NON-HYDROSTATIC WAVE MODEL NHWAVE

USER'S GUIDE FOR MODELING SUBMARINE LANDSLIDE TSUNAMI

(VERSION 1.1)

BY

FENGYAN SHI, JAMES T. KIRBY, GANGFENG MA

AND BABAK TEHRANIRAD

RESEARCH REPORT NO. CACR-12-04
MAY 2012

CENTER FOR APPLIED COASTAL RESEARCH

University of Delaware
Newark, Delaware 19716

Abstract

This report provides a user's guide for operation of the non-hydrostatic wave model NHWAVE in application to generation of submarine landslide-induced tsunami waves. NHWAVE was developed by Ma et al. (2012) for modeling fully dispersive surface wave processes. It solves the non-hydrostatic Navier-Stokes equations in a domain over a surface and terrain in the sigma coordinate system. Tsunami waves generated by a prescribed submarine landslide can be directly simulated by taking advantage of the σ coordinate system. The model assumes a single-valued water surface and represents turbulent stresses in terms of an eddy viscosity closure scheme. For tsunami wave applications in the present report, turbulent stresses are not modeled, and thus the model is basically solving the Euler equations for incompressible flow with a moving surface and bottom.

NHWAVE has been benchmarked in Tehranirad et al. (2012) for tsunami application using PMEL-135 benchmarks provided by Synolakis et al. (2007), and a landslide benchmark developed from results in Enet and Grilli (2007). This report provides a user's manual for landslide applications only. A brief description of model theory, numerical scheme and landslide configuration is also included in the report. The model setup and results presented here represent testing of Version 1.1 of the code, and will be updated online at http://chinacat.coastal.udel.edu/programs/ with each version change for the publicly distributed code.

Contents

List of Figures

1 Introduction

The non-hydrostatic wave model NHWAVE was developed by Ma et al. (2012) for modeling fully dispersive surface wave processes from deep water to coastal region. It solves the non-hydrostatic Navier-Stokes equations in a domain over a surface and terrain in the σ coordinate system. The Smagorinsky subgrid turbulence model and $k-\epsilon$ turbulence model are implemented in NHWAVE. The model takes wave breaking and moving shorelines into account. For tsunami wave applications presented in this report, turbulent stresses, wave breaking and moving shoreline are not modeled, and thus the model is basically solving the Euler equations for incompressible flow with moving surface and bottom. Numerically, it uses a Godunov-type finite volume scheme to represent spatial derivatives, with time stepping done using a Runge-Kutta scheme. The model allows for a prescribed time-dependent bottom motion, making it directly applicable to the simulation of landslide-induced tsunami wave generation. The detailed description of the model can be found in Ma et al. (2012).

Tehranirad et al. (2012) reported the benchmark testing of NHWAVE for tsunami runup application using PMEL-135 benchmarks provided by Synolakis et al (2007). This set of benchmarks are the presently accepted benchmarking standards adopted by the National Tsunami Hazard Mitigation Program (NTHMP) for judging model acceptance for use in development of coastal inundation maps and evacuation plans. In addition, benchmark tests for landslide generated tsunami waves were also carried out in Tehranirad et al. using the laboratory experimental data provided by Enet and Grilli (2007). This user's guide uses the case of Enet and Grilli (2007) as one example to demonstrate the model setup process.

The report is organized as follows. Sections 2 and 3 provide brief descriptions of the model equations and numerical techniques, respectively, proposed by Ma et al. (2012). Section 4 introduces an idealized solid slide used in both Enet and Grilli (2007) and a field-scale application. Section 5 describes the model operation procedures. Two examples including a laboratory case and a field-scale case are demonstrated in Section 6.

NHWAVE is distributed as open source code. General users may obtain the most recent tested version from the web site http://chinacat.coastal.udel.edu/programs/index.html which provides this code along with other programs developed at the Center for Applied Coastal Research. The code is provided along with a unix/linux makefile and input files for executing the tests described in the manual. The present report will also be updated with each major change in program version.

Version control for NHWAVE is managed using Subversion (Collins-Sussman et al, 2004). Users who would like to become part of the development team should contact Fengyan Shi (fyshi@udel.edu) or Jim Kirby (kirby@udel.edu).

2 Governing Equations

2.1 Navier-Stokes equations

The incompressible Navier-Stokes equations in Cartesian coordinates (x_1^*, x_2^*, x_3^*), where $x_1^* = x^*$, $x_2^* = y^*$ and $x_3^* = z^*$ and time t^* are given by

$$\frac{\partial u_i}{\partial x_i^*} = 0 \tag{1}$$

$$\frac{\partial u_i}{\partial t^*} + u_j \frac{\partial u_i}{\partial x_j^*} = -\frac{1}{\rho}\frac{\partial \tilde{p}}{\partial x_i^*} + g_i + \frac{\partial \tau_{ij}}{\partial x_j^*} \tag{2}$$

where $(i, j) = 1, 2, 3$, u_i is velocity component in the x_i^* direction, $\tilde{p}$ is total pressure, ρ is water density, $g_i = -g\delta_{i3}$ is the gravitational body force and $\tau_{ij} = \nu_t(\partial u_i/\partial x_j^* + \partial u_j/\partial x_i^*)$ is turbulent stress with ν_t the turbulent kinematic viscosity. Equations (1) and (2) are augmented by kinematic constraints given at the surface and bottom boundaries given by

$$\frac{\partial \eta}{\partial t^*} + u\frac{\partial \eta}{\partial x^*} + v\frac{\partial \eta}{\partial y^*} = w; \qquad z^* = \eta \tag{3}$$

$$\frac{\partial h}{\partial t^*} + u\frac{\partial h}{\partial x^*} + v\frac{\partial h}{\partial y^*} = -w; \qquad z^* = -h \tag{4}$$

where η and h are single valued functions of (x^*, y^*, t^*), and by appropriate dynamic constraints.

2.2 Governing equations in σ coordinate system

In order to accurately represent bottom and surface geometry, a σ coordinate transformation developed by Phillips (1957) is used in NHWAVE. The coordinate transformation maps the bottom and surface onto constant boundaries of a strip of unit thickness. The transformation is given by

$$t = t^* \quad x = x^* \quad y = y^* \quad \sigma = \frac{z^* + h}{D} \tag{5}$$

where $D = h + \eta$. Using the chain rule, the partial derivatives of a variable $f = f(x^*, y^*, z^*, t^*)$ in the physical domain are transformed as follows.

$$\begin{aligned}
\frac{\partial f}{\partial t^*} &= \frac{\partial f}{\partial t} + \frac{\partial f}{\partial \sigma}\frac{\partial \sigma}{\partial t^*} \\
\frac{\partial f}{\partial x^*} &= \frac{\partial f}{\partial x} + \frac{\partial f}{\partial \sigma}\frac{\partial \sigma}{\partial x^*} \\
\frac{\partial f}{\partial y^*} &= \frac{\partial f}{\partial y} + \frac{\partial f}{\partial \sigma}\frac{\partial \sigma}{\partial y^*} \\
\frac{\partial f}{\partial z^*} &= \frac{\partial f}{\partial \sigma}\frac{\partial \sigma}{\partial z^*}
\end{aligned} \tag{6}$$

We obtain the governing equations in the new coordinate system (x, y, σ, t) using (5-6) in (1) and (2). The continuity equation (1) is first transformed as

$$\frac{\partial u}{\partial x} + \frac{\partial u}{\partial \sigma}\frac{\partial \sigma}{\partial x^*} + \frac{\partial v}{\partial y} + \frac{\partial v}{\partial \sigma}\frac{\partial \sigma}{\partial y^*} + \frac{1}{D}\frac{\partial w}{\partial \sigma} = 0 \tag{7}$$

Using the results

$$\begin{aligned}
\frac{\partial \sigma}{\partial t^*} &= \frac{1}{D}\frac{\partial h}{\partial t} - \frac{\sigma}{D}\frac{\partial D}{\partial t} \\
\frac{\partial \sigma}{\partial x^*} &= \frac{1}{D}\frac{\partial h}{\partial x} - \frac{\sigma}{D}\frac{\partial D}{\partial x} \\
\frac{\partial \sigma}{\partial y^*} &= \frac{1}{D}\frac{\partial h}{\partial y} - \frac{\sigma}{D}\frac{\partial D}{\partial y} \\
\frac{\partial \sigma}{\partial z^*} &= \frac{1}{D}
\end{aligned} \tag{8}$$

we rewrite (7) as

$$\frac{\partial D}{\partial t} + \frac{\partial Du}{\partial x} + \frac{\partial Dv}{\partial y} + \frac{\partial \omega}{\partial \sigma} = 0 \tag{9}$$

where ω is the vertical velocity relative to constant σ surfaces, given by

$$\omega = D\left(\frac{\partial \sigma}{\partial t^*} + u\frac{\partial \sigma}{\partial x^*} + v\frac{\partial \sigma}{\partial y^*} + w\frac{\partial \sigma}{\partial z^*}\right) \tag{10}$$

The transformed continuity equation (9) may be integrated over depth to obtain

$$\frac{\partial D}{\partial t} + \frac{\partial}{\partial x}(D\int_0^1 u d\sigma) + \frac{\partial}{\partial y}(D\int_0^1 v d\sigma) = 0 \tag{11}$$

where the kinematic constraints (3) - (4) have been used. Equation (11) is used subsequently to determine the surface position.

The transformed momentum equations may be written as

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} + \frac{\partial \mathbf{H}}{\partial \sigma} = \mathbf{S}_h + \mathbf{S}_p + \mathbf{S}_\tau \tag{12}$$

where $\mathbf{U} = (Du, Dv, Dw)^T$. The fluxes are given by

$$\mathbf{F} = \begin{pmatrix} Duu + \frac{1}{2}gD^2 \\ Duv \\ Duw \end{pmatrix} \quad \mathbf{G} = \begin{pmatrix} Duv \\ Dvv + \frac{1}{2}gD^2 \\ Dvw \end{pmatrix} \quad \mathbf{H} = \begin{pmatrix} u\omega \\ v\omega \\ w\omega \end{pmatrix} \tag{13}$$

The source terms are given by

$$\mathbf{S}_h = \begin{pmatrix} gD\frac{\partial h}{\partial x} \\ gD\frac{\partial h}{\partial y} \\ 0 \end{pmatrix} \quad \mathbf{S}_p = \begin{pmatrix} -\frac{D}{\rho}(\frac{\partial p}{\partial x} + \frac{\partial p}{\partial \sigma}\frac{\partial \sigma}{\partial x^*}) \\ -\frac{D}{\rho}(\frac{\partial p}{\partial y} + \frac{\partial p}{\partial \sigma}\frac{\partial \sigma}{\partial y^*}) \\ -\frac{1}{\rho}\frac{\partial p}{\partial \sigma} \end{pmatrix} \quad \mathbf{S}_\tau = \begin{pmatrix} DS_{\tau_x} \\ DS_{\tau_y} \\ DS_{\tau_z} \end{pmatrix} \tag{14}$$

where the total pressure $\tilde{p} = p + \rho g(\eta - z^*)$ has been divided into it's dynamic (p) and hydrostatic ($\rho g(\eta - z^*)$) parts. It has been noted in a number of previous studies that the application of standard finite volume Godunov-type scheme directly to equations of this form does not lead to an automatic preservation of steady state (Zhou et al., 2001; Kim et al, 2008; Liang and Marche, 2009). Therefore, It is desirable to reformulate the equations so that the flux and source terms can be automatically balanced at the discrete level in the steady state. Following Shi et al. (2012), the source term is rewritten as

$$g(h+\eta)\frac{\partial h}{\partial x} = \frac{\partial}{\partial x}\left(\frac{1}{2}gh^2\right) + g\eta\frac{\partial h}{\partial x} \tag{15}$$

in which the first term in the right hand side can be combined together with the flux terms. Based on this, the flux terms $\mathbf{F}$ and $\mathbf{G}$ and source term $\mathbf{S_h}$ are rewritten as

$$\mathbf{F} = \begin{pmatrix} Duu + \frac{1}{2}g\eta^2 + gh\eta \\ Duv \\ Duw \end{pmatrix} \quad \mathbf{G} = \begin{pmatrix} Duv \\ Dvv + \frac{1}{2}g\eta^2 + gh\eta \\ Dvw \end{pmatrix} \quad \mathbf{S}_h = \begin{pmatrix} g\eta\frac{\partial h}{\partial x} \\ g\eta\frac{\partial h}{\partial y} \\ 0 \end{pmatrix} \tag{16}$$

The main advantage of the above formulation is that the flux and source terms are well-balanced so that no artificial flow due to bottom slope will be generated.

For the present landslide tsunami application, turbulent diffusion terms $S_{\tau_x}, S_{\tau_y}, S_{\tau_z}$ are neglected.

3 Numerical method

3.1 Grid configuration

A combined finite-volume and finite-difference scheme with a Godunov-type method was applied to discretize equations (9) and (12). It is straightforward to define all dependent variables at cell centers to solve the Riemann problem. However, this treatment results in checkerboard solutions in which the pressure and velocity become decoupled when they are defined at the same location (Patankar, 1980). Therefore, most existing models use a staggered grid in which the pressure is defined at the centers of computational cells and the velocities are defined at cell faces (Bradford, 2005). However, staggered grids do not lend themselves as easily as co-located grids to the use of Godunov-type schemes. Meanwhile, difficulty in treating the cell-centered pressure at the top layer may arise when applying the pressure boundary condition at the free surface (Yuan and Wu, 2004).

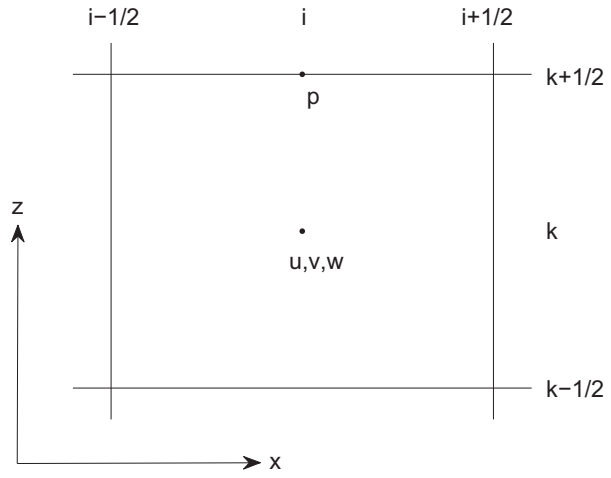

Figure 1: Layout of computational variables. Velocities (u, v, w) are placed at cell center and dynamic pressure p is defined at vertical cell face.

With these considerations, a different kind of staggered grid framework is introduced, in which the velocities are placed at the cell centers and the pressure is defined at the vertically-facing cell faces as shown in figure 1. The momentum equations are solved by a second-order Godunov-type finite volume method. The HLL approximate Riemann solver (Harten et al., 1983) is used to estimate fluxes at the cell faces. As in Stelling and Zijlema (2003), the pressure boundary condition at the free surface can be precisely assigned to zero.

3.2 Time Stepping

To obtain second-order temporal accuracy, the two-stage second-order nonlinear Strong Stability-Preserving (SSP) Runge-Kutta scheme (Gottlieb et al., 2001) is adopted for time stepping. At the first stage, an intermediate quantity $\mathbf{U}^{(1)}$ is evaluated using a typical first-order, two-step projection method given by

$$\frac{\mathbf{U}^* - \mathbf{U}^n}{\Delta t} = -\left(\frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} + \frac{\partial \mathbf{H}}{\partial \sigma}\right)^n + \mathbf{S}_h^n + \mathbf{S}_\tau^n \tag{17}$$

$$\frac{\mathbf{U}^{(1)} - \mathbf{U}^*}{\Delta t} = \mathbf{S}_p^{(1)} \tag{18}$$

where $\mathbf{U}^n$ represents $\mathbf{U}$ value at time level n, $\mathbf{U}^*$ is the intermediate value in the two-step projection method, and $\mathbf{U}^{(1)}$ is the final first stage estimate. In the second stage, the velocity field is again updated to a second intermediate level using the same projection method, after which the Runge-

Kutta algorithm is used to obtain a final value of the solution at the $n+1$ time level.

$$\frac{\mathbf{U}^* - \mathbf{U}^{(1)}}{\Delta t} = -\left(\frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} + \frac{\partial \mathbf{H}}{\partial \sigma}\right)^{(1)} + \mathbf{S}_h^{(1)} + \mathbf{S}_\tau^{(1)} \tag{19}$$

$$\frac{\mathbf{U}^{(2)} - \mathbf{U}^*}{\Delta t} = \mathbf{S}_p^{(2)} \tag{20}$$

$$\mathbf{U}^{n+1} = \frac{1}{2}\mathbf{U}^n + \frac{1}{2}\mathbf{U}^{(2)} \tag{21}$$

Each stage of the calculation requires the specification of the nonhydrostatic component of the pressure force as expressed through the quantities $\mathbf{S}_p^{(1,2)}$. The pressure field needed to specify these is based on the solution of the Poisson equation described below. Also at each stage, the surface elevation is obtained by solving equation (11) explicitly. The time step Δt is adaptive during the simulation, following the Courant-Friedrichs-Lewy (CFL) criterion

$$\Delta t = C\min\left[\min\frac{\Delta x}{|u_{i,j,k}| + \sqrt{gD_{i,j}}}, \min\frac{\Delta y}{|v_{i,j,k}| + \sqrt{gD_{i,j}}}, \min\frac{\Delta \sigma D_{i,j}}{|w_{i,j,k}|}\right] \tag{22}$$

where C is the Courant number, which is taken to be 0.5 to ensure accuracy and stability in the current model.

3.3 Spatial finite volume scheme

Equation (9) and (12) are discretized using a second-order Godunov-type finite volume method. To solve equation (9) and (12), fluxes based on the conservative variables are required at the cell faces. In high-order Godunov-type methods, the values of the conservative variables within a cell are calculated using a reconstruction method based on the cell center data (Zhou et al., 2001). Usually a piecewise linear reconstruction is used, leading to a second order scheme. For $\mathbf{U}$ in the cell i, we have

$$\mathbf{U} = \mathbf{U}_i + (x - x_i)\Delta \mathbf{U}_i \tag{23}$$

where $\Delta \mathbf{U}_i$ is the gradient of $\mathbf{U}$, which is calculated by

$$\Delta \mathbf{U}_i = \text{avg}\left(\frac{\mathbf{U}_{i+1} - \mathbf{U}_i}{x_{i+1} - x_i}, \frac{\mathbf{U}_i - \mathbf{U}_{i-1}}{x_i - x_{i-1}}\right) \tag{24}$$

in which avg is a slope limiter which is used to avoid spurious oscillations in the reconstruction data at the cell faces. In NHWAVE, the van Leer limiter is adopted, which is given by

$$\text{avg}(a, b) = \frac{a|b| + |a|b}{|a| + |b|} \tag{25}$$

The left and right values of $\mathbf{U}$ at cell face $(i + \frac{1}{2})$ are given by

$$\mathbf{U}^L_{i+\frac{1}{2}} = \mathbf{U}_i + \frac{1}{2}\Delta x_i \Delta \mathbf{U}_i \quad \mathbf{U}^R_{i+\frac{1}{2}} = \mathbf{U}_{i+1} - \frac{1}{2}\Delta x_{i+1}\Delta \mathbf{U}_{i+1} \tag{26}$$

The flux $\mathbf{F}(\mathbf{U}^L, \mathbf{U}^R)$ is calculated by solving a local Riemann problem at each horizontally-facing cell face. In the present study, HLL Riemann solver is employed. The flux at the cell interface $(i + \frac{1}{2})$ is determined by

$$\mathbf{F}(\mathbf{U}^L, \mathbf{U}^R) = \begin{cases} \mathbf{F}(\mathbf{U}^L) & \text{if} \quad s_L \geq 0 \\ \mathbf{F}^*(\mathbf{U}^L, \mathbf{U}^R) & \text{if} \quad s_L < 0 < s_R \\ \mathbf{F}(\mathbf{U}^R) & \text{if} \quad s_R \leq 0 \end{cases} \tag{27}$$

where

$$\mathbf{F}^*(\mathbf{U}^L, \mathbf{U}^R) = \frac{s_R \mathbf{F}(\mathbf{U}^L) - s_L \mathbf{F}(\mathbf{U}^R) + s_L s_R (\mathbf{U}^R - \mathbf{U}^L)}{s_R - s_L} \tag{28}$$

with wave speed s_L and s_R defined by

$$s_L = \min(u^L - \sqrt{gD_L}, u_s - \sqrt{gD_s}) \tag{29}$$

$$s_R = \max(u^R + \sqrt{gD_R}, u_s + \sqrt{gD_s}) \tag{30}$$

where u_s and $\sqrt{gD_s}$ are estimated by

$$u_s = \frac{1}{2}(u^L + u^R) + \sqrt{gD_L} - \sqrt{gD_R} \tag{31}$$

$$\sqrt{gD_s} = \frac{\sqrt{gD_L} + \sqrt{gD_R}}{2} + \frac{u^L - u^R}{4} \tag{32}$$

To obtain the non-hydrostatic velocity field, the dynamic pressure p has to be calculated first. From equation (18) and (20), we get

$$u^{(k)} = u^* - \frac{\Delta t}{\rho} \left(\frac{\partial p}{\partial x} + \frac{\partial p}{\partial \sigma} \frac{\partial \sigma}{\partial x^*} \right)^{(k)} \tag{33}$$

$$v^{(k)} = v^* - \frac{\Delta t}{\rho} \left(\frac{\partial p}{\partial y} + \frac{\partial p}{\partial \sigma} \frac{\partial \sigma}{\partial y^*} \right)^{(k)} \tag{34}$$

$$w^{(k)} = w^* - \frac{\Delta t}{\rho} \frac{1}{D^{(k)}} \frac{\partial p^{(k)}}{\partial \sigma} \tag{35}$$

where $k = 1, 2$ represents the kth stage in the Runge-Kutta integration.

Substituting equation (33) - (35) into the continuity equation (7), we obtain the Poisson equation in (x, y, σ) coordinate system, given by

$$\begin{aligned} &\frac{\partial}{\partial x} \left[\frac{\partial p}{\partial x} + \frac{\partial p}{\partial \sigma} \frac{\partial \sigma}{\partial x^*} \right] + \frac{\partial}{\partial y} \left[\frac{\partial p}{\partial y} + \frac{\partial p}{\partial \sigma} \frac{\partial \sigma}{\partial y^*} \right] + \frac{\partial}{\partial \sigma} \left(\frac{\partial p}{\partial x} \right) \frac{\partial \sigma}{\partial x^*} + \\ &\frac{\partial}{\partial \sigma} \left(\frac{\partial p}{\partial y} \right) \frac{\partial \sigma}{\partial y^*} + \left[\left(\frac{\partial \sigma}{\partial x^*} \right)^2 + \left(\frac{\partial \sigma}{\partial y^*} \right)^2 + \frac{1}{D^2} \right] \frac{\partial}{\partial \sigma} \left(\frac{\partial p}{\partial \sigma} \right) = \\ &\frac{\rho}{\Delta t} \left(\frac{\partial u^*}{\partial x} + \frac{\partial u^*}{\partial \sigma} \frac{\partial \sigma}{\partial x^*} + \frac{\partial v^*}{\partial y} + \frac{\partial v^*}{\partial \sigma} \frac{\partial \sigma}{\partial y^*} + \frac{1}{D} \frac{\partial w^*}{\partial \sigma} \right) \end{aligned} \tag{36}$$

The above equation is discretized using second-order space-centered finite differences. The velocities (u^*, v^*, w^*) at vertical cell faces are interpolated from adjacent cell-centered values. The resulting linear equation is given by

$$\begin{aligned} &a_1 p_{i,j-1,k-1} + a_2 p_{i-1,j,k-1} + a_3 p_{i,j,k-1} + a_4 p_{i+1,j,k-1} + a_5 p_{i,j+1,k-1} + \\ &a_6 p_{i,j-1,k} + a_7 p_{i-1,j,k} + a_8 p_{i,j,k} + a_9 p_{i+1,j,k} + a_{10} p_{i,j+1,k} + a_{11} p_{i,j-1,k+1} + \\ &a_{12} p_{i-1,j,k+1} + a_{13} p_{i,j,k+1} + a_{14} p_{i+1,j,k+1} + a_{15} p_{i,j+1,k+1} = R_p \end{aligned} \tag{37}$$

where

$$a_1 = -\left(\frac{(\sigma_y)_{i,j-1,k}}{2\Delta y(\Delta\sigma_k + \Delta\sigma_{k-1})} + \frac{(\sigma_y)_{i,j,k}}{2\Delta y(\Delta\sigma_k + \Delta\sigma_{k-1})}\right)$$

$$a_2 = -\left(\frac{(\sigma_x)_{i-1,j,k}}{2\Delta x(\Delta\sigma_k + \Delta\sigma_{k-1})} + \frac{(\sigma_x)_{i,j,k}}{2\Delta x(\Delta\sigma_k + \Delta\sigma_{k-1})}\right)$$

$$a_3 = -\frac{(\sigma_x^2 + \sigma_y^2 + \frac{1}{D^2})_{i,j,k}}{0.5(\Delta\sigma_k + \Delta\sigma_{k-1})\Delta\sigma_{k-1}}$$

$$a_4 = \frac{(\sigma_x)_{i+1,j,k}}{2\Delta x(\Delta\sigma_k + \Delta\sigma_{k-1})} + \frac{(\sigma_x)_{i,j,k}}{2\Delta x(\Delta\sigma_k + \Delta\sigma_{k-1})}$$

$$a_5 = \frac{(\sigma_y)_{i,j+1,k}}{2\Delta y(\Delta\sigma_k + \Delta\sigma_{k-1})} + \frac{(\sigma_y)_{i,j,k}}{2\Delta y(\Delta\sigma_k + \Delta\sigma_{k-1})}$$

$$a_6 = a_{10} = -\frac{1}{\Delta y^2} \quad a_7 = a_9 = -\frac{1}{\Delta x^2}$$

$$a_8 = \frac{2}{\Delta x^2} + \frac{2}{\Delta y^2} + \frac{(\sigma_x^2 + \sigma_y^2 + \frac{1}{D^2})_{i,j,k}}{0.5(\Delta\sigma_k + \Delta\sigma_{k-1})\Delta\sigma_k} + \frac{(\sigma_x^2 + \sigma_y^2 + \frac{1}{D^2})_{i,j,k}}{0.5(\Delta\sigma_k + \Delta\sigma_{k-1})\Delta\sigma_{k-1}}$$

$$a_{11} = \frac{(\sigma_y)_{i,j-1,k}}{2\Delta y(\Delta\sigma_k + \Delta\sigma_{k-1})} + \frac{(\sigma_y)_{i,j,k}}{2\Delta y(\Delta\sigma_k + \Delta\sigma_{k-1})}$$

$$a_{12} = \frac{(\sigma_x)_{i-1,j,k}}{2\Delta x(\Delta\sigma_k + \Delta\sigma_{k-1})} + \frac{(\sigma_x)_{i,j,k}}{2\Delta x(\Delta\sigma_k + \Delta\sigma_{k-1})}$$

$$a_{13} = -\frac{(\sigma_x^2 + \sigma_y^2 + \frac{1}{D^2})_{i,j,k}}{0.5(\Delta\sigma_k + \Delta\sigma_{k-1})\Delta\sigma_k}$$

$$a_{14} = -\left(\frac{(\sigma_x)_{i+1,j,k}}{2\Delta x(\Delta\sigma_k + \Delta\sigma_{k-1})} + \frac{(\sigma_x)_{i,j,k}}{2\Delta x(\Delta\sigma_k + \Delta\sigma_{k-1})}\right)$$

$$a_{15} = -\left(\frac{(\sigma_y)_{i,j+1,k}}{2\Delta y(\Delta\sigma_k + \Delta\sigma_{k-1})} + \frac{(\sigma_y)_{i,j,k}}{2\Delta y(\Delta\sigma_k + \Delta\sigma_{k-1})}\right)$$

$$R_p = -\frac{\rho}{\Delta t}\left(\frac{\partial u^*}{\partial x} + \frac{\partial u^*}{\partial \sigma}\frac{\partial \sigma}{\partial x^*} + \frac{\partial v^*}{\partial y} + \frac{\partial v^*}{\partial \sigma}\frac{\partial \sigma}{\partial y^*} + \frac{1}{D}\frac{\partial w^*}{\partial \sigma}\right)$$

where $\sigma_x = \frac{\partial \sigma}{\partial x^*}$ and $\sigma_y = \frac{\partial \sigma}{\partial y^*}$.

Uniform gridding is used in the horizontal direction while gridding in the vertical direction is generalized to be non-uniform in order to capture the bottom and surface boundary layers when desired. The coefficient matrix is asymmetric and has a total of 15 diagonal lines. The linear system is solved using the high performance preconditioner HYPRE software library. With p solved, the non-hydrostatic velocities at each stage can be updated from equation (33) to (35).

3.4 Boundary conditions

Boundary conditions are required for all the physical boundaries in order to solve the governing equations. Kinematic constraints on the surface and bottom boundaries have been discussed previously and are given by equations (3) and (4).

At the free surface, the continuity of normal and tangential stresses is enforced. With wind effects absent, the tangential stress equals zero, resulting in

$$\frac{\partial u}{\partial \sigma}|_{z=\eta} = \frac{\partial v}{\partial \sigma}|_{z=\eta} = 0 \tag{38}$$

The zero pressure condition on the free surface is applied when the Poisson equation is solved.

$$p|_{z=\eta} = 0 \tag{39}$$

At the bottom, the normal velocity and the tangential stress are prescribed. The normal velocity w is imposed through the kinematic boundary condition (4). For the inviscid case considered here, we use free-slip boundary conditions on the bottom and impose a zero-tangential-stress condition

$$\frac{\partial u}{\partial \sigma}|_{z=-h} = \frac{\partial v}{\partial \sigma}|_{z=-h} = 0 \tag{40}$$

Note that a form of bottom shear stresses is also implemented in the model and is not described here in the present application.

The Neumann boundary condition is used for dynamic pressure, which is directly obtained from the governing equation for w.

$$\frac{\partial p}{\partial \sigma}|_{z=-h} = -\rho D \frac{dw}{dt}|_{z=-h} \tag{41}$$

where w at $z = -h$ is given by (37). In the application to an underwater landslide, we linearize the resulting boundary condition which gives

$$\frac{\partial p}{\partial \sigma}|_{z=-h} = \rho D \frac{\partial^2 h}{\partial t^2} \tag{42}$$

At the closed boundaries or vertical walls, free-slip boundary conditions are imposed, so that the normal velocity and the tangential stress are set to zero. The normal pressure gradient is zero. At inflow, both free surface and velocities calculated from the analytical solutions are specified. In the lateral direction, periodic boundary conditions can be applied. To facilitate the parallel implementation, we used two ghost cells at each boundaries. The boundary conditions are specified at the ghost cells.

3.5 Parallelization

In parallelizing the computational model, we used a domain decomposition technique to subdivide the problem into multiple regions and assign each subdomain to a separate processor core. Each subdomain region contains an overlapping area of ghost cells. The Message Passing Interface (MPI) with non-blocking communication is used to exchange data in the overlapping region between neighboring processors. The parallelized HYPRE software library is used in solving the Poisson equation.

4 Configuration of idealized solid slide

Enet and Grilli (2007) carried out a laboratory experiment of surface waves generated by a three-dimensional under-water slide in a 3.7 m wide, 1.8 m deep and 30 m long wave tank with a plane underwater slope with $\theta = 15^\circ$ angle. The dataset of the laboratory experiment was suggested as a benchmark test for simulating landslide-generated tsunamis during the National Tsunami Hazard Mitigation Program workshop at Galveston, TX, in 2010. The field-scale application of submarine landslide tsunami follows the laboratory configuration of the solid slide with a realistic bathymetry.

4.1 Laboratory case

In Enet and Grilli (2007), the solid slide geometry was modeled using truncated hyperbolic secant functions in x and y direction, which is given by

$$\zeta = \frac{T}{1-\epsilon}[\mathrm{sech}(k_b x)\mathrm{sech}(k_w y) - \epsilon] \tag{43}$$

where $k_b = 2C/b$, $k_w = 2C/w$ and $C = a\cosh(1/\epsilon)$. The slide has length b, width w and thickness T. The truncation parameter ϵ is 0.717 as chosen by Enet and Grilli. The vertical cross section of the landslide is shown in Figure **??**.

The slide volume is calculated by

$$V_b = bwT\left(\frac{f^2-\epsilon}{1-\epsilon}\right) \quad \text{with} \quad f = \frac{2}{C}atan\sqrt{\frac{1-\epsilon}{1+\epsilon}} \tag{44}$$

The landslide is initially located at the submergence depth d. The movement of the landslide is prescribed as

$$s(t) = s_0 \ln\left(\cosh\frac{t}{t_0}\right) \tag{45}$$

which closely approximates the landslide displacement measured in experiments. s_0 and t_0 are given by

$$s_0 = \frac{u_t^2}{a_0}, \quad t_0 = \frac{u_t}{a_0} \tag{46}$$

where a_0 is the landslide initial acceleration measured in the experiment.

4.2 Field-scale case

The slide geometry for field-scale cases is expressed by the same truncated hyperbolic secant function (43) as used in Enet and Grilli (2007). The motion of the landslide can be expressed by balancing inertia, gravity, buoyancy, Coulomb friction, hydrodynamic friction and drag forces (Enet and Grilli, 2007). Then the motion of the landslide is governed by

$$(M_b + \Delta M_b)\frac{d^2 s}{dt^2} = (M_b - \rho_w V_b)(sin\theta - C_n cos\theta)g - \frac{1}{2}\rho_w (C_F A_w + C_D A_b)(\frac{ds}{dt})^2 \qquad (47)$$

where g is gravitational acceleration, ΔM_b,A_w and A_b are slide model added mass, wetted surface area and main cross section perpendicular to the direction of motion, respectively. C_F is skin friction coefficient. C_D is form drag coefficient. $C_n = \tan\phi$ is the basal Coulomb friction coefficient.

With the above equation, the terminal velocity of the landslide can be obtained by (Enet and Grilli, 2007)

$$u_t = \sqrt{gb\sin\theta\left(1 - \frac{\tan\phi}{\tan\theta}\right)\frac{\gamma - 1}{C_d'}\frac{2(f^2 - \epsilon)}{f - \epsilon}} \qquad (48)$$

where $\gamma = \rho_b/\rho_w$, the ratio of slide density to water density. $C_d' = C_F A_w/A_b + C_d$ is global drag coefficient. Based on the geotechnical considerations, the basal Coulomb friction is negligible as compared to inertia, gravity and hydrodynamic forces when the landslide is in motion (Enet and Grilli, 2007). This is expressed as $\tan\phi \ll \tan\theta$ in the equation. The drag coefficient $C_d' = 1.0$ as suggested by Enet and Grilli.

The initial acceleration a_0 can be described as

$$a_0 = g\sin\theta \qquad (49)$$

where θ is the shelf slope angle. (49) does not take into account the added mass coefficient for a conservative consideration. The slide is assumed to move with a constant acceleration a_0 to reach the terminal velocity u_t in field case studies.

5 Direction for modeling landslide tsunami waves

5.1 Program flow chart

The code was written using Fortran 90 with c preprocessor (cpp) statements for separation of the source code. Arrays are dynamically allocated at runtime. Precision is selected using the selected real kind Fortran intrinsic function defined in the makefile. The default precision is single. The present version of NHWAVE includes a number of options including

1. choice of serial or parallel code
2. landslide

3. Intel compiler which uses a different random function for wavemaker

and other options not for landslide tsunami cases.

The flow chart is shown in Figure 2.

5.2 Installation and compilation

NHWAVE is is distributed in a compressed file. To install the programs, first, uncompress the package. Then use

> tar xvf *.tar

to extract files from the uncompressed package. The exacted files will be distributed in two new directories: /src and /work.

To compile the program, go to /src and modify Makefile if needed. There are several necessary flags in Makefile which need to be specified. These include:

-DDOUBLE PRECISION: use double precision, default is single precision.

-DPARALLEL: use parallel mode, default is serial mode.

-DLANDSLIDE: include landslide applications. *This option must be used for landslide applications*

-DINTEL: if INTEL compiler is used, this option can make use of FPORT for the RAND() function

CPP: path to CPP directory

FC: Fortran compiler

For parallel runs, the software library ***HYPRE*** **needs to be installed**. The library can be downloaded from

https://computation.llnl.gov/casc/hypre/software.html

After the installation of *HYPRE*, the library has to be specified in the *Libraries* section:

LIBS = -L/user/hypre/parallel/lib -lHYPRE

INCS = -I/user/hypre/parallel/include

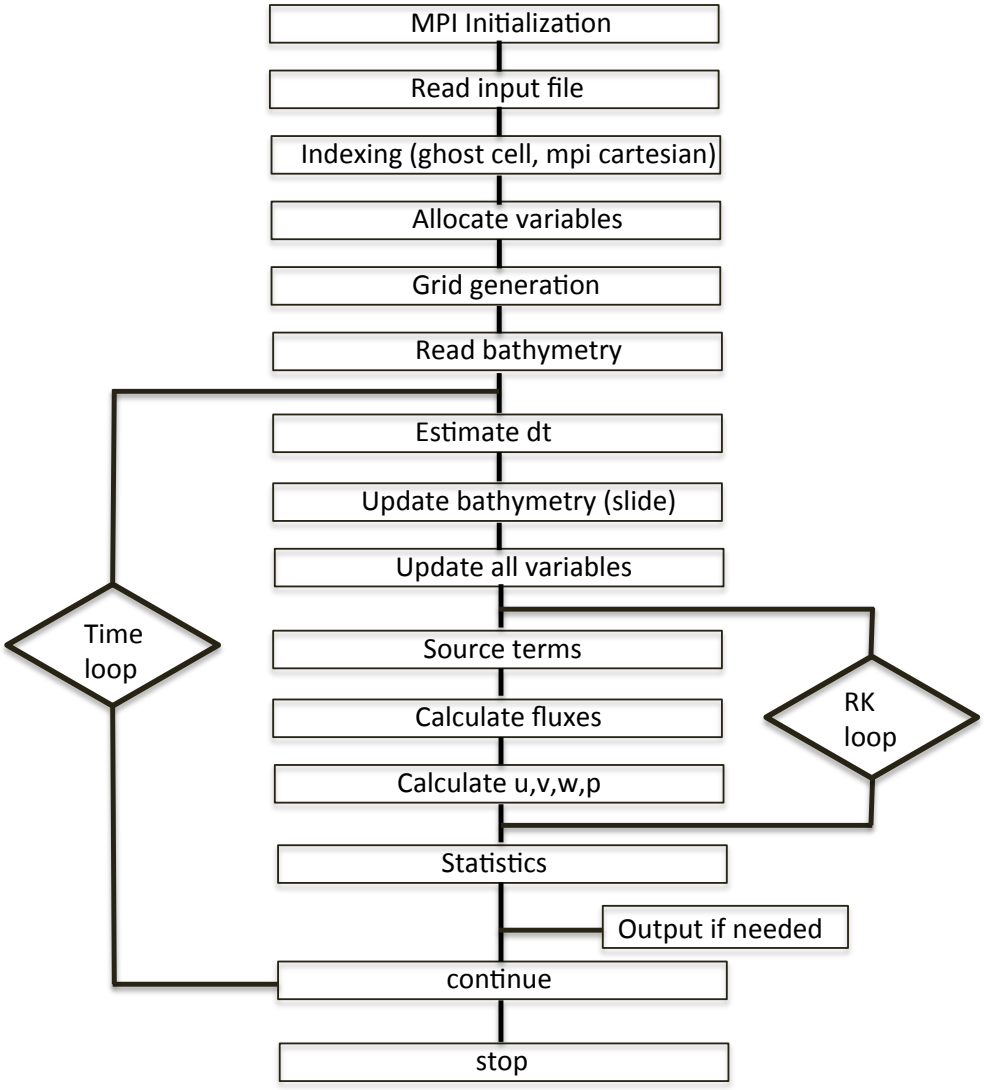

Figure 2: Flow chart of the main program.

Then execute

> make

The executable named *nhwave* will be generated. Use *make clean* each time when modifying Makefile.

Copy *nhwave* to /work/. Modify *input.txt* as needed and execute *nhwave*.

5.3 Input

Following are descriptions of parameters in input.txt. All parameter names are case sensitive.

TITLE: title of your case, only used for log file.

RESULT FOLDER

RESULT_FOLDER = ./results/

DIMENSION

Mglob: global dimension in x direction

Nglob: global dimension in y direction

Kglob: global dimension in z (σ) direction

TIME

SIM_STEPS: total time steps, for debug use, set a large number if not in debug mode

TOTAL_TIME: simulation time in seconds

PLOT_INTV: output interval in seconds (Note, output time is not exact because adaptive dt is used.)

SCREEN_INTV: time interval (s) of screen print.

GRID SIZE

DX: grid size(m) in x direction.

DY: grid size(m) in y direction.

VERTICAL GRID OPTION

IVGRD: 1 for uniform; 2 for exponential (not available for landslide application)

TIME STEP

DT_INI : initial time step

DT_MIN: minimum time step, if DT calculated based on CFL is smaller than DT_MIN, the program stops

DT_MAX: maximum time step

BATHYMETRY

DEPTH_TYPE: CELL_CENTER for input depth is defined at cell center, CELL_GRID for input depth defined at cell corner. Note that for LANDSLIDE application, depth must be defined at cell center.

The read format in the code is shown below.

DO J=1,Nglob

READ(1,*)(Depth(I,J),I=1,Mglob)

ENDDO

ANA_BATHY : ANA_BATHY = T represents analytical bathymetry in the laboratory experiments by Enet and Grilli (2007).

DEPTH_FILE: depth file name.

NUMERICS

HIGH_ORDER: order of spatial scheme, usually set HIGH_ORDER = SECOND, meaning the 2nd order scheme

TIME_ORDER: order of RK scheme, usually set TIME_ORDER = SECOND, meaning the 2nd order scheme

NON-HYDRO

NON_HYDRO: T for non-hydrostatic solver, F for shallow water equation solver.

COURANT NUMBER

CFL: Courant number, usually set CFL = 0.5

TURBULENCE MODEL

VISCOUS_FLOW: T for viscous flow, F for non viscous flow. For Landslide application, set F.

VISCOSITY: viscosity coefficient. It is invalid if VISCOUS_FLOW = F.

WETTING AND DRYING

MinDep: minimum depth for wetting and drying.

POISSON SOLVER

ISOLVER: 1 for Modified Incomplete Cholesky CG. 2 for Incomplete Cholesky GMRES. 3 for Successive Overrelaxation (SOR) GMRES.

ITMAX: maximum number of iterations. Usually ITMAX = 1000.

TOL: accuracy for iteration. Usually TOL = 1.0 E-8.

BOUNDARY CONDITION

BC_X0: west boundary condition. 1 for free-slip 2 for no-slip. 3 for influx. 4 for outflux. 5 for periodic in x. 6 for periodic in y.

BC_Xn: east boundary condition. The same as above.

BC_Y0: south boundary condition. The same as above.

BC_Yn: north boundary condition. The same as above.

BC_Z0: top boundary condition. The same as above.

BC_Zn: bottom boundary condition. The same as above.

For landslide applications, set 1 for all boundaries.

WAVEMAKER

Wavemaker is not used for landslide applications.

SPONGE LAYER

SPONGE_ON: logical parameter for sponge layer. T = .TURE., F = .FALSE.
Sponge_West_Width: width of sponge layer (m) at west boundary
Sponge_East_Width: width of sponge layer (m) at east boundary
Sponge_South_Width: width of sponge layer (m) at south boundary
Sponge_North_Width: width of sponge layer (m) at north boundary
R_Sponge: decay rate in sponge layer. Its values are between 0.85 and 0.95.
A_Sponge: maximum damping magnitude. The value is about 5.0.

LANDSLIDE PARAMETERS

H_slide : height of slide
L_slide : length of slide
W_slide : width of slide
e_slide : parameter e in Enet and Grilli
Angle_slide : slide angle (deg) from east clockwise
X0_slide : initial center x0 of slide (in meters,
(0,0) is SW corner of computational domain)
Y0_slide : initial center y0 of slide (meters)
Slope_slide : bathymetry slope at slide (deg), use constant
TermV_slide : terminal velocity of slide (calculated using Enet and Grilli)

OUTPUT

Logical parameters for output, T = .TRUE., F = .FALSE.
OUT_H : water depth
OUT_E : surface elevation
OUT_U : velocity in x direction
OUT_V : velocity in y direction
OUT_W : velocity in z direction
OUT_P : dynamic pressure
OUT_K : turbulent kinetic energy (not used for landslide application)
OUT_D : turbulent dissipation rate (not used for landslide application)
OUT_S : shear production (not used for landslide application)
OUT_C : eddy viscosity (not used for landslide application)
OUT_B : bubble void fraction (not used for landslide application)
OUT_preview: output for debug (not used for landslide application)
For PROBE OUTPUT:
NSTAT: gauge number. set NSTAT=0 if there is no probe output. The probe location (x,y) is defined in stat.txt.
PLOT_INTV_STAT: time interval (s) for probe output.

5.4 Output

The output files are saved in the result directory defined by RESULT FOLDER in input.txt. For outputs in ASCII, a file name is a combination of variable name and an output series number such as eta_0001, eta_0002, ... The format and read/write algorithm are consistent with the input depth file. Time series output for stations is a series of numbered files such as probe_0001, probe_0002.

6 Examples

Two examples are presented in the user's guide. One is a laboratory case (Enet and Grilli, 2007) which has been done in Ma et al. (2012) and is a part of benchmark tests in Tehranirad et al. (2012). The other is a field-scale case in which the slide is set up based on a collapse of the shelf break slope of the carbonate platform.

6.1 Enet and Grilli (2007) Laboratory Case

In Enet and Grilli (2007), the laboratory experiments were performed on a plane slope with angel $\theta = 15^{\circ}$, using a smooth streamlined Gaussian-Shaped body as described in (**??**). The solid body was released at time $t = 0$ from different initial submergence depths (d) as shown in Figure 3. Available measurements include slide kinematics obtained from slide acceleration utilizing a micro-accelerometer within the slide, time passage of the slide, and surface elevation for four gauges. All seven laboratory cases have been simulated in Tehranirad et al. (2012) with satisfactory model/data comparisons. Here, we demonstrate the first case (Case A) of Tehranirad et al. with the initial slide location at d =61 mm water depth.

The model bathymetry and slide geometry follow the laboratory experiment setup. The computational domain is only half of the lab domain in terms of crosswise symmetry of the experiment. The model dimension is $500 \times 90 \times 3$ (points in x, y, and z). $dx = dy = 0.02$m. The detailed model input parameters are listed below (parameters irrelevant to the application are not listed).

```
! ——————————DIMENSION————————————————
Mglob = 500
Nglob = 90
Kglob = 3
! ——————————————TIME————————————————-
SIM_STEPS = 100000000
TOTAL_TIME = 4.0
PLOT_START = 0.02
PLOT_INTV = 0.02
SCREEN_INTV = 0.02
! ——————————————GRID————————————————-
DX = 0.02
```

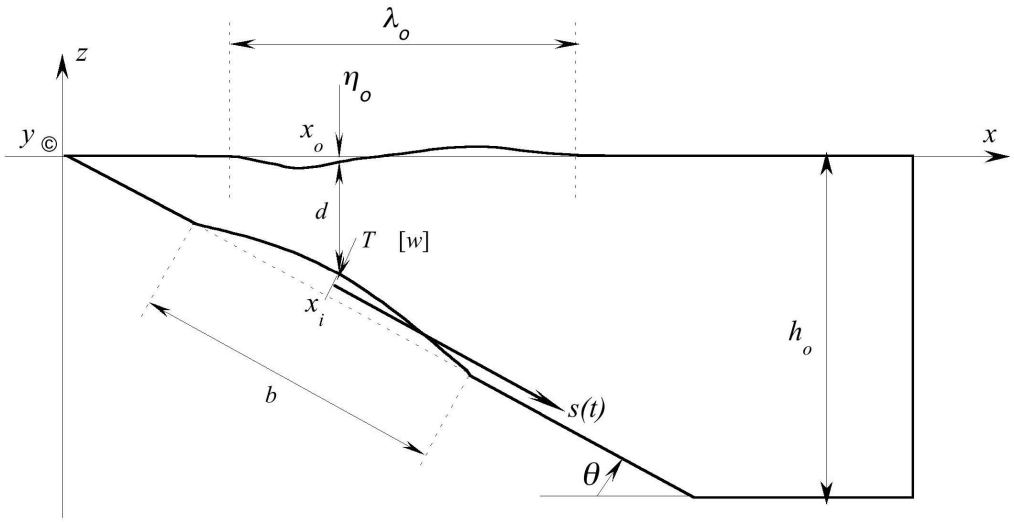

Figure 3: Vertical Cross section of underwater landslide (Enet and Grilli, 2007)

```
DY = 0.02
! ————————————-TIME STEP————————————
DT_INI = 0.00001
DT_MIN = 0.00001
DT_MAX = 0.10000
! ————————————-BATHYMETRY————————————
DEPTH_TYPE = CELL_CENTER
ANA_BATHY = T
DEPTH_FILE = depth.txt
! ————————————-NUMERICS————————————-
HIGH_ORDER = SECOND
TIME_ORDER = SECOND
! ————————————-NON-HYDRO————————————
! if non-hydrostatic simulation
NON_HYDRO = T
! ————————————POISSON SOLVER————————————-
ISOLVER = 2
ITMAX = 1000
TOL = 1.e-8
! ————————————-BOUNDARY_TYPE————————————
BC_X0 = 1
BC_Xn = 1
```

BC_Y0 = 1
BC_Yn = 1
BC_Z0 = 1
BC_Zn = 1
! ——————————PROBE OUTPUT————————————————
NSTAT = 4
PLOT_INTV_STAT = 0.01
! ——————————— LANDSLIDE PARAMETERS ———————————
H_slide = 0.082
L_slide = 0.395
W_slide = 0.680
e_slide = 0.717
Angle_slide = 0.0
X0_slide = 0.551
Y0_slide = 0.0
Slope_slide = 15.0
TermV_slide = 1.70
ACC_lab = 1.20
! ——————————————OUTPUT————————————————-
OUT_H = T
OUT_E = T
OUT_U = T
OUT_V = T
OUT_W = T
OUT_P = F
OUT_K = F
OUT_D = F
OUT_S = F
OUT_C = F
OUT_B = F
OUT_preview = F

The matlab script BM3_loader_1.m can be used for post-processing. Figure 4 shows the model/data comparisons at four gauge locations.

6.2 Field Example

The field example is based on a collapse of the shelf break slope of the carbonate platform which forms the continental shelf on the western side of the Florida peninsula. The Florida Escarpment landslide was described in ten Brink et al (2009). The bathymetry and the outlined slide are shown in Figure 5. Based on Brink et al., the slide is modeled using an area A = 647.47 km^2, width w =

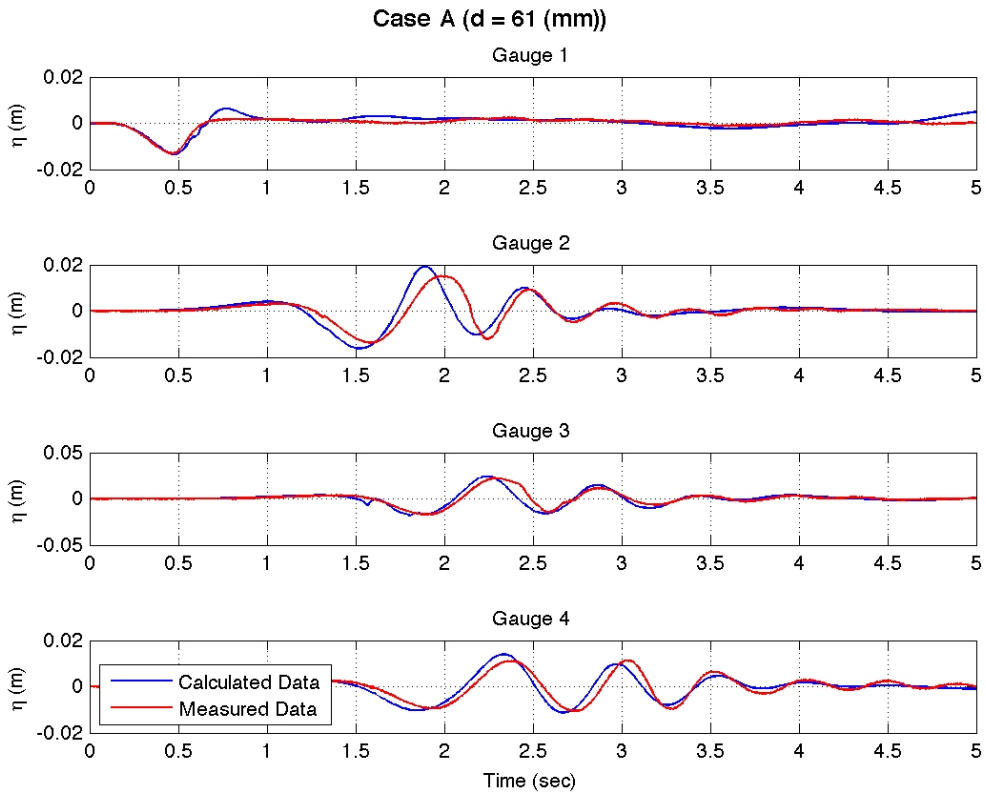

Figure 4: Model/data comparisons at four gauge locations (case A, Enet and Grilli, 2007).

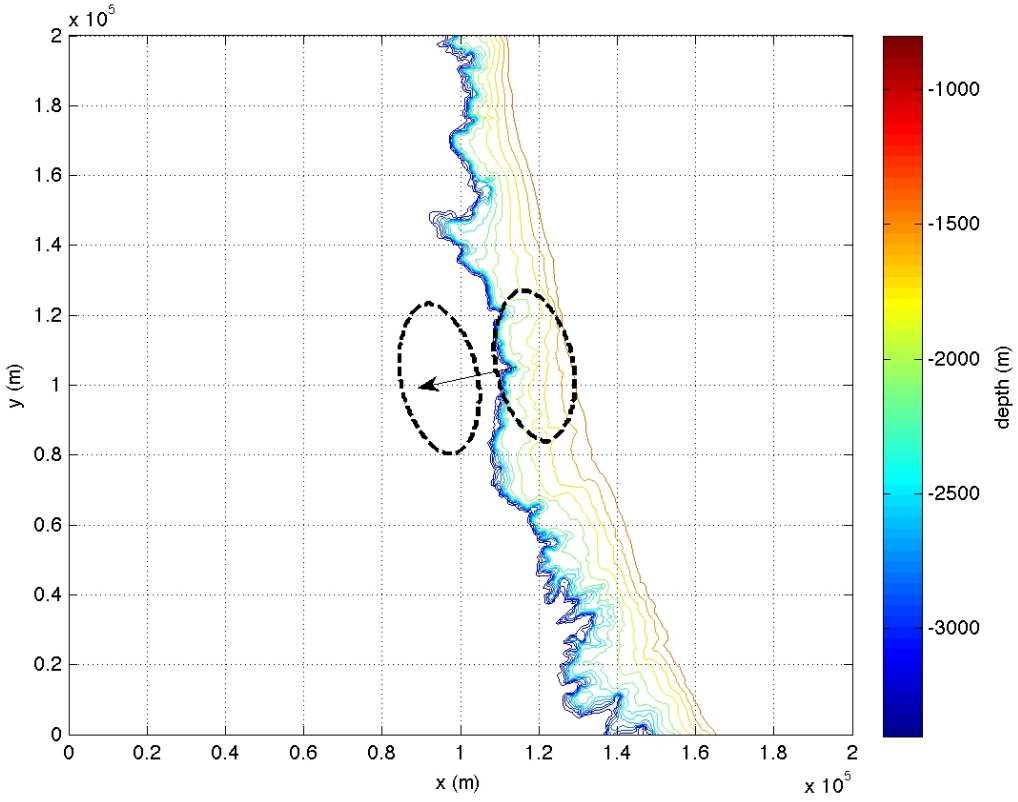

Figure 5: Bathymetry and the slide moving path.

42.938 km, length b=19.202 km and height $T = 66.0$ m (volume V_b = 16.2 km^3). The effective local slope angle was estimated as 5.8 degrees based on bathymetry contours. The terminal velocity is 134.34 m/s, calculated by (48).

The model dimension is $400 \times 400 \times 3$ (points in x, y, and z). $dx = dy = 500$m. The detailed model input parameters are listed below (parameters irrelevant to the application are not listed). Figure 6 shown the surface elevation distribution at T = 250 s.

```
! ——————DIMENSION————————————————
Mglob = 400
Nglob = 400
Kglob = 3
! ——————————TIME————————————————-
SIM_STEPS = 100000000
TOTAL_TIME = 600.0
PLOT_START = 0.0
PLOT_INTV = 5.0
SCREEN_INTV = 5.0
! ——————————GRID————————————————-
DX = 500.0
DY = 500.0
! ——————————-TIME STEP———————————
DT_INI = 1.0
DT_MIN = 0.00001
DT_MAX = 60.0
! ——————————-BATHYMETRY———————————
DEPTH_TYPE = CELL_CENTER
ANA_BATHY = F
DEPTH_FILE = depth_fl.txt
! ——————————-NUMERICS————————————-
HIGH_ORDER = SECOND
TIME_ORDER = SECOND
! ——————————-NON-HYDRO———————————
! if non-hydrostatic simulation
NON_HYDRO = T
! ——————————POISSON SOLVER——————————-
ISOLVER = 2
ITMAX = 1000
TOL = 1.e-8
! ——————————-BOUNDARY_TYPE———————————
BC_X0 = 1
BC_Xn = 1
```

BC_Y0 = 1
BC_Yn = 1
BC_Z0 = 1
BC_Zn = 1
! ——————PROBE OUTPUT————————————
NSTAT = 4
PLOT_INTV_STAT = 0.01
! ——————— LANDSLIDE PARAMETERS ———————
H_slide = 66.0
L_slide = 19202.376
W_slide = 42938.0
e_slide = 0.717
Angle_slide =-171.02
X0_slide = 118908.0
Y0_slide = 105655.0
Slope_slide = 5.8
TermV_slide = 134.34
ACC_lab = 1.20
! ——————————OUTPUT————————————-
OUT_H = T
OUT_E = T
OUT_U = T
OUT_V = T
OUT_W = T
OUT_P = F
OUT_K = F
OUT_D = F
OUT_S = F
OUT_C = F
OUT_B = F
OUT_preview = F

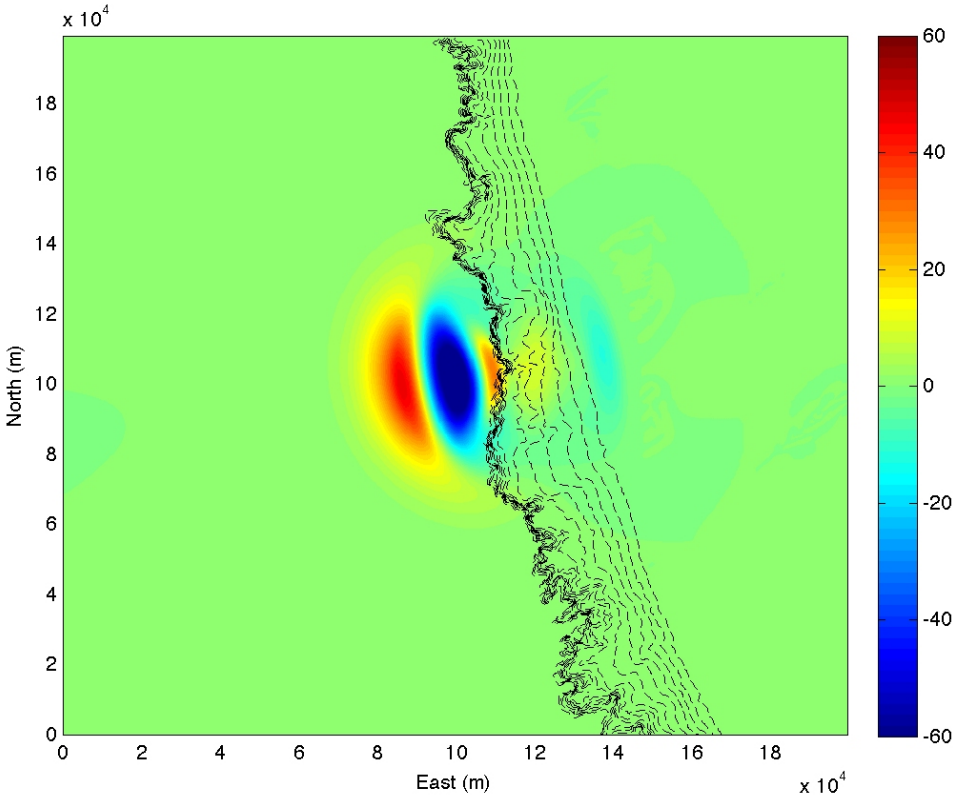

Figure 6: Surface elevation at T = 250 s

References

Bradford, S.F., 2005, "Godunov-based model for nonhydrostatic wave dynamics", *J. Waterway Port Coastal Ocean Eng.*, **131**, 226-238.

Collins-Sussman, B., Fitzpatrick, B. W. and Pilato, C. M., 2004, *Version control with Subversion*, OReilly Media, Inc., Sebastopol.

Enet, F. and Grilli, S. T., 2007, "Experimental study of tsunami generation by three-dimensional rigid underwater landslides, *J. Waterway, Port, Coastal and Ocean Engineering*, **133**, 442-454.

Gottlieb, S., Shu C.-W., and Tadmore, E., 2001, "Strong stability-preserving high-order time discretization methods", *SIAM Rev.*, **43**, 89 - 112.

Harten, A., Lax, P., van Leer, B., 1983, "On upstream differencing and Godunov-type schemes for hyperbolic conservation laws", *SIAM Rev.*, **25**, 35.

Kim, D. H., Cho, Y. S. and Kim, H. J., 2008, "Well balanced scheme between flux and source terms for computation of shallow-water equations over irregular bathymetry", *J. Eng. Mech.*, **134**, 277-290.

Liang, Q. and Marche, F., 2009, "Numerical resolution of well-balanced shallow water equations with complex source terms", *Adv. Water Res.*, **32**, 873 - 884.

Ma, G., Shi, F. and Kirby, J. T., 2012, "Shock-capturing non-hydrostatic model for fully dispersive surface wave processes", *Ocean Modelling*, **43-44**, 22-35.

Patankar, S.V., 1980. *Numerical heat transfer and fluid flow*, McGraw-Hill, New York.

Phillips, N.A., 1957, "A coordinate system having some special advantages for numerical forecasting", *J. Meteor.*, **14**, 184-185.

Shi, F., Kirby, J. T., Harris, J. C., Geiman, J. D. and Grilli, S. T., 2012, "A high-order adaptive time-stepping TVD solver for Boussinesq modelling of breaking waves and coastal inundation", *Ocean Modelling*, **43-44**, 36-51.

Stelling, G., Zijlema, M., 2003, "An accurate and efficient finite-difference algorithm for non-hydrostatic free-surface flow with application to wave propagation", *Int. J. Numer. Methods Fluids*, **43**, 1-23.

Synolakis, C. E., Bernard, E. N., Titov, V. V., Kânoğlu, U. and González, F. I., 2007, "Standards, criteria, and procedures for NOAA evaluation of tsunami numerical models", NOAA Tech. Memo. OAR PMEL-135, Pacific Marine Env. Lab., Seattle.

ten Brink, U., Twichell, D., Lynett, P., Geist, E., Chaytor, J., Lee, H., Buczkowski, B. and Flores, C., 2009, "Regional assessment of tsunami potential in the Gulf of Mexico, U. S. Geological Survey Administrative Report.

Tehranirad, B., Kirby, J. T., Ma, F. and Shi, F., 2012, "Tsunami benchmark results for non-hydrostatic wave model NHWAVE, Version 1.0, Research Report No. CACR-12-03, Center for Applied Coastal Research, University of Delaware, Newark.

Yuan, H., Wu, C.-H., 2004, "A two-dimensional vertical non-hydrostatic model with an implicit method for free-surface flows", *Int. J. Numer. Methods Fluids*, **44**, 811-835.

Zhou, J. G., Causon, D. M., Mingham C. G., and Ingram, D. M., 2001, "The surface gradient method for the treatment of source terms in the shallow-water equations", *J. Comp. Phys.*, 168, 1-25.