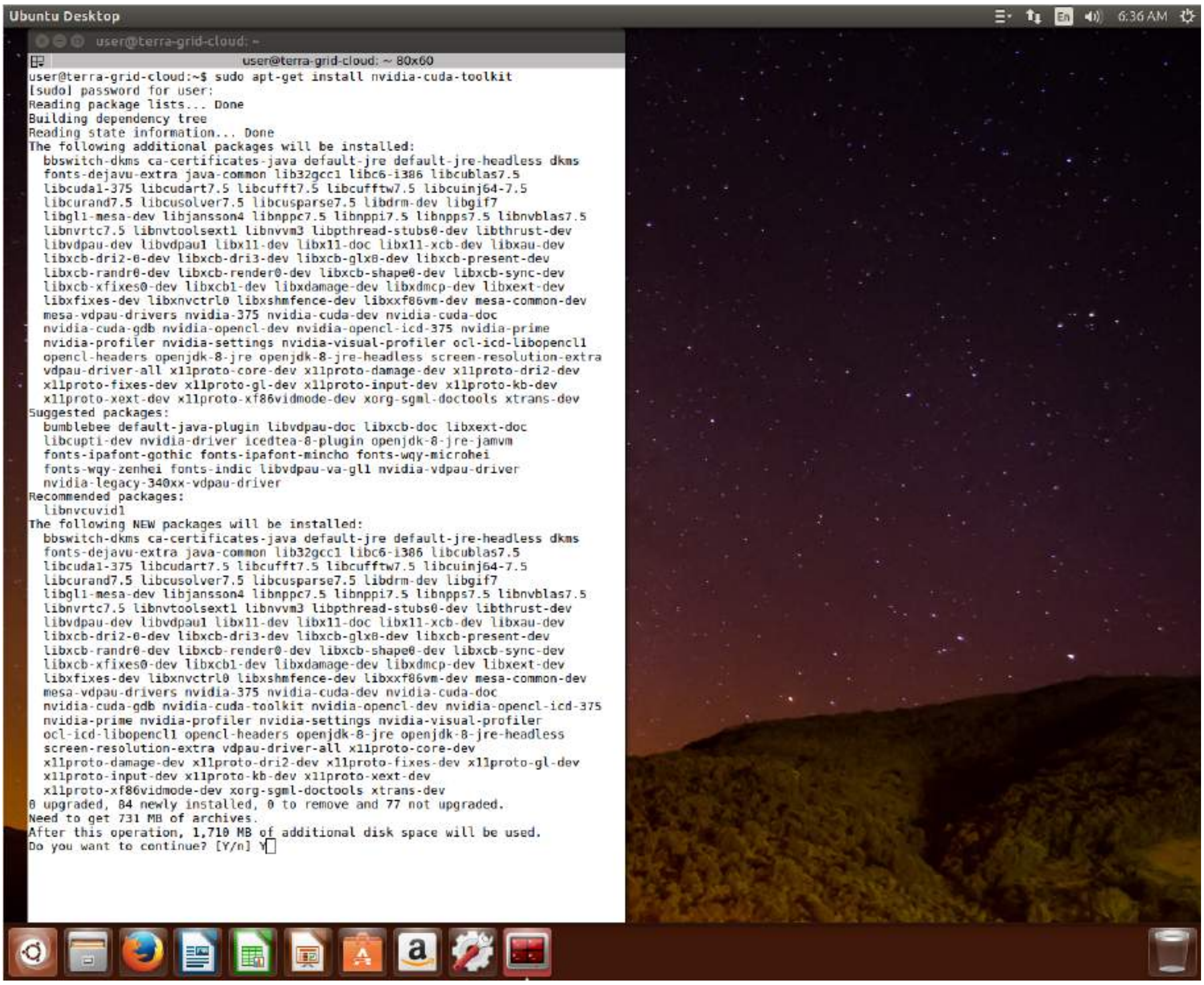
```
Installing the CUDA Toolkit in /usr/local/cuda-4.0 ...
Missing recommended library: libGLU.so
Missing recommended library: libX11.so
Missing recommended library: libXi.so
Missing recommended library: libXmu.so
Missing recommended library: libGL.so

Installing the CUDA Samples in /home/user ...
Copying samples to /home/user/NVIDIA_CUDA-4.0_Samples now...
Finished copying samples.

===========
= Summary =
===========

Driver:   Not Selected
Toolkit:  Installed in /usr/local/cuda-4.0
Samples:  Installed in /home/user, but missing recommended libraries

Please make sure that
 -   PATH includes /usr/local/cuda-4.0/bin
 -   LD_LIBRARY_PATH includes /usr/local/cuda-4.0/lib64, or, add /usr/local/cuda-4.0/lib64 to /etc/l
d.so.conf and run ldconfig as root

To uninstall the CUDA Toolkit, run the uninstall script in /usr/local/cuda-4.0/bin

Please see CUDA_Installation_Guide_Linux.pdf in /usr/local/cuda-4.0/doc/pdf for detailed information
 on setting up CUDA.

***WARNING: Incomplete installation! This installation did not install the CUDA Driver. A driver of
version at least 361.00 is required for CUDA 4.0 functionality to work.
To install the driver using this installer, run the following command, replacing <CudaInstaller> wit
h the name of this run file:
    sudo <CudaInstaller>.run -silent -driver

Logfile is /tmp/cuda_install_19019.log
user@terra-grid-cloud:~/Downloads$
```

```
/group__CUDART__MEMORY_ge07c97b96efd09abaeb3ca3b5f8da4ee.html'
'doc/html/structcudaDeviceProp_9a63114766c4d2309f00403c1bf056c8.html' -> '/usr/local/cuda/doc/html/s
tructcudaDeviceProp_9a63114766c4d2309f00403c1bf056c8.html'
'doc/html/functions.html' -> '/usr/local/cuda/doc/html/functions.html'
'doc/html/group__CUDA__MEM_gaef08a7ccd61112f94e82f2b30d43627.html' -> '/usr/local/cuda/doc/html/grou
p__CUDA__MEM_gaef08a7ccd61112f94e82f2b30d43627.html'
'doc/OpenCL_Programming_Overview.pdf' -> '/usr/local/cuda/doc/OpenCL_Programming_Overview.pdf'
'doc/OpenCL_Programming_Guide.pdf' -> '/usr/local/cuda/doc/OpenCL_Programming_Guide.pdf'
'doc/Thrust_Quick_Start_Guide.pdf' -> '/usr/local/cuda/doc/Thrust_Quick_Start_Guide.pdf'
'doc/CUDA_VideoDecoder_Library.pdf' -> '/usr/local/cuda/doc/CUDA_VideoDecoder_Library.pdf'
'doc/OpenCL_Best_Practices_Guide.pdf' -> '/usr/local/cuda/doc/OpenCL_Best_Practices_Guide.pdf'
'doc/CUSPARSE_Library.pdf' -> '/usr/local/cuda/doc/CUSPARSE_Library.pdf'
'doc/CUDA_Toolkit_Release_Notes.txt' -> '/usr/local/cuda/doc/CUDA_Toolkit_Release_Notes.txt'
'doc/CURAND_Library.pdf' -> '/usr/local/cuda/doc/CURAND_Library.pdf'
'doc/cuda-gdb.pdf' -> '/usr/local/cuda/doc/cuda-gdb.pdf'
'doc/Fermi_Tuning_Guide.pdf' -> '/usr/local/cuda/doc/Fermi_Tuning_Guide.pdf'
'doc/OpenCL_Jumpstart_Guide.pdf' -> '/usr/local/cuda/doc/OpenCL_Jumpstart_Guide.pdf'
'doc/Using_Inline_PTX_Assembly_In_CUDA.pdf' -> '/usr/local/cuda/doc/Using_Inline_PTX_Assembly_In_CUD
A.pdf'

========================================

* Please make sure your PATH includes /usr/local/cuda/bin
* Please make sure your LD_LIBRARY_PATH
*   for 32-bit Linux distributions includes /usr/local/cuda/lib
*   for 64-bit Linux distributions includes /usr/local/cuda/lib64:/usr/local/cuda/lib
* OR
*   for 32-bit Linux distributions add /usr/local/cuda/lib
*   for 64-bit Linux distributions add /usr/local/cuda/lib64 and /usr/local/cuda/lib
* to /etc/ld.so.conf and run ldconfig as root

* Please read the release notes in /usr/local/cuda/doc/

* To uninstall CUDA, delete /usr/local/cuda
* Installation Complete

user@terra-grid-cloud:~/Downloads$ sudo sh cudatoolkit_4.0.17_linux_64_ubuntu10.10.run
```

5. You might be required to reboot when the installation finishes. If not, run sudo service lightdm start or sudo start lightdm to start your X server again.

<u>Procedure - 2:</u> Using "*apt-get"* install command step as described in following pictures.





Once CUDA toolkit installed with appropriate GPU driver version along with c++ compiler, the Model setup is completed for compilation and execution.

6. Model Execution

Setting up of "Model Parameters" for 'Execution' and 'Compilation':

BathymetryInputfile - Bathymetry

fileSource - Source: Okada faults or Surfer grid

KEtotalTime - Time steps of Total simulation - end of computation, [sec]

oopsName - Simulation Run

coriolis - Use Coriolis force

MAXOUT - Periodic dumping of mariograms and cumulative SpatialWave-plots (wavemax, arrival times), [sec]

outProgress - Reporting simulation progress, [sec model time]

KDspatialOut - Time step lengths to output spatial wave profiles, [sec model time]

dmin - minimal calculation depth, [m]

DT - time step length in second

zout0TRel - Initial uplift: relative threshold

zout0TAbs - Initial uplift: absolute threshold, [m]

zoutAT - Threshold for SpatialWave arrival time (0 - do not calculate), [m]

zoutCT - Threshold for clipping of expanding computational area, [m]

zoutZT - Threshold for resetting the small zout (keep expanding area from unnecessary growing), [m]

zoutTT - Threshold for transparency, [m]

fileGAGUEs - Points Of Interest (GAGUEs) input file

GAGUEDIMX - GAGUE fitting: max search distance, [km]

GAGUEDEMN - GAGUE fitting: min depth, [m]

GAGUEDEMX - GAGUE fitting: max depth, [m]

GAGUERT - report of GAGUE loading

KCgagueOUT - Time step lengths in outputting the time history of water level, [sec]

gpu - GPU computation enable or disable

Note: The parameter '***DT***' time step length in second (**= 1 sec 'default'**) needs to be set appropriately in accordance with '**CFL criteria**', based on spatial resolution of input Bathymetry data file '***BathymetryInputfile***'.

Example:

```
BathymetryInputfile     = strdup("Ogrid.grd");
fileSource              = strdup("faultdeform.grd'
KEtotalTime             = 600;
KEtotalTime *           = 60;
oopsName                = strdup( "OOPS" );
coriolis                = 1;
MAXOUT                  = 0;
outProgress             = 60;
KDspatialOut            = 60;
dmin                    = 10.;
DT                      = 1;
zout0TRel               = 0.01;
zout0TAbs               = 0.0;
zoutAT                  = 0.001;
zoutCT                  = 1.e-4;
zoutZT                  = 1.e-5;
zoutTT                  = 0.0;
fileGAGUEs              = NULL;
GAGUEDIMX               = 10.0;
GAGUEDIMX *             = 1000.;
GAGUEDEMN               = 1.0;
GAGUEDEMX               = 10000.0;
GAGUERT                 = 0;
KCgagueOUT              = 60;
gpu                     = true;
//gpu = false;
adjustZtop              = false;
verbose                 = true;
```

a. Compilation

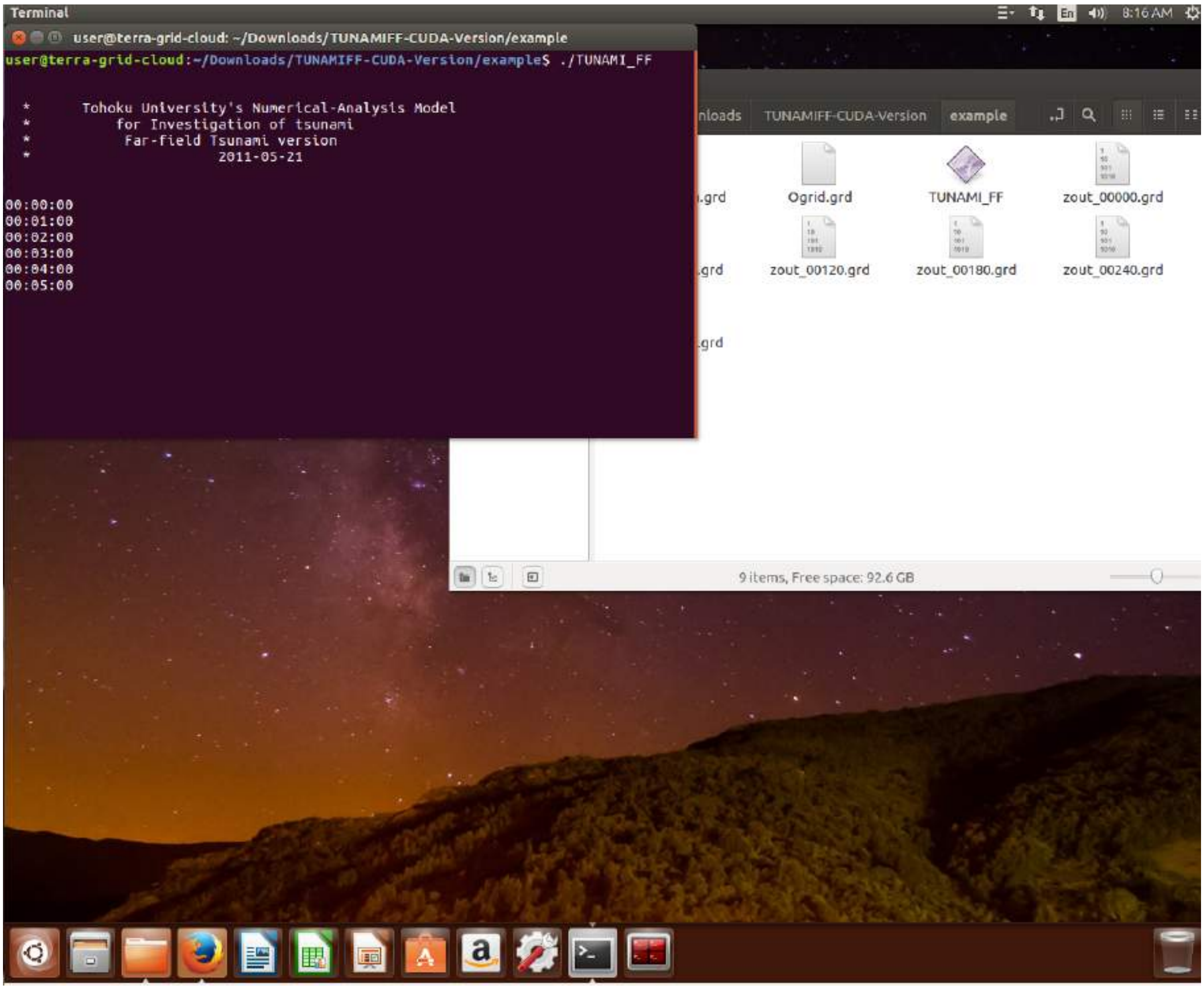
The software TUNAMI FF can be complied with gnu c++ compiler as well Intel C compiler as described follows:

```
g++ -I ./ -c TFFcalutilsub.cpp

g++ -I ./ -c GRDUTIL.cpp

g++ -I ./ -c MASSGBOUNDMOMENT.cpp

nvcc -ccbin g++ -I ./ -c GPUGNODEBOUNCK.cu

nvcc -ccbin g++ -I ./ -c TUNAMI_FFGpu.cu
```

Command to get TUNAMI FF executable

nvcc -Xcompiler -fopenmp -lgomp -ccbin g++ -o TUNAMI_FF TFFcalutilsub.o GRDUTIL.o MASSGBOUNDMOMENT.o GPUGNODEBOUNCK.o TUNAMI_FFGpu.o

Sample run of TUNAMI FF executable



In execution of TUNAMI FF software program it shows the following header at starting of terminal output

Tohoku University's Numerical-Analysis Model   ***for Investigation of tsunami***   ***Far-field Tsunami version***   ***2011-05-21***

7. Model Results visualisation

There are several output files which are mostly grid files showing sea state at different times and locations.

These files are:

zout*****.grd files: these files show the sea state at a specified instant. Stars in the file name represents the time at which the sea state is stored. These files are generated according to the time interval given (**KDspatialOut**) from the beginning of the simulation until the total simulation time (**KEtotalTime**).

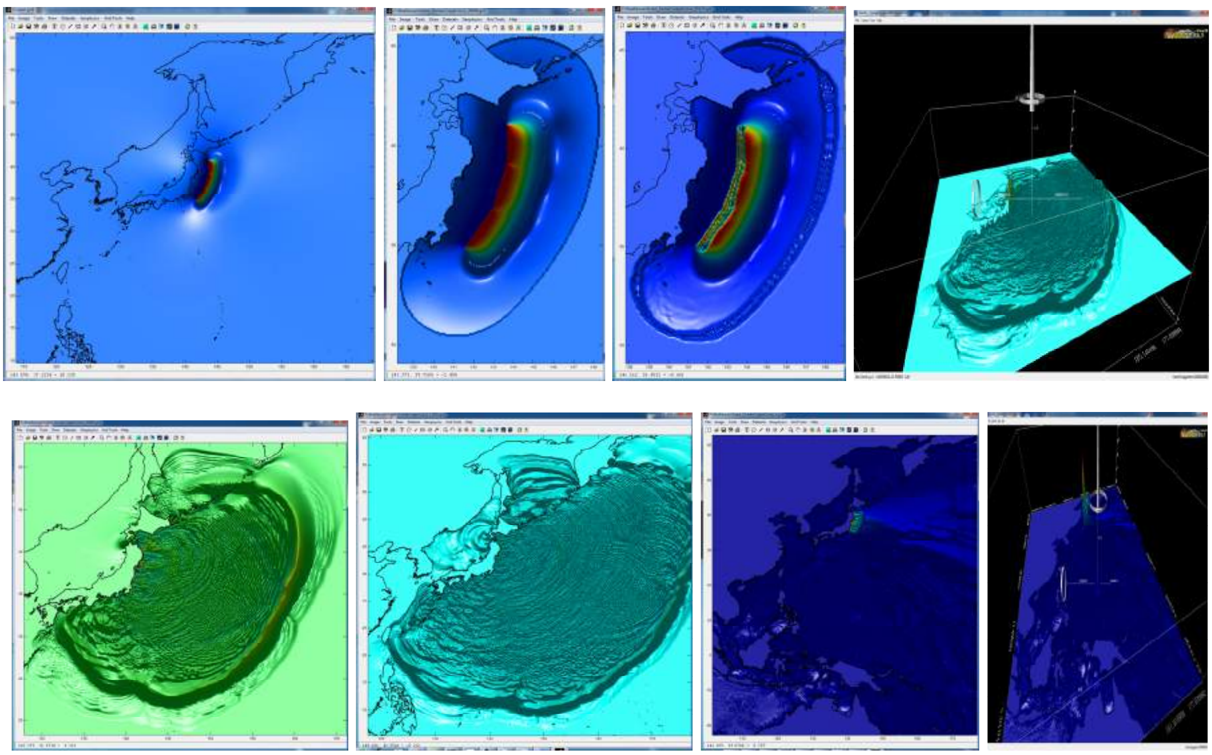
zmax.grd: shows the maximum water level at each grid during simulation at time = **KEtotalTime**.

a. Visualisation

The output of TUNAMI FF program is not easily interpreted as given in the output files. In order to get the best results, the output files are needed to be converted in to diagrams or graphs so that the interpretation and the comparison of different data can be achieved easily.

Using zout***.grd files and the bathymetry file of the area, the simulation of the tsunami is accomplished.

The diagrams or the graphs of the output files are mostly prepared by using the programs Mirone and/or iview-3D viewer. To learn about these programs, please refer to the help menus of each software.



8. References

i. TIME Project : http://unesdoc.unesco.org/images/0012/001223/122367eb.pdf

ii. Tunami Manual: http://tunamin2.ce.metu.edu.tr/

iii. Tunami N2 Manual: http://www.tsunami.civil.tohoku.ac.jp/hokusai3/J/projects/manual-ver-3.1.pdf

iv. Numerical method of tsunami simulation with the leap-frog scheme : www.vliz.be/imisdocs/publications/269372.pdf

v. Tunami FF: https://drive.google.com/drive/folders/0B1g9W4W9vKmsOWVQSkxNOW42M00?usp=sharing

vi. Terra Grid Cloud

Appendix -A

Mapping of CUDA Kernels from TIME Project

Computation of the equation of continuity
Open sea, boundary condition
Computation of the equation of motion

<u>Implementation of CUDA Kernels</u>

Computation of the equation of continuity MASS

VARIABLES
Water Level z
Discharge In I-direction fM
Discharge In J-direction fN
Area where the tsunami exists iMin, jMin, iMax, jMax and the computation is carried out
Highest water level zMax
zoutZeroThreshold

COEFFICIENTS:
Coefficients given cR1 and cR6=COS (THETA fM+1/2)

Table: Showing Mapping of ***MASS*** Subroutine from TIME Project to CUDA Kernel

	From Time Project	CUDA Implementation
	Computation of the equation of continuity	

	SUBROUTINE MASS
	(IG,JG,IS,JS,IE,JE,Z,M,N,R1,R6)
	C
	REAL M,N
	DIMENSION
	Z(IG,JG),M(IG,JG),N(IG,JG)
	DIMENSION R1(IG,JG),R6(JG)
	C
	DO 10 J=JS,JE
	DO 10 I=IS,IE
	Z(I,J)=Z(I,J)-R1(I,J)*(M(I,J)-M(I-1,J))
	& -
	R1(I,J)*(N(I,J)*R6(J)-N(I,J-1)*R6(J-1))
	10 CONTINUE
	
	DO 20 J=JS,JE
	DO 20 I=IS,IE
	IF(ABS(Z(I,J)).LT.1.0E-5)Z(I,J)=0.0
	20 CONTINUE
	RETURN
		__global__ void runMASSKernel(KernelData data) {
		Params& dp = data.params;
		
		int i = blockIdx.y * blockDim.y + threadIdx.y +
		dp.iMin;
		int j = blockIdx.x * blockDim.x + threadIdx.x +
		dp.jMin;
		int ij = data.idx(i,j);
		float absH;
		
		if(i <= dp.iMax && j <= dp.jMax && data.d[ij] != 0)
		{
		
		float zz = data.z[ij] - data.cR1[ij] * (
		data.fM[ij] - data.fM[data.le(ij)] + data.fN[ij] *
		data.cR6[j] - data.fN[data.dn(ij)]*data.cR6[j-1]);
		
		absH = fabs(zz);
		
		if(absH < dp.zoutZeroThreshold) {

	END	zz = 0.f;
		} else if(zz > data.zMax[ij]) {
		data.zMax[ij] = zz;
		//zMax[ij] = fmaxf(zMax[ij],h[ij]);
		}
		
		if(dp.zoutArrivalThreshold && data.tArr[ij] < 0 && absH > dp.zoutArrivalThreshold)
		data.tArr[ij] = dp.mTime;
		
		data.z[ij] = zz;
		}
		
		}

<u>Implementation of CUDA Kernels</u>

Computation of the equation of motion MOMENT

VARIABLES
Water Level z
Discharge In I-direction fM
Discharge In J-direction fN
Area where the tsunami exists iMin, jMin, iMax, jMax and the computation is carried out

COEFFICIENTS:
Coefficients given cR2 and cR4

Table: Showing Mapping of ***MOMENT*** Subroutine from TIME Project to CUDA Kernel

	From Time Project	CUDA Implementation
	Computation of the equation of motion MOMENT	

	SUBROUTINE MOMENT (IG,JG,IS,JS,IE,JE,
	& Z,M,N,R2,R3,R4,R5,
	& V1,V2)
	C
	REAL M,N
	DIMENSION Z(IG,JG),M(IG,JG),N(IG,JG)
	DIMENSION R2(IG,JG),R3(IG,JG),R4(IG,JG),R5(IG,JG)
	DIMENSION V1(JG),V2(JG)
	C
	DO 10 J=JS,JE
	DO 10 I=IS,IE
		__global__ void runMOMENTKernel(KernelData data) {
		
		Params& dp = data.params;
		
		int i = blockIdx.y * blockDim.y + threadIdx.y + dp.iMin;
		int j = blockIdx.x * blockDim.x + threadIdx.x + dp.jMin;
		int ij = data.idx(i,j);
		
		if(i <= dp.iMax && j <= dp.jMax && data.d[ij] != 0) {

<table>
<tr>
<td></td>
<td>

```
        V1(I)=Z(I+1,J)-Z(I,J)
        V2(I)=N(I,J-1)+N(I,J)+N(I+1,J-
1)+N(I+1,J)
        M(I,J)=M(I,J)-
R2(I,J)*V1(I)+R3(I,J)*V2(I)
  10 CONTINUE
        IF(JS.LE.2)THEN
        DO 15 I=1,IG-1
  15 M(I,1)=M(I,1)-R2(I,1)*(Z(I+1,1)-Z(I,1))
        ENDIF
        IF(JE.GE.JG-1)THEN
        DO 16 I=1,IG-1
  16 M(I,JG)=M(I,JG)-R2(I,JG)*(Z(I+1,JG)-
Z(I,JG))
        ENDIF
        IF (IS.LE.2)THEN
        DO 17 J=1,JG
  17 M(1,J)=M(I,J)-R2(1,J)*(Z(2,J)-Z(1,J))
        ENDIF
C
        DO 20 J=JS,JE
        DO 20 I=IS,IE
        V1(I)=Z(I,J+1)-Z(I,J)
        V2(I)=M(I-1,J)+M(I,J)+M(I-
1,J+1)+M(I,J+1)
        N(I,J)=N(I,J)-R4(I,J)*V1(I)-
R5(I,J)*V2(I)
  20 CONTINUE
        IF(IS.LE.2)THEN
        DO 25 J=1,JG-1
  25 N(1,J)=N(1,J)-R4(1,J)*(Z(1,J+1)-Z(1,J))
        ENDIF
        IF(IE.GE.IG-1)THEN
        DO 26 J=1,JG-1
  26 N(IG,J)=N(IG,J)-R4(IG,J)*(Z(IG,J+1)-
Z(IG,J))
        ENDIF
        IF(JS.LE.2)THEN
        DO 27 I=1,IG
  27 N(I,1)=N(I,1)-R4(I,1)*(Z(I,2)-Z(I,1))
        ENDIF
C
        RETURN
        END
```

</td>
<td>

```
        float zz = data.z[ij];

        if( data.d[data.ri(ij)] != 0 ) {
                data.fM[ij] = data.fM[ij] -
data.cR2[ij]*(data.z[data.ri(ij)] - zz);
        }

        if( data.d[data.up(ij)] != 0 )
                data.fN[ij] = data.fN[ij] -
data.cR4[ij]*(data.z[data.up(ij)] - zz);

    }

}
```

</td>
</tr>
</table>

<u>Implementation of CUDA Kernels</u>

Open sea, boundary condition GBOUND

VARIABLES

Water Level	z
Discharge In I-direction	fM
Discharge In J-direction	fN

COEFFICIENTS:

Coefficients given

(THETA M+1/2)in radian	= cB1
(THETA M)in radian	= cB2
(THETA M-1/2)in radian	= cB3
coefficent	= cB4

Table: Showing Mapping of ***GBOUND*** Subroutine from TIME Project to CUDA Kernel

	From Time Project	CUDA Implementation
	Open sea, boundary condition GBOUND	

	SUBROUTINE GBOUND (IG,JG,IS,JS,IE,JE,
	&
	Z,M,N,C1,C2,C3,C4)
	C
	REAL M,N
	DIMENSION Z(IG,JG),M(IG,JG),N(IG,JG)
	DIMENSION C1(IG),C2(JG),C3(IG),C4(JG)
	C
	IF(JS.LE.2)THEN
	DO 10 I=2,IG-1
	Z(I,1)=SQRT(N(I,1)**2
	&+0.25*(M(I,1)+M(I-1,1))**2)*C1(I)
	10 IF(N(I,1).GT.0.)Z(I,1)=-Z(I,1)
	ENDIF
	IF(IS.LE.2)THEN
	DO 20 J=2,JG-1
	Z(1,J)=SQRT(M(I,J)**2
	&+0.25*(N(1,J)+N(1,J-1))**2)*C2(J)
	C IF(Z(1,J).GT.1.0)Z(1,J)=1.0
	C IF(Z(1,J).LT.-1.0)Z(1,J)=-1.0
	20 IF(M(1,J).GT.0.)Z(1,J)=-Z(1,J)
	ENDIF
	IF(JE.GE.JG-1)THEN
	DO 30 I=2,IG-1
	Z(I,JG)=SQRT(N(I,JG-1)**2
	&+0.25*(M(I,JG)+M(I,JG-1))**2)*C3(I)
	IF(N(I,JG-1).LT.0.)Z(I,JG)=-Z(I,JG)
	30 CONTINUE
	C
		__global__ void runGBOUNDKernel(KernelData data) {
		KernelData& dt = data;
		Params& dp = data.params;
		
		int id = blockIdx.x * blockDim.x + threadIdx.x + 2;
		int ij;
		
		if(id <= dp.nI-1) {
		ij = dt.idx(id,1);
		dt.z[ij] = sqrtf(powf(dt.fN[ij],2.0f) + 0.25f*powf((dt.fM[ij] + dt.fM[dt.le(ij)]),2.0f))*dt.cB1[id-1];
		if(dt.fN[ij] > 0) dt.z[ij] = -dt.z[ij];
		}
		
		if(id <= dp.nI-1) {
		ij = dt.idx(id,dp.nJ);
		dt.z[ij] = sqrtf(powf(dt.fN[dt.dn(ij)],2.0f) + 0.25f*powf((dt.fM[ij] + dt.fM[dt.dn(ij)]),2.0f))*dt.cB3[id-1];
		if(dt.fN[dt.dn(ij)] < 0) dt.z[ij] = -dt.z[ij];
		}
		
		if(id <= dp.nJ-1) {
		ij = dt.idx(1,id);
		dt.z[ij] = sqrtf(powf(dt.fM[ij],2.0f) +

```
      ENDIF
      IF(IE.GE.IG-1)THEN
      DO 40 J=2,JG-1
      Z(IG,J)=SQRT(M(IG-1,J)**2
     &+0.25*(N(IG,J)+N(IG,J-1))**2)*C4(J)
   40 IF(M(IG-1,J).LT.0.)Z(IG,J)=-Z(IG,J)
      ENDIF
      Z(1,1)=SQRT(M(1,1)**2+N(1,1)**2)*C1(1)
      IF(N(1,1).GT.0.0)Z(1,1)=-Z(1,1)
      Z(IG,1)=SQRT(M(IG-1,1)**2+N(IG,1)**2)*C1(IG)
      IF(N(IG,1).GT.0.0)Z(IG,1)=-Z(IG,1)
      Z(1,JG)=SQRT(M(1,JG)**2+N(1,JG-1)**2)*C3(1)
      IF(N(1,JG-1).LT.0.0)Z(1,JG)=-Z(1,JG)
      Z(IG,JG)=SQRT(M(IG-1,JG)**2+N(IG,JG-1)**2)*C3(IG)
      IF(N(IG,JG-1).LT.0.0)Z(IG,JG)=-Z(IG,JG)
      RETURN
      END
C
```

```
0.25f*powf((dt.fN[ij] + dt.fN[dt.dn(ij)]),2.0f) )*dt.cB2[id-1];
        if( dt.fM[ij] > 0 ) dt.z[ij] = -dt.z[ij];
    }

    if( id <= dp.nJ-1 ) {
        ij = dt.idx(dp.nI,id);
        dt.z[ij] = sqrtf( powf(dt.fM[dt.le(ij)],2.0f) + 0.25f*powf((dt.fN[ij] + dt.fN[dt.dn(ij)]),2.0f) )*dt.cB4[id-1];
        if( dt.fM[dt.le(ij)] < 0 ) dt.z[ij] = -dt.z[ij];
    }

    if( id == 2 ) {
        ij = dt.idx(1,1);
        dt.z[ij] = sqrtf( powf(dt.fM[ij],2.0f) + powf(dt.fN[ij],2.0f) )*dt.cB1[0];
        if( dt.fN[ij] > 0 ) dt.z[ij] = -dt.z[ij];

        ij = dt.idx(dp.nI,1);
        dt.z[ij] = sqrtf( powf(dt.fM[dt.le(ij)],2.0f) + powf(dt.fN[ij],2.0f) )*dt.cB1[dp.nI-1];
        if( dt.fN[ij] > 0 ) dt.z[ij] = -dt.z[ij];

        ij = dt.idx(1,dp.nJ);
        dt.z[ij] = sqrtf( powf(dt.fM[ij],2.0f) + powf(dt.fN[dt.dn(ij)],2.0f) )*dt.cB3[0];
        if( dt.fN[dt.dn(ij)] < 0 ) dt.z[ij] = -dt.z[ij];

        ij = dt.idx(dp.nI,dp.nJ);
        dt.z[ij] = sqrtf( powf(dt.fM[dt.le(ij)],2.0f) + powf(dt.fN[dt.dn(ij)],2.0f) )*dt.cB3[dp.nI-1];
        if( dt.fN[dt.dn(ij)] < 0 ) dt.z[ij] = -dt.z[ij];
    }
}
```

<u>Implementation of CUDA Kernels</u>

ALIMT BLIMT

Making area of computation be with in the area under consideration as well as
Enlargement of the area of computation as the tsunami propagates

VARIABLES

Water Level	z
Area where the tsunami exists and the computation is carried out	iMin, jMin, iMax, jMax

zoutClipThreshold, x, y, w

Table: Showing Mapping of **ALIMT, BLIMIT** Subroutine from TIME Project to CUDA Kernel

	From Time Project	CUDA Implementation
	Making area of computation and Enlargement as tsunami propgates ALIMT, BLIMIT	

	SUBROUTINE ALIMIT (IG,JG,IS,JS,IE,JE,IR,JR,ID,JD)
	C
	IS=IR-1
	JS=JR-1
	IE=ID+1
	JE=JD+1
	IF(IS.LE.2)IS=2
	IF(JS.LE.2)JS=2
	IF(IE.GE.IG-1)IE=IG-1
	IF(JE.GE.JG-1)JE=JG-1
	WRITE(6,100)IS,JS,IE,JE
	RETURN
	100 FORMAT(1H,"IS=",I5,2X,"JS=",I5,2X,"IE=",I5,2X,"JE=",I5)
	END
	
	SUBROUTINE BLIMIT (IG,JG,IS,JS,IE,JE,Z)
	C
	PARAMETER (GX=1.E-4)
	DIMENSION Z(IG,JG)
	C
	IF(IS.EQ.2)GOTO 61
	L=0
	DO 10 J=JS,JE
	IF(ABS(Z(IS+2,J)).GT.GX)L=1
	10 CONTINUE
	IF(L.EQ.1)THEN
	IS=IS-1
	IF(IS.LE.2)IS=2
	ENDIF
	61 IF(IE.EQ.IG-1)GOTO 62
	L=0
	DO 20 J=JS,JE
	IF(ABS(Z(IE-2,J)).GT.GX)L=1
	20 CONTINUE
	IF(L.EQ.1)THEN
	IE=IE+1
	IF(IE.GE.IG-1)IE=IG-1
	ENDIF
		__global__ void runAlimitBlimitGridKernel(KernelData data) {
		
		Params& dp = data.params;
		
		int id = blockIdx.x * blockDim.x + threadIdx.x + 1;
		
		#if (__CUDA_ARCH__ >= 130)
		
		if(id >= dp.jMin && id <= dp.jMax) {
		
		if(fabsf(data.z[data.idx(dp.iMin+2,id)]) > dp.zoutClipThreshold)
		atomicAdd(&(data.g_MinMax->x), 1);
		
		if(fabsf(data.z[data.idx(dp.iMax-2,id)]) > dp.zoutClipThreshold)
		atomicAdd(&(data.g_MinMax->y), 1);
		}
		
		if(id >= dp.iMin && id <= dp.iMax) {
		
		if(fabsf(data.z[data.idx(id,dp.jMin+2)]) > dp.zoutClipThreshold)
		atomicAdd(&(data.g_MinMax->z), 1);
		
		if(fabsf(data.z[data.idx(id,dp.jMax-2)]) > dp.zoutClipThreshold)
		atomicAdd(&(data.g_MinMax->w), 1);

	62 IF(JS.EQ.2)GOTO 63
	L=0
	DO 30 I=IS,IE
	IF(ABS(Z(I,JS+2)).GT.GX)L=1
	30 CONTINUE
	IF(L.EQ.1)THEN
	JS=JS-1
	IF(JS.LE.2)JS=2
	ENDIF
	63 IF(JE.EQ.JG-1)GOTO 64
	L=0
	DO 40 I=IS,IE
	IF(ABS(Z(I,JE-2)).GT.GX)L=1
	40 CONTINUE
	IF(L.EQ.1)THEN
	JE=JE+1
	IF(JE.GE.JG-1)JE=JG-1
	ENDIF
	64 RETURN
	END
		}
		
		#else
		
		if(id == 1) {
		
		for(int j = dp.jMin; j <= dp.jMax; j++
		) {
		
		if(
		fabsf(data.z[data.idx(dp.iMin+2,j)]) >
		dp.zoutClipThreshold) {
		data.g_MinMax->x = 1;
		break;
		}
		}
		for(int j = dp.jMin; j <= dp.jMax; j++
		) {
		
		if(fabsf(data.z[data.idx(dp.iMax-
		2,j)]) > dp.zoutClipThreshold) {
		data.g_MinMax->y = 1;
		break;
		}
		
		}
		
		for(int i = dp.iMin; i <= dp.iMax; i++
		) {
		if(
		fabsf(data.z[data.idx(i,dp.jMin+2)]) >
		dp.zoutClipThreshold) {
		data.g_MinMax->z = 1;
		break;
		}
		}
		for(int i = dp.iMin; i <= dp.iMax; i++
		) {
		
		if(fabsf(data.z[data.idx(i,dp.jMax-
		2)]) > dp.zoutClipThreshold) {
		data.g_MinMax->w = 1;
		break;
		}
		}
		}
		#endif
		}

Appendix -B

GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

Mathematical Equations (Latex)

Equation 1: \frac{\partial \eta}{\partial t} + \frac{1}{R \cos \theta } \begin{bmatrix} \frac{\partial M}{\partial \lambda } + \frac{\partial}{\partial \theta}(N \cos \theta))\end{bmatrix} = 0

Equation2: \frac{\partial M}{\partial t} + \frac{gh}{R \cos \theta } \frac{\partial \eta}{\partial \lambda } = fN

Equation3: \frac{\partial N}{\partial t} + \frac{gh}{R} \frac{\partial \eta}{\partial \theta } = -fM

Equation4: \frac {\eta_{j,m}^{n+1/2} - \eta_{j,m}^{n-1/2}} {\Delta t} + \frac{1}{R \cos \theta_m } \begin{bmatrix} \frac {M_{j+1/2,m}^{n} - M_{j-1/2,m}^{n}} {\Delta \lambda } + \frac {N_{j,m+1/2}^{n} \cos \theta_{m+1/2} - N_{j,m-1/2}^{n} \cos \theta_{m-1/2}} {\Delta \lambda }\end{bmatrix} = 0

Equation 5: \frac {M_{j+1/2,m}^{n} - M_{j-1/2,m}^{n}} {\Delta t } + \frac {gh_{j+1/2,m}}{R \cos \theta_m} \frac { \eta_{j+1,m}^{n+1/2}- \eta_{j,m-1/2}^{n+1/2} } {\Delta \lambda } = fN^1

Equation 6: \frac {N_{j+1/2,m}^{n+1} - N_{j+1/2,m}^{n}} {\Delta t } + \frac {gh_{j+1/2,m}}{R \sin \theta_m} \frac { \eta_{j,m+1}^{n+1/2}- \eta_{j,m}^{n+1/2} } {\Delta \theta } = fM^1

URL: https://www.codecogs.com/latex/eqneditor.php

start;

Input of water depth

and initial profile;

initial condition:

Still water level;

(check of the area

for computation)

{

Equation of continuity;

Open sea boundary condition;

Equation of motion;

check of the area

for computation;

```
}
// False
if (K>KE)
output;
else
```