INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
*Int. J. Numer. Meth. Fluids* 2007; **54**:1173–1199
Published online 23 January 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1421

# Effect of vertical grid variability on a free surface flow model[§]

Scott F. Bradford[*, †, ‡]

*Image Science and Applications Branch, Code 7261, Naval Research Laboratory, 4555 Overlook Ave. SW,*
*Washington, DC 20375, U.S.A.*

## SUMMARY

A previously developed numerical model that solves the incompressible, non-hydrostatic, Navier–Stokes equations for free surface flow is analysed on a non-uniform vertical grid. The equations are vertically transformed to the $\sigma$-coordinate system and solved in a fractional step manner in which the pressure is computed implicitly by correcting the hydrostatic flow field to be divergence free. Numerical consistency, accuracy and efficiency are assessed with analytical methods and numerical experiments for a varying vertical grid discretization. Specific discretizations are proposed that attain greater accuracy and minimize computational effort when compared to a uniform vertical discretization. Published in 2007 by John Wiley & Sons, Ltd.

Received 25 September 2006; Revised 16 November 2006; Accepted 17 November 2006

KEY WORDS: Navier–Stokes equations; numerical models; flow simulation; wave propagation

## INTRODUCTION

The accurate simulation of currents and wave transformation in shallow water is needed for the reliable design of coastal structures and prediction of sediment transport and subsequent bottom evolution. Models based on the incompressible Navier–Stokes equations have been shown to realistically simulate surf zone dynamics because of minimal simplifying assumptions in their physics [1, 2]. These models use the marker-and-cell (MAC) method [3] or the volume-of-fluid (VOF) method [4] for free surface tracking, which are computationally intensive but yield accurate simulations of complex-free surfaces that include plunging and splashing. The MAC and VOF methods track the free surface by monitoring the movement of water in and out of stationary, Cartesian, computational cells. This Eulerian approach resolves shallow and deep regions to

[*] Correspondence to: Scott F. Bradford, Image Science and Applications Branch, Code 7261, Naval Research Laboratory, 4555 Overlook Ave. SW, Washington, DC 20375, U.S.A.

[†] E-mail: Scott.Bradford@nrl.navy.mil

[‡] Research Scientist.

[§] This article is a U.S. Government work and is in the public domain in the U.S.A.

Published in 2007 by John Wiley & Sons, Ltd.

the same degree and makes the application of free surface boundary conditions (BCs) difficult. In addition, many cells may be unused above the free surface, as well as below the bottom for domains with variable bathymetry.

Several other models have instead transformed the Navier–Stokes equations into the $\sigma$-coordinate system in which the top of the grid follows the free surface and the bottom of the grid follows the bathymetry [5]. Mahadevan *et al.* [6], Lin and Li [7], and Bradford [8] vertically integrated the incompressibility constraint to derive an equation for the surface position, while Li and Fleming [9] directly used the kinematic free surface BC. These Lagrangian approaches require fewer computational cells in the vertical direction than the MAC and VOF methods, and do not require the complicated reconstruction of the free surface. These factors yield a more computationally efficient model that makes large-scale coastal simulations more feasible, but at the expense of simulating discontinuous free surfaces.

Some researchers have also used a non-uniform vertical grid discretization to further reduce computational expense. The grid may be locally refined near the bed or the free surface to better resolve strong solution gradients, which reduces the truncation error with fewer computational cells than would otherwise be required when using a uniform grid. However, using a non-uniform grid can have undesirable consequences. MacCracken and Bornstein [10] discuss the implications of using a non-uniform, staggered grid with the donor-cell method. Brown and Pandolfo [11] analysed the implicit, centred finite difference method applied to the linear advection equation and found that refining the grid in the flow direction can introduce numerical instability. Thompson and Mastin [12] demonstrated that the grid point distribution must be carefully chosen to preserve numerical accuracy when discretizing derivatives on non-uniform grids. Yin and Fung [13], Marti *et al.* [14] and Treguier *et al.* [15] showed that using a constant grid stretching (the ratio of adjacent grid spacings is constant) reduces the accuracy of ocean circulation models.

In this paper, the model developed by Bradford [8] is analysed when using a non-uniform vertical grid. This model was previously analysed on a uniform grid by Bradford [16]. The effects of using a non-uniform vertical grid on the model's consistency, stability, accuracy, and efficiency are demonstrated and discussed. Specific vertical discretizations are proposed that increase model accuracy, as well as reduce computational expense.

## GOVERNING EQUATIONS

The model of interest solves the incompressible, Reynolds averaged, Navier–Stokes equations for small-scale flows in which the Earth's rotation and buoyancy effects are negligible. The governing equations are transformed vertically from $z$ space to $\sigma$ space *via* the following transformation:

$$\sigma = \frac{z-h}{D} \tag{1}$$

where $h$ is the free surface location and $D$ is the total water depth. Furthermore, the equations are transformed from $x$ and $y$ space *via* a curvilinear transformation to $\xi$ and $\eta$ space such that the grid size in each of the $\xi$ and $\eta$ directions is one. In this coordinate system, the governing equations are

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{1}{J}\left(\frac{\partial(\mathbf{F}-\mathbf{F}_{\mathrm{v}})}{\partial \xi} + \frac{\partial(\mathbf{G}-\mathbf{G}_{\mathrm{v}})}{\partial \eta}\right) + \frac{\partial(\mathbf{H}-\mathbf{H}_{\mathrm{v}})}{\partial \sigma} = \mathbf{S}_{\mathrm{h}} + \mathbf{S}_{\mathrm{p}} \tag{2}$$

where $\mathbf{U}=(D\ Du\ Dv\ Dw)^{\mathrm{T}}$, and $u$, $v$, and $w$ are the Cartesian velocity components. $J$ is the projected area of a computational cell in $x$–$y$ space, and $\mathbf{F}$, $\mathbf{G}$, and $\mathbf{H}$ denote the advective fluxes and hydrostatic pressure terms, which are defined as

$$\mathbf{F}=\begin{pmatrix} DU \\ DUu+\frac{1}{2}gD^2J\xi_x \\ DUv+\frac{1}{2}gD^2J\xi_y \\ DUw \end{pmatrix},\quad \mathbf{G}=\begin{pmatrix} DV \\ DVu+\frac{1}{2}gD^2J\eta_x \\ DVv+\frac{1}{2}gD^2J\eta_y \\ DVw \end{pmatrix},\quad \mathbf{H}=\begin{pmatrix} W \\ Wu \\ Wv \\ Ww \end{pmatrix} \tag{3}$$

where $g$ represents gravity and

$$\begin{aligned} U&=(u\xi_x+v\xi_y)J \\ V&=(u\eta_x+v\eta_y)J \\ W&=D(\sigma_t+u\sigma_x+v\sigma_y)+w=\mathscr{W}+D\sigma_t \end{aligned} \tag{4}$$

The terms $\xi_x$, $\xi_y$, $\eta_x$, $\eta_y$, $\sigma_t$, $\sigma_x$ and $\sigma_y$ are the grid transformation metrics. $\mathbf{F}_{\mathrm{v}}$, $\mathbf{G}_{\mathrm{v}}$ and $\mathbf{H}_{\mathrm{v}}$ represent viscous effects

$$\mathbf{F}_{\mathrm{v}}=\begin{pmatrix} 0 \\ vDJ\left(\frac{\partial u}{\partial\xi}l_{\xi\xi}+\frac{\partial u}{\partial\eta}l_{\xi\eta}+\frac{\partial u}{\partial\sigma}l_{\xi\sigma}\right) \\ vDJ\left(\frac{\partial v}{\partial\xi}l_{\xi\xi}+\frac{\partial v}{\partial\eta}l_{\xi\eta}+\frac{\partial v}{\partial\sigma}l_{\xi\sigma}\right) \\ vDJ\left(\frac{\partial w}{\partial\xi}l_{\xi\xi}+\frac{\partial w}{\partial\eta}l_{\xi\eta}+\frac{\partial w}{\partial\sigma}l_{\xi\sigma}\right) \end{pmatrix} \tag{5}$$

$$\mathbf{G}_{\mathrm{v}}=\begin{pmatrix} 0 \\ vDJ\left(\frac{\partial u}{\partial\xi}l_{\xi\eta}+\frac{\partial u}{\partial\eta}l_{\eta\eta}+\frac{\partial u}{\partial\sigma}l_{\eta\sigma}\right) \\ vDJ\left(\frac{\partial v}{\partial\xi}l_{\xi\eta}+\frac{\partial v}{\partial\eta}l_{\eta\eta}+\frac{\partial v}{\partial\sigma}l_{\eta\sigma}\right) \\ vDJ\left(\frac{\partial w}{\partial\xi}l_{\xi\eta}+\frac{\partial w}{\partial\eta}l_{\eta\eta}+\frac{\partial w}{\partial\sigma}l_{\eta\sigma}\right) \end{pmatrix} \tag{6}$$

$$\mathbf{H}_{\mathrm{v}} = \begin{pmatrix} 0 \\ vD\left(\dfrac{\partial u}{\partial \xi} l_{\xi\sigma} + \dfrac{\partial u}{\partial \eta} l_{\eta\sigma} + \dfrac{\partial u}{\partial \sigma} l_{\sigma\sigma}\right) \\ vD\left(\dfrac{\partial v}{\partial \xi} l_{\xi\sigma} + \dfrac{\partial v}{\partial \eta} l_{\eta\sigma} + \dfrac{\partial v}{\partial \sigma} l_{\sigma\sigma}\right) \\ vD\left(\dfrac{\partial w}{\partial \xi} l_{\xi\sigma} + \dfrac{\partial w}{\partial \eta} l_{\eta\sigma} + \dfrac{\partial w}{\partial \sigma} l_{\sigma\sigma}\right) \end{pmatrix} \tag{7}$$

$v$ is the eddy viscosity and

$$\begin{aligned} &l_{\xi\xi} = \xi_x^2 + \xi_y^2, \quad l_{\xi\eta} = \xi_x \eta_x + \xi_y \eta_y, \quad l_{\eta\eta} = \eta_x^2 + \eta_y^2 \\ &l_{\xi\sigma} = \xi_x \sigma_x + \xi_y \sigma_y, \quad l_{\eta\sigma} = \eta_x \sigma_x + \eta_y \sigma_y \end{aligned} \tag{8}$$

$\mathbf{S}_{\mathrm{h}}$ contains the source terms associated with a sloping bottom, while $\mathbf{S}_{\mathrm{p}}$ contains the non-hydrostatic pressure terms

$$\mathbf{S}_{\mathrm{h}} = \begin{pmatrix} 0 \\ -gDJ\left(\dfrac{\partial z_{\mathrm{b}}}{\partial \xi} \xi_x + \dfrac{\partial z_{\mathrm{b}}}{\partial \eta} \eta_x\right) \\ -gDJ\left(\dfrac{\partial z_{\mathrm{b}}}{\partial \xi} \xi_y + \dfrac{\partial z_{\mathrm{b}}}{\partial \eta} \eta_y\right) \\ 0 \end{pmatrix}, \quad \mathbf{S}_{\mathrm{p}} = \begin{pmatrix} 0 \\ -D\left(\dfrac{\partial p}{\partial \xi} \xi_x + \dfrac{\partial p}{\partial \eta} \eta_x + \dfrac{\partial p}{\partial \sigma} \sigma_x\right) \\ -D\left(\dfrac{\partial p}{\partial \xi} \xi_y + \dfrac{\partial p}{\partial \eta} \eta_y + \dfrac{\partial p}{\partial \sigma} \sigma_y\right) \\ -\dfrac{\partial p}{\partial \sigma} \end{pmatrix} \tag{9}$$

where $z_{\mathrm{b}}$ the bottom elevation and $p$ is the non-hydrostatic pressure component.

## SOLUTION PROCEDURE

The finite volume method is used to discretize Equation (2) in which the domain is divided into hexagonal computational cells indexed with $j, k, l$ where $\xi$, $\eta$ and $\sigma$ are in the direction of contiguous $j$, $k$ and $l$ indices, respectively. The dependent variables are defined as cell-average values and the computation is broken up in a fractional step manner in which $D$, $u$, $v$ and $w$ are computed under the assumption of hydrostatic flow ($\mathbf{S}_{\mathrm{p}} = 0$) as described in [8].

The final step in the solution process is to correct the hydrostatic flow field by including $\mathbf{S}_{\mathrm{p}}$. Momentum equations for the final velocities may be written as

$$u = \tilde{u} - \Delta t\left(\xi_x \frac{\partial p}{\partial \xi} + \eta_x \frac{\partial p}{\partial \eta} + \sigma_x \frac{\partial p}{\partial \sigma}\right) \tag{10}$$

$$v = \tilde{v} - \Delta t\left(\xi_y \frac{\partial p}{\partial \xi} + \eta_y \frac{\partial p}{\partial \eta} + \sigma_y \frac{\partial p}{\partial \sigma}\right) \tag{11}$$

$$w = \tilde{w} - \frac{\Delta t}{D} \frac{\partial p}{\partial \sigma} \tag{12}$$

DOI: 10.1002/fld

where $\tilde{u}$, $\tilde{v}$ and $\tilde{w}$ denote the hydrostatic velocities and $\Delta t$ is the time step. The final velocities are required to satisfy the discrete incompressibility constraint, which can be written as

$$\frac{1}{J}\{U_{j+1/2,k,l}-U_{j-1/2,k,l}+V_{j,k+1/2,l}-V_{j,k-1/2,l}\}+\frac{\mathscr{W}_{j,k,l+1/2}-\mathscr{W}_{j,k,l-1/2}}{\Delta\sigma_l}=0 \tag{13}$$

This is accomplished by discretizing Equations (10)–(12) at each cell face and substituting into Equation (13), which yields the following Poisson equation for $p_{j,k,l}$:

$$\begin{aligned}&a_1 p_{j-1,k,l-1}+a_2 p_{j,k-1,l-1}+a_3 p_{j,k,l-1}+a_4 p_{j,k+1,l-1}+a_5 p_{j+1,k,l-1}\\&\quad+a_6 p_{j-1,k-1,l}+a_7 p_{j-1,k,l}+a_8 p_{j-1,k+1,l}+a_9 p_{j,k-1,l}+a_{10} p_{j,k,l}\\&\quad+a_{11} p_{j,k+1,l}+a_{12} p_{j+1,k-1,l}+a_{13} p_{j+1,k,l}+a_{14} p_{j+1,k+1,l}+a_{15} p_{j-1,k,l+1}\\&\quad+a_{16} p_{j,k-1,l+1}+a_{17} p_{j,k,l+1}+a_{18} p_{j,k+1,l+1}+a_{19} p_{j+1,k,l+1}=R_{j,k,l}\end{aligned} \tag{14}$$

The $a$ coefficients are given in Appendix A and the right side is

$$\begin{aligned}R_{j,k,l}=&\frac{-1}{J\Delta t}(\tilde{U}_{j+1/2,k,l}-\tilde{U}_{j-1/2,k,l}+\tilde{V}_{j,k+1/2,l}-\tilde{V}_{j,k-1/2,l})\\&+\frac{\tilde{\mathscr{W}}_{j,k,l+1/2}-\tilde{\mathscr{W}}_{j,k,l-1/2}}{\Delta t\Delta\sigma_l}\end{aligned} \tag{15}$$

Assembly of Equation (14) over the entire domain yields a system of equations of the form

$$\mathbf{Ap}=\mathbf{R} \tag{16}$$

where $\mathbf{A}$ is an $N\times N$ matrix consisting of the $a$ coefficients and $N$ is the number of cells in the computational domain. $\mathbf{p}$ and $\mathbf{R}$ are the vectors of $p$ and $R$, respectively. Once Equation (16) is solved for $\mathbf{p}$, Equations (10)–(12) can be solved for the final non-hydrostatic velocities.

Boundary conditions (BCs) are applied by defining fictitious 'ghost' cells at all grid boundaries, which are adjacent to the boundary but outside the computational domain. At all lateral and bottom boundaries, a Neumann BC is used in which a zero gradient (normal to the boundary) of $p$ is specified. At the free surface, two BC choices are possible. The first is the Dirichlet type with $p=0$ in the ghost cell. The second is the Neumann type in which $p$ is set in the ghost cell to yield $p=0$ at the free surface. Both of these choices are discussed later.

## NUMERICAL CONSISTENCY, STABILITY, AND ACCURACY

The proposed method on a non-uniform grid is first analysed by applying it to solve the linearized, 1D, advection–diffusion equation for an arbitrary scalar, $c$

$$c_t+uc_x-\nu c_{xx}=0 \tag{17}$$

where the subscripts $t$ and $x$ denote differentiation in time and space, respectively. The flow velocity, $u$, and diffusion coefficient, $\nu$, are assumed to be constant. Advection is explicitly integrated in time with a predictor corrector approach, and diffusion is implicitly integrated with the Crank–Nicholson method. This method was analysed on a uniform grid by Bradford [16]. Note that in the 3D model, only vertical diffusion is treated implicitly, while lateral diffusion is explicitly integrated.

The model uses a slope limiter to improve spatial accuracy without introducing numerical oscillations near solution discontinuities. Since limiters use a computational stencil that varies in space and time, schemes that utilize them are non-linear even when applied to the linear advection–diffusion equation. This complicates the numerical analysis, but the Double Minmod limiter is closely related to the linear Fromm method, which was found to possess excellent phase accuracy by Bradford [16]. Therefore the analysis presented here is done for the Fromm method, which provides insight into the behaviour of the proposed model that uses the Double Minmod limiter.

For $u>0$, the advective flux at the $j+1/2$ face is computed in an upwind manner as

$$uc=u\left(c_j+\frac{\Delta c_j}{2}(1-\mathbf{C}_j)\right) \tag{18}$$

An analogous expression can be written for the $j-1/2$ face. The term $\mathbf{C}_j=u\Delta t/\Delta x_j$ is the Courant number, $\Delta x_j$ is the length of cell $j$ and $\Delta c_j$ is a cell average gradient of $c$ that is computed with a centred difference as

$$\Delta c_j=\left[\frac{\Delta x_{j-1/2}(c_{j+1}-c_j)}{\Delta x_{j+1/2}}+\frac{\Delta x_{j+1/2}(c_j-c_{j-1})}{\Delta x_{j-1/2}}\right]\frac{\Delta x_j}{\Delta x_{j+1/2}+\Delta x_{j-1/2}} \tag{19}$$

where $\Delta x_{j+1/2}=x_{j+1}-x_j$ and $\Delta x_{j-1/2}=x_j-x_{j-1}$.

The diffusive flux across the $j+1/2$ face is computed as

$$vc_x=v\frac{c_{j+1}-c_j}{\Delta x_{j+1/2}} \tag{20}$$

Differencing the diffusive fluxes at the $j+1/2$ and $j-1/2$ faces yields an approximation to $vc_{xx}$ in cell $j$ as

$$vc_{xx}=\frac{v}{\Delta x_j}\left(\frac{c_{j+1}-c_j}{\Delta x_{j+1/2}}-\frac{c_j-c_{j-1}}{\Delta x_{j-1/2}}\right) \tag{21}$$

On a uniform grid, Equation (21) is $O(\Delta x^2)$ accurate, but on a non-uniform grid it is not since the $c_x$ approximation in Equation (20) is no longer space-centred around the cell face and therefore not $O(\Delta x^2)$ accurate either.

The final form of the proposed method on a non-uniform grid as applied to Equation (17) is represented as

$$A_1c_{j+1}^{n+1}+A_2c_j^{n+1}+A_3c_{j-1}^{n+1}=A_4c_{j+1}^n+A_5c_j^n+A_6c_{j-1}^n+A_7c_{j-2}^n \tag{22}$$

where $n$ denotes the time level and the $A$ coefficients are given in Appendix B.

Taylor series in time and space can be substituted into Equation (22) to yield the following equivalent differential equation:

$$\begin{aligned}c_t+uc_x-v\frac{\Delta x_{j+1}+2\Delta x_j+\Delta x_{j-1}}{4\Delta x_j}c_{xx}&\\=\frac{u(\Delta x_j-\Delta x_{j-1})}{4}\left\{\frac{1}{2}\left(\frac{\Delta x_{j-1}}{\Delta x_j}+1\right)-\mathbf{C}_j\right\}c_{xx}+T&\end{aligned} \tag{23}$$

DOI: 10.1002/fld

where $T$ represents the remaining terms in the Taylor series, the first of which is proportional to $c_{xxx}$. This is the equation that the model solves exactly, but it is difficult to analyse without further knowledge of the grid variability.

Thompson and Mastin [12] introduced a grid point distribution function of the form

$$x(s) = f(s) \tag{24}$$

where $s$ is a uniform grid with $0 \leqslant s \leqslant 1$ and $\Delta s = 1/N$ ($N$ is the number of cells). $x$ may be expanded in a Taylor series in terms of $f$, which can then be used to rewrite the $\Delta x$'s in Equation (23) yielding

$$c_t + uc_x - v\left(1 + \frac{f_{sss}}{4N^2 f_s}\right) c_{xx} = \frac{uf_{ss}}{4N^2}\left(1 - \mathbf{C}_j - \frac{f_{ss}}{2Nf_s}\right) c_{xx} + T \tag{25}$$

The order of accuracy of a model on a non-uniform grid is ambiguous and Thompson and Mastin [12] defined it in two ways. The first definition is based on the behaviour of the truncation error as the number of cells is increased, while maintaining a fixed $f(s)$. The second definition is the behaviour of the error with a fixed the number of cells, but a varying $f(s)$. The latter definition is really a measure of local error since changing the distribution reduces the error where the cell size is reduced, and increases the error where the cell size is increased. The first definition measures global error, and a second-order accurate method on a non-uniform grid is defined as having a truncation error proportional to $N^{-2}$. Equation (25) shows that the method is consistent and second-order accurate if $f_{sss}/f_s$, $f_{ss}$ and $f_{ss}/f_s$ remain bounded as $N \to \infty$. Furthermore, $f_{sss}/f_s$, $f_{ss} > 0$ and $f_{ss}/f_s < 0$ introduces numerical diffusion, while the opposite introduces anti-diffusion and possible numerical instability.

Yin and Fung [13] chose to use a constant grid stretching in which the ratio of adjacent grid spacings is constant. In terms of $f$ this is expressed as

$$f(s) = S\frac{1 - r^{Ns}}{1 - r^N} \tag{26}$$

where $S$ is the length of the domain and $r$ is the constant rate of grid stretch. Using $r < 1$ refines the grid in the flow direction, while using $r > 1$ coarsens the grid. However, Marti *et al.* [14] and Treguier *et al.* [15] have shown this to be a poor choice since $f_s(0) = 0$, and therefore $f_{sss}/f_s \to -\infty$, which introduces numerical instability. For $s > 0$, $f_{sss}/f_s$ and $f_{ss}$ become proportional to $N^2$ and therefore numerical consistency and second-order accuracy are lost.

Thompson and Mastin [12] recommended the hyperbolic sine function for applications that require very small grid spacing, i.e. near solid boundaries when simulating turbulent boundary layers. For coarsening the grid in the flow direction this function is defined as

$$f(s) = S\frac{\sinh(rs)}{\sinh r} \tag{27}$$

where the larger the value of $r$, the greater the stretching. Refining the grid in the flow direction is accomplished by choosing

$$f(s) = S\left(1 + \frac{\sinh[r(s-1)]}{\sinh r}\right) \tag{28}$$

In either case, $f_{sss}/f_s = r^2$, which adds numerical dissipation and therefore does not create numerical instability. Incidentally, this term is a result of the approximation of Equation (21) and increases with increasing $r$. Coarsening the grid in the flow direction yields $f_{ss}/f_s>0$, which can introduce instability. This term is maximized at $s=1$, i.e. where the grid spacing is largest. This is also where $f_{ss}$ is maximized and in this case with $r>0$ Equation (25) becomes

$$c_t+uc_x-v\left(1+\frac{r^2}{4N^2}\right)c_{xx}=\frac{ur^2}{2N^2}\left(1-\mathbf{C}_j-\frac{r}{2N}\right)c_{xx}+T \tag{29}$$

In order to prevent numerical instability, the term multiplied by $c_{xx}$ on the right side of Equation (29) must remain positive in the worst case with $v=0$, which requires

$$r\leqslant 2N(1-\mathbf{C}_{\min})=2N\left(1-\mathbf{C}_{\max}\frac{\Delta x_{\min}}{\Delta x_{\max}}\right) \tag{30}$$

Note that $\mathbf{C}_{\max}$ and $\mathbf{C}_{\min}$ are the maximum and minimum $\mathbf{C}_j$ in the domain, while $\Delta x_{\max}$ and $\Delta x_{\min}$ are the maximum and minimum grid spacings. As $r$ and $N$ increases, $\Delta x_{\min}/\Delta x_{\max}$ becomes smaller and Equation (30) becomes

$$r<2N \tag{31}$$

Refining the grid in the flow direction yields nearly the same result except for the sign change of the $f_{ss}/f_s$ term. This term, along with the $f_{ss}$ term, are maximized at $s=0$ (once again at the location of greatest grid spacing.) In this case, Equation (25) becomes

$$c_t+uc_x-v\left(1+\frac{r^2}{4N^2}\right)c_{xx}=\frac{ur^2}{2N^2}\left(1-\mathbf{C}_j+\frac{r}{2N}\right)c_{xx}+T \tag{32}$$

In this case, the stability constraint reverts back to the uniform grid Courant condition ($\mathbf{C}_{\max}\leqslant 1$). However, Equation (31) should be obeyed in order to minimize numerical dissipation.

A Von Neumann analysis of this method is not strictly valid on a non-uniform grid. However, Brown and Pandolfo [11] demonstrated that such an analysis can provide insight into the model's behaviour. This is accomplished by inserting into Equation (22) the solution for a single harmonic of the form

$$c_j^n=g^n\mathrm{e}^{Ik_ix_j} \tag{33}$$

where $I=\sqrt{-1}$ and $k_i$ is the wave number of the $i$th harmonic. This yields the following expression for the amplification factor, $G=g^{n+1}/g^n$:

$$G=A_G+B_GI \tag{34}$$

where

$$\begin{aligned}A_G&=\frac{AC+BD}{C^2+D^2}\\ B_G&=\frac{BC-AD}{C^2+D^2}\end{aligned} \tag{35}$$

and

$$\begin{aligned}
A &= C_5 + C_4 \cos k_i \Delta x_{j+1/2} + C_6 \cos k_i \Delta x_{j-1/2} + C_7 \cos k_i (\Delta x_{j-1/2} + \Delta x_{j-3/2}) \\
B &= C_4 \sin k_i \Delta x_{j+1/2} - C_6 \sin k_i \Delta x_{j-1/2} - C_7 \sin k_i (\Delta x_{j-1/2} + \Delta x_{j-3/2}) \\
C &= C_2 + C_1 \cos k_i \Delta x_{j+1/2} + C_3 \cos k_i \Delta x_{j-1/2} \\
D &= C_1 \sin k_i \Delta x_{j+1/2} - C_3 \sin k_i \Delta x_{j-1/2}
\end{aligned} \tag{36}$$

where the $C$ coefficients are defined by Equation (B1) in Appendix B.

For numerical stability, $|G| \leqslant 1$ must be true for all possible values of $k_i$, which is defined as $k_i = i\pi/S$, $i = 0$ to $N$. Numerical evaluation of Equation (34) indicates that $|G|$ is maximized when $v = 0$ and maintaining stability requires that

$$\mathbf{C}_{\max} \leqslant 1 \tag{37}$$

The relative amplitude is defined as the ratio of the numerical to the exact solution amplitudes, i.e.

$$\varepsilon_{\mathrm{a}} = \frac{|G|}{\mathrm{e}^{-v\Delta t k_i^2}} \tag{38}$$

while the relative phase is similarly defined as the ratio of the numerical and exact phases

$$\varepsilon_{\mathrm{p}} = \frac{\tan^{-1}(B_G/A_G)}{-u\Delta t k_i} \tag{39}$$

Figure 1 compares $\varepsilon_{\mathrm{a}}$ with $\mathbf{C}_{\max} = 0.9$ and $v = 0$, as a function of $2N/i$, which can be interpreted as the number of cells per wave length. In the non-uniform cases, $r = 2$ and the grid is coarsened

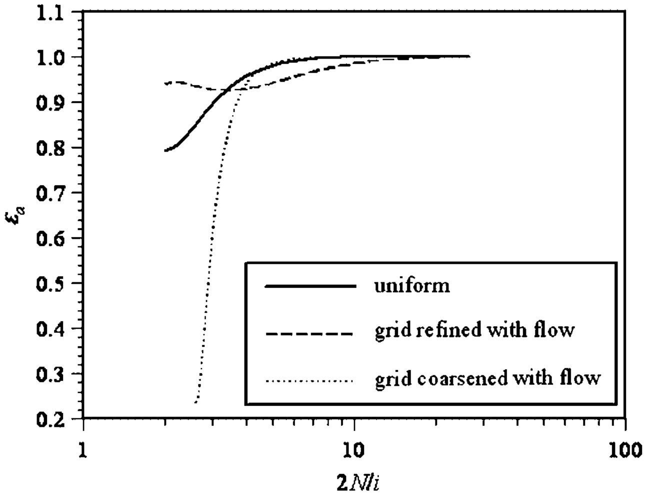

Figure 1. Relative amplitude, $\varepsilon_{\mathrm{a}}$, with $\mathbf{C}_{\max} = 0.9$ and $v = 0$.

DOI: 10.1002/fld

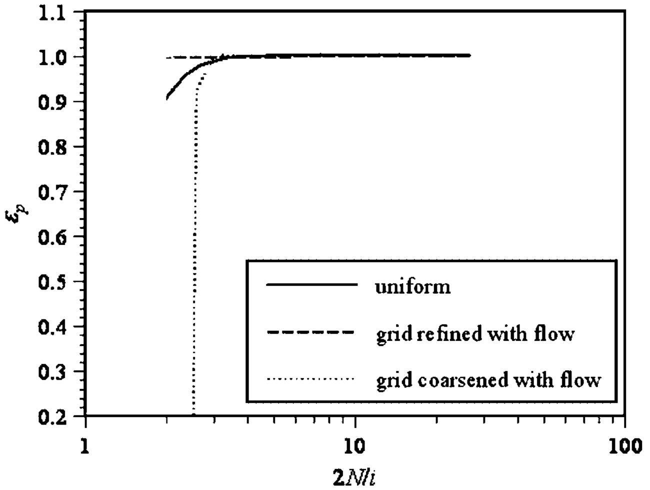

Figure 2. Relative phase, $\varepsilon_p$, with $\mathbf{C}_{\max} = 0.9$ and $\nu = 0$.

and refined in the flow direction. The uniform grid case is also plotted and shows that refining the grid in the flow direction slightly increases dissipation for $2N/i = 3$–$20$, which confirms the previous result given by Equation (32). Coarsening the grid is more dissipative only for $2N/i < 4$ and is very slightly less dissipative for greater values of $2N/i$. Figure 2 shows the corresponding $\varepsilon_p$ and reveals little difference except for the smallest $2N/i < 3$. Similar results were obtained for $\nu > 0$ and are therefore not presented.

## NUMERICAL EFFICIENCY

The computational effort for the hydrostatic portion of the model does not vary significantly with vertical grid variability. However, the solution of Equation (16) contributes a significant amount to the overall computational effort of the non-hydrostatic model. The biconjugate gradient (BCG) method is an attractive method for solving large, sparse systems of linear equations such as Equation (16) because the method only requires vector–matrix multiplications [17, 18]. For symmetric, positive definite matrices, the BCG method is equivalent to the conjugate gradient (CG) method. A symmetric, positive definite matrix is weakly diagonally dominant and therefore does not require pivoting when using a direct method to invert it [19]. The CG method converges at the rate given by [20]

$$\frac{E_i}{E_0} \leqslant 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^i \tag{40}$$

where $i$ is the iteration number and the error $E$ is defined as

$$E_i = \sqrt{(\mathbf{p}-\mathbf{p}_i)^{\mathrm{T}}\mathbf{A}(\mathbf{p}-\mathbf{p}_i)} \tag{41}$$

and the zero subscript denotes the initial guess of the solution. The condition number is defined as

$$\kappa = \|\mathbf{A}\|\,\|\mathbf{A}^{-1}\| \tag{42}$$

and for real matrices

$$\|\mathbf{A}\| = \sqrt{\lambda_{\max}(\mathbf{A}^{\mathrm{T}}\mathbf{A})} \tag{43}$$

where $\lambda_{\max}(\mathbf{A}^{\mathrm{T}}\mathbf{A})$ is the largest eigenvalue of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$. If $\mathbf{A}$ is symmetric, $\kappa$ becomes

$$\kappa = \frac{\lambda_{\max}(\mathbf{A})}{\lambda_{\min}(\mathbf{A})} \tag{44}$$

where $\lambda_{\min}$ denotes the smallest eigenvalue. Equation (40) indicates that an $\mathbf{A}$ with $\kappa \gg 1$ is ill-conditioned and will require more iterations to converge to a desired accuracy.

For typical large-scale problems, the $a$ coefficients in Equation (16) associated with vertical gradients are much larger than the $a$'s associated with lateral gradients. Therefore, insight into the computational efficiency of the model can be gained by considering a 1D problem with uniform flow in the $x$ and $y$ directions and constant $D$ and $h$. Equation (14) reduces to the following form:

$$-\frac{1}{\Delta\sigma_l \Delta\sigma_{l-1/2}} p_{l-1} + \left( \frac{1}{\Delta\sigma_l \Delta\sigma_{l-1/2}} + \frac{1}{\Delta\sigma_l \Delta\sigma_{l+1/2}} \right) p_l - \frac{1}{\Delta\sigma_l \Delta\sigma_{l+1/2}} p_{l+1} = DR_l \tag{45}$$

If the discretization is uniform, $\Delta\sigma = 1/n_\sigma$ (where $n_\sigma$ is the number of cells in the vertical direction) and Equation (45) can be written as

$$-p_{l-1} + 2p_l - p_{l+1} = \frac{DR_l}{n_\sigma^2} \tag{46}$$

$p_1$ is the pressure in the cell adjacent to the free surface and $p_{n_\sigma}$ is the pressure adjacent to the bottom. At the free surface, the Dirichlet BC is applied by setting $p_0 = 0$. At the bottom, $\partial p/\partial\sigma = 0$, which is implemented as $p_{n_\sigma+1} = p_{n_\sigma}$. Assembly of Equation (46) over the vertical direction yields the following $n_\sigma \times n_\sigma$, symmetric, tridiagonal form for $\mathbf{A}$

$$\mathbf{A} = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & \ddots & \ddots & \\ & & \ddots & 2 & -1 \\ & & & -1 & 1 \end{pmatrix}$$

and the eigenvalues of $\mathbf{A}$ are

$$\lambda_l = 4\sin^2 \frac{(2l-1)\pi}{2(2n_\sigma + 1)}, \quad l = 1 \text{ to } n_\sigma \tag{47}$$

The previous free surface BC is only $O(\Delta\sigma)$ accurate. The $O(\Delta\sigma^2)$ accurate BC is the Neumann type with $p_0 = -p_1$, which yields the following form for $\mathbf{A}$:

$$\mathbf{A} = \begin{pmatrix} 3 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & \ddots & \ddots & \\ & & \ddots & 2 & -1 \\ & & & -1 & 1 \end{pmatrix}$$

This matrix is more diagonally dominant in row 1 than the previous form and its eigenvalues are

$$\lambda_l = 4\sin^2\frac{(2l-1)\pi}{4n_\sigma}, \quad l = 1 \text{ to } n_\sigma \tag{48}$$

Preconditioning $\mathbf{A}$ can also be used to accelerate convergence, which is accomplished by solving the system

$$\tilde{\mathbf{A}}^{-1}\mathbf{A}\mathbf{p} = \tilde{\mathbf{A}}^{-1}\mathbf{R} \tag{49}$$

instead of Equation (16). $\tilde{\mathbf{A}}$ is the preconditioner matrix and its selection is an active area of research. In general, $\mathbf{A}^{-1}\mathbf{A} \approx \mathbf{I}$ to accelerate convergence and $\tilde{\mathbf{A}}$ should be easy to invert for computational efficiency. One simple choice is to select

$$\tilde{\mathbf{A}} = \text{diag}(\mathbf{A}) \tag{50}$$

which is equivalent to dividing each equation by its diagonal term. Therefore, the modified matrix $\tilde{\mathbf{A}}^{-1}\mathbf{A}$ consists of all 1's on the diagonal and because of the weak diagonal dominance of $\mathbf{A}$, all off-diagonal terms of the modified matrix are less than 1. The $\lambda_l$'s for the preconditioned matrix have been numerically computed. Figure 3 shows the relative $\kappa$ for the Neumann BC and the preconditioned Neumann BC, which have been normalized by $\kappa$ for the Dirichlet BC. This figure illustrates that the preconditioned Neumann method reduces $\kappa$ by 35% for small $n_\sigma$, but the reduction steadily decreases to less than 10% for $n_\sigma > 30$.

Equation (40) may be used to compare the number of iterations needed by the CG method to attain a given accuracy and therefore indicate the savings in computational effort. For example, the following equation compares the required iterations for the Neumann BC ($i_\text{N}$) to the Dirichlet BC ($i_\text{D}$):

$$\frac{i_\text{N}}{i_\text{D}} = \frac{\ln(\sqrt{\kappa_\text{D}}-1)/(\sqrt{\kappa_\text{D}}+1)}{\ln(\sqrt{\kappa_\text{N}}-1)/(\sqrt{\kappa_\text{N}}+1)} \tag{51}$$

where $\kappa_\text{N}$ and $\kappa_\text{D}$ are the condition numbers for the Neumann and Dirichlet BCs, respectively. The ratio $i_\text{N}/i_\text{D}$ is defined as the relative effort and is plotted in Figure 4, which shows that for small $n_\sigma$ the preconditioned Neumann BC requires about 78% of the iterations that the Dirichlet BC needs to attain the same level of convergence. For $n_\sigma > 30$ the relative effort grows to more than 96%. Note that the trend in Figure 4 closely follows the trend in Figure 3 as expected from Equation (51).

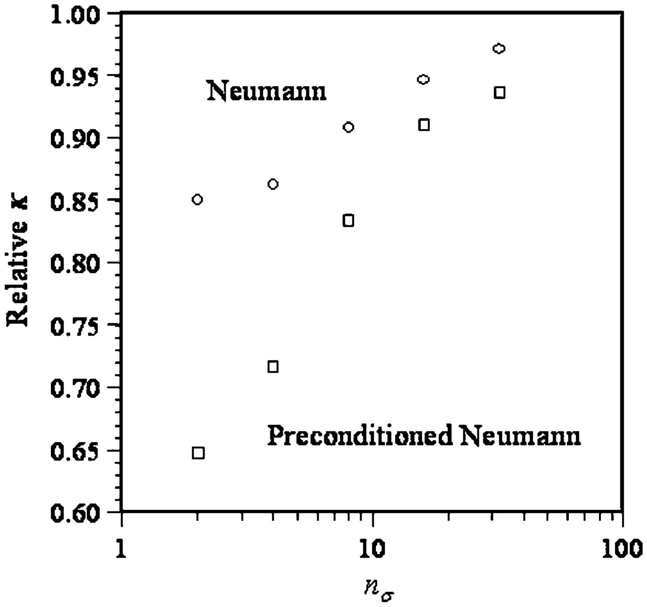

Figure 3. Condition number, $\kappa$, relative to the non-preconditioned Dirichlet free surface BC as a function of $n_\sigma$.

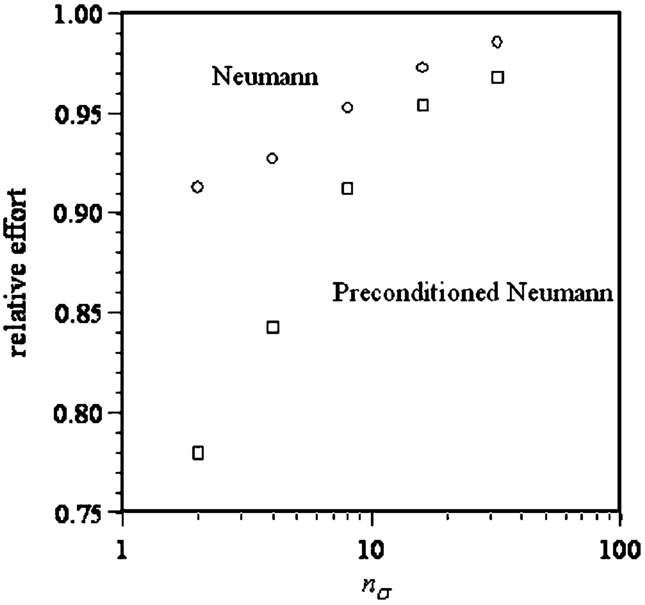

Figure 4. Relative effort as a function of $n_\sigma$.

The previous analysis revealed that the use of the Neumann BC yields a lower $\kappa$ because this increases the $p_1$ coefficient and therefore the diagonal dominance of row 1 of $\mathbf{A}$. Locally refining the grid at the free surface would further increase the diagonal dominance of row 1 and should therefore lead to improved convergence. If $\Delta\sigma_0 = \Delta\sigma_1$ and $\Delta\sigma_{n_\sigma+1} = \Delta\sigma_{n_\sigma}$, assembly of Equation (45) yields the resulting non-symmetric form for $\mathbf{A}$

$$\mathbf{A} = \begin{pmatrix} 2/\Delta\sigma_1^2 + b_1 & -b_1 & & & \\ a_2 & a_2 + b_2 & -b_2 & & \\ & -a_3 & \ddots & \ddots & \\ & & \ddots & a_{n_\sigma - 1} + b_{n_\sigma - 1} & -b_{n_\sigma - 1} \\ & & & -a_{n_\sigma} & a_{n_\sigma} \end{pmatrix}$$

where

$$a_l = \frac{2}{\Delta\sigma_l(\Delta\sigma_l + \Delta\sigma_{l-1})} \tag{52}$$

$$b_l = \frac{2}{\Delta\sigma_l(\Delta\sigma_l + \Delta\sigma_{l+1})} \tag{53}$$

Figure 5 shows plots of the relative $\kappa$ (which have been normalized by $\kappa$ for a uniform grid) as a function of $n_\sigma$ for $r = 1, 2$ and 3. For the grids refined at the free surface, the reductions in

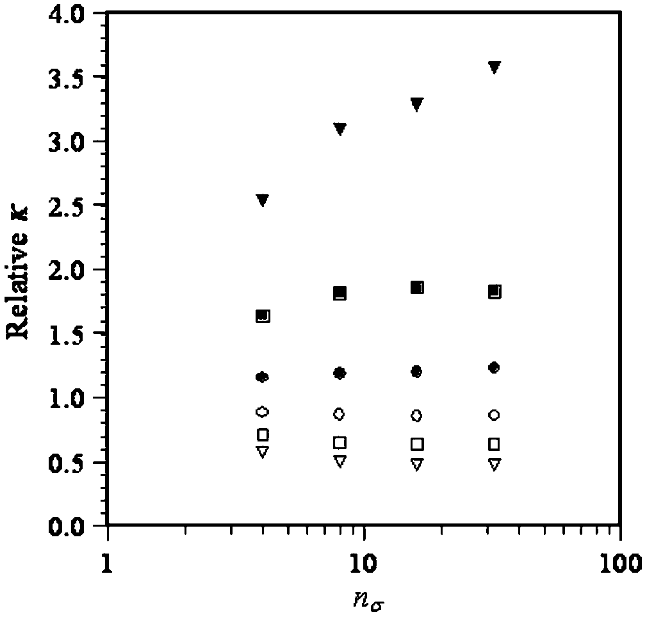

Figure 5. $\kappa$ relative to a uniform grid as a function of $n_\sigma$ and $r$ for grids refined at the bed and free surface. Open symbols denote grids refined at the free surface while closed symbols denote grids refined at the bed. The circles represent $r = 1$, squares $r = 2$, and triangles $r = 3$.

DOI: 10.1002/fld

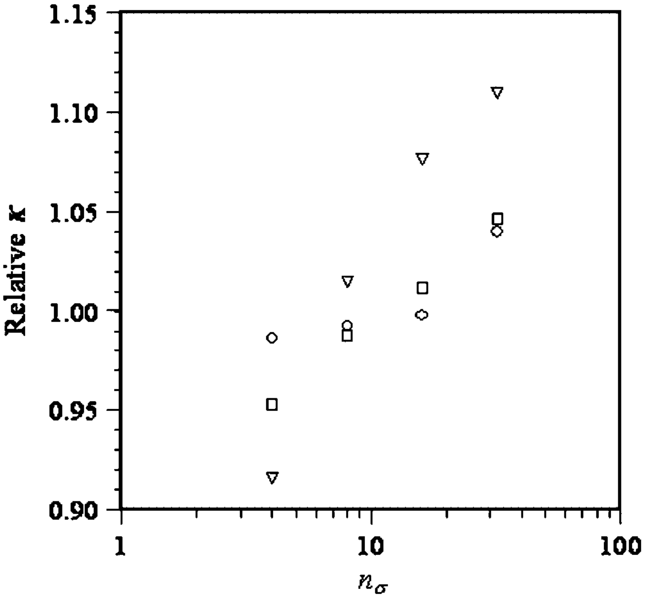

Figure 6. $\kappa$ relative to a uniform grid as a function of $n_\sigma$ and $r$ for grids simultaneously refined at the bed and free surface. The circles represent $r = 1$, squares $r = 2$, and triangles $r = 3$.

$\kappa$ are approximately 14% for $r = 1$, 36% for $r = 2$ and 52% for $r = 3$. Figure 5 shows the same plots for the grid refined at the bottom. In these cases, $\kappa$ increases relative to a uniform grid and for $r = 1$ the increase is almost 20%. For $r = 2$ the increase is roughly 90%, while for $r = 3$ the relative increase grows with $n_\sigma$ and is over 250% for $n_\sigma > 30$. In these cases, $\mathbf{A}$ is non-symmetric and therefore Equation (51) cannot be used to estimate the corresponding effect on computational expense. However, the previous analysis of the CG method indicated that the trend in relative $\kappa$ closely matched the trend in relative effort. If this relationship is true in the present case, then a severe computational penalty would be expected when refining the grid at the bed, while a more modest savings is expected when refining the grid at the free surface.

Turbulent flow exhibits sharp velocity gradients near the bed and is an obvious case where it would be beneficial to refine the grid near the bed to reduce the local truncation error. One possible way to avoid the computational penalty is to simultaneously refine the grid at the bed and free surface. In other words, refine the grid at the bed and coarsen it upwards until $f = S/2$ at which point the grid is refined again up to the free surface. Figure 6 shows the plots of the relative $\kappa$ in this case and it is seen that it remains close to 1 regardless of $r$. However, for large $n_\sigma$, the relative $\kappa$ does grow larger than 1, which indicates that refining the grid at the free surface does not completely negate the adverse effects of refining the grid at the bottom.

## SCALAR TRANSPORT

Some straightforward test problems are now solved to examine the effect of using a non-uniform grid on computational effort and solution accuracy. The first problem is the advection of

DOI: 10.1002/fld

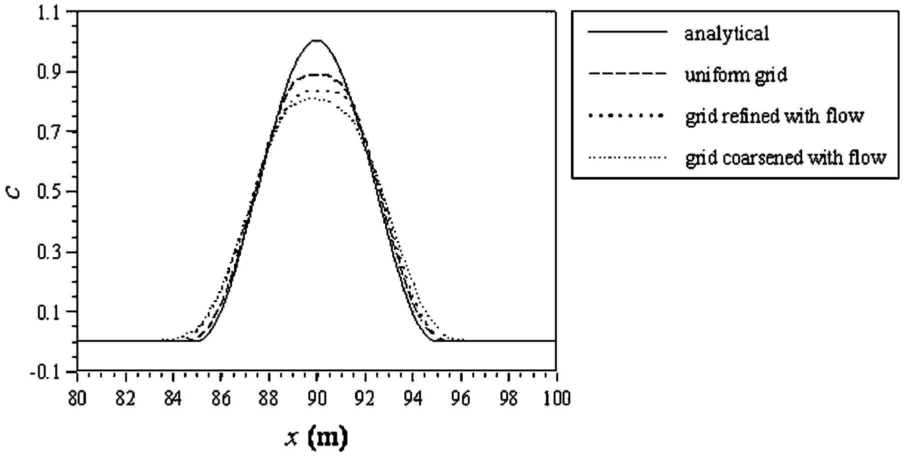

Figure 7. Comparison of scalar transport predictions of a smooth pulse at $t = 75$ s.

a conservative scalar in a 100 m long channel. The initial concentration is given as

$$c(x, 0) = \cos^2 \frac{\pi(x - 15)}{10} \quad \text{for } 10\,\text{m} \leqslant x \leqslant 20\,\text{m} \tag{54}$$

The model was run for 75 s with a constant velocity equal to 1 m/s. In this case, there is no obvious reason to use a non-uniform grid so this test is presented to show the practical effects of grid variability on an unsteady simulation. Figure 7 compares predictions with 200 cells and $\Delta t = 0.25$ s. The uniform grid prediction is the most accurate, followed by the grid refined with the flow with $r = 2$. The grid coarsened with the flow (also with $r = 2$) is the least accurate, which is expected since the grid is the coarsest at the location of comparison. Figure 7 confirms the previous analysis in that the variable grid solutions are more dissipative than the uniform grid solution, but possess similar phase accuracy. Figure 8 compares $L_2$ errors as a function of $1/N$, which is defined as

$$L_2(c) = \sqrt{\frac{1}{N} \sum_{j=1}^{N} (c - c_j)^2} \tag{55}$$

where $c$ is the exact solution and $c_j$ is the numerical prediction. Figure 8 shows that the uniform grid converges at the expected rate of $N^{-2}$, while the grid refined with the flow converges at a rate of $N^{-1.87}$ and the grid coarsened with the flow converges as $N^{-1.83}$. Replacing the smooth initial condition given by Equation (54) with a square pulse ($c = 1$) severely hinders the convergence of the model regardless of the grid choice. Figure 9 compares the predictions and once again the uniform grid is the most accurate, followed by the grid refined with the flow. Figure 10 compares the $L_2$ errors and in this case the model converges at a rate of $N^{-0.4}$ regardless of the grid. The discontinuities in this problem cause large errors to persist at the leading and trailing edges of the

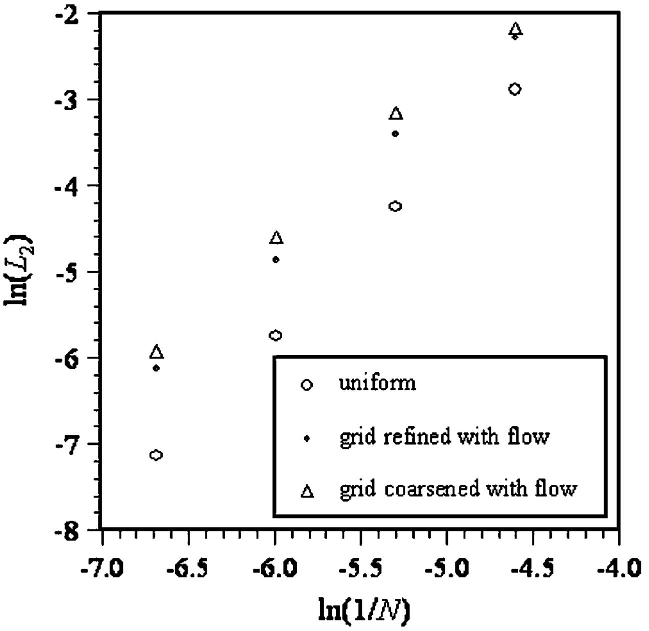

Figure 8. $L_2(c)$ as a function of $1/N$ for scalar transport of a smooth pulse.

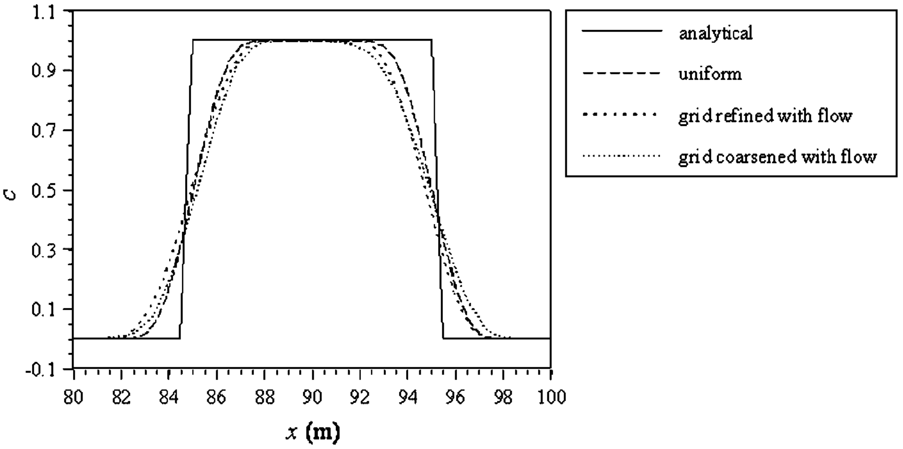

Figure 9. Comparison of scalar transport predictions of a sharp pulse at $t = 75$ s.

pulse, which very slowly decrease with increasing grid resolution. This test shows that although the variable grid predictions are less accurate than the uniform grid solution, the order of accuracy (or convergence rate) of the model is only slightly reduced.

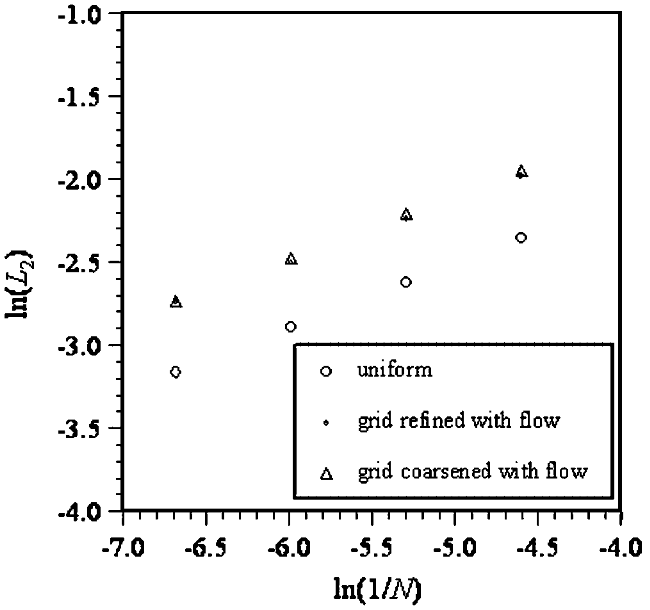

Figure 10. $L_2(c)$ as a function of $1/N$ for scalar transport of a sharp pulse.

## CHANNEL FLOW

The next test problem is steady, 2D, hydrostatic, turbulent flow in a 100 m long by 1 m wide channel with a bottom slope equal to 0.001. In this case, the large velocity gradient near the bed provides motivation for using a finer grid spacing to reduce the local truncation error in that region. At the upstream boundary, a flow rate equal to 1 $\mathrm{m}^3/\mathrm{s}$ is input while the downstream boundary is open. The channel bed roughness is 0.022 m and $v = 1 \times 10^{-6}\,\mathrm{m}^2/\mathrm{s}$. Under these conditions, the normal flow depth is approximately 1.26 m and the depth averaged velocity, $\overline{u} = 0.77\,\mathrm{m/s}$.

The model was run for 200 s with 1500 timesteps, $\Delta x = 1$ m, and the $k$–$\omega$ turbulence model [21] was used for closure. The results could be compared with semi-empirical models such as Coles' Law of the Wake, but this would confuse the analysis since the mathematical error of the turbulence model is mixed with the truncation error of the discretization. In addition, the semi-analytical expressions are not consistent with a zero shear stress free surface BC, which would cause a fixed discrepancy at that location regardless of vertical grid resolution. Therefore, model predictions are compared with the 'converged' numerical solution obtained with a uniform grid with $n_\sigma = 64$. Figure 11 compares predictions of $u$ for $n_\sigma = 8$ with a uniform grid and the grid refined near the bottom with varying $r$ at 50 m from the upstream end of the channel. As expected, the solution computed with the greatest grid refinement is closest to the converged solution.

Figure 12 compares the $L_2(u)$ errors of the predictions along with least squares regression lines for varying $r$. The figure shows that the uniform grid predictions converge at a rate of $n_\sigma^{-0.46}$, which is consistent with the sharp pulse scalar transport case. However, for $r = 1, 2$ and 3, the

DOI: 10.1002/fld

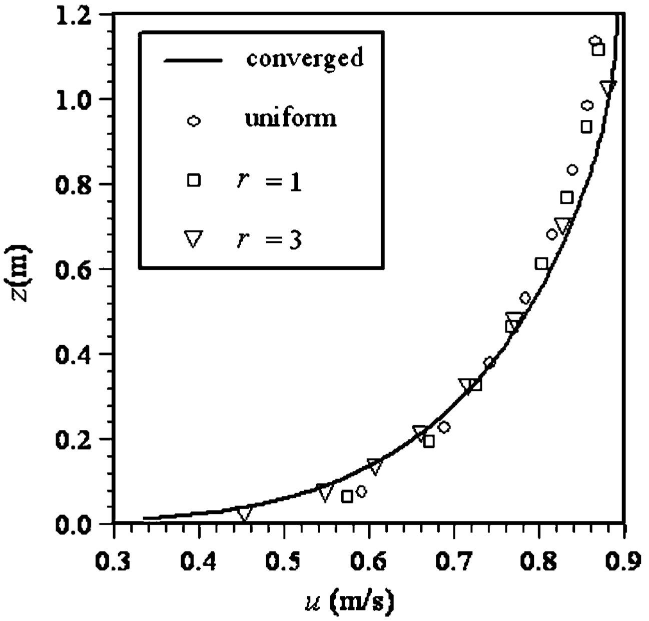

Figure 11. Comparison turbulent channel flow velocity predictions at $x = 50\,\text{m}$ with $n_\sigma = 8$ and varying $r$.

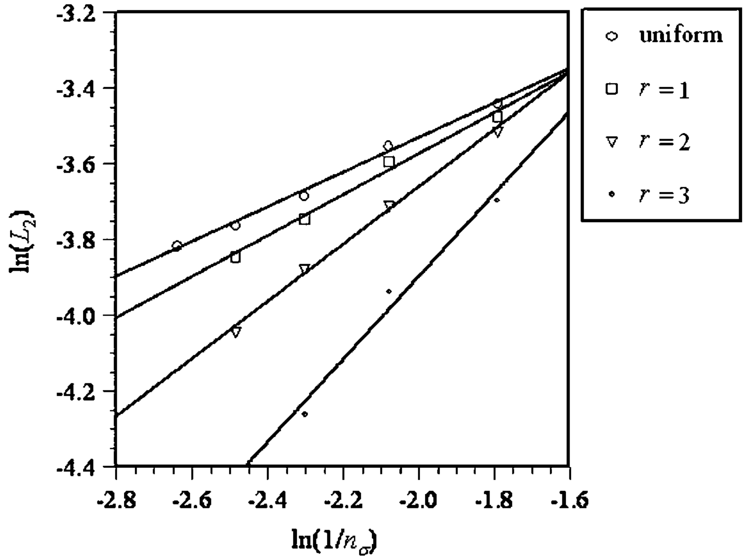

Figure 12. $L_2(u)$ as a function of $1/n_\sigma$ and $r$ for turbulent channel flow.

DOI: 10.1002/fld

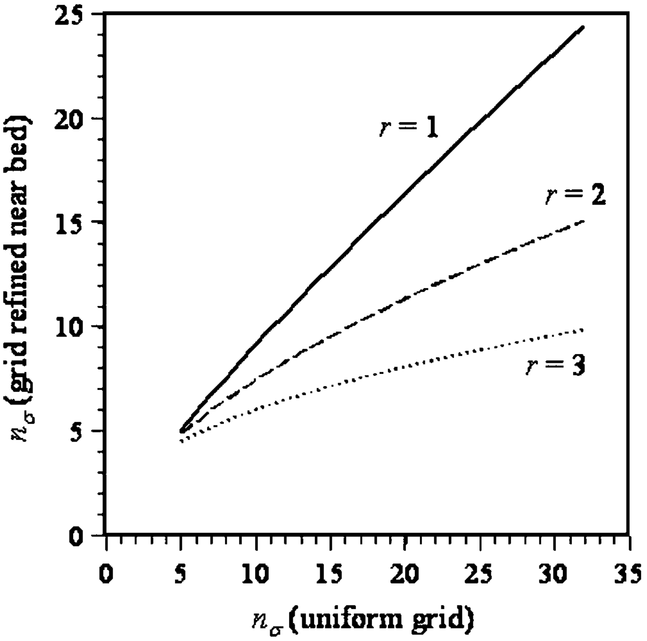

Figure 13. Plot of $n_\sigma$ needed by the non-uniform grid to match the $L_2(u)$ error of the uniform grid prediction.

convergence rates are $n_\sigma^{-0.55}$, $n_\sigma^{-0.76}$ and $n_\sigma^{-1.09}$, respectively. In this case, refining the grid in the region with sharp velocity gradients greatly reduces the large local truncation error near the bed while slightly increasing the smaller local truncation error near the free surface. This reduces the global error and increases the convergence rate. For a given uniform grid $n_\sigma$, the regression lines may be used to determine $n_\sigma$ for the refined grid that yields the same amount of error. The results are plotted in Figure 13, which shows that $r = 1$ yields a slight reduction in $n_\sigma$, but $r = 3$ yields a much greater reduction. Indeed, results not presented here revealed that further increasing of $r$ yielded even faster convergence, so much so that the converged solution with $n_\sigma = 64$ was no longer an appropriate benchmark. Furthermore, the vertical velocity at an impermeable bed is zero so the Courant stability condition is never violated regardless of the vertical grid spacing. However, if aggressive vertical grid refinement is used, then vertical diffusion must be implicitly integrated to preserve numerical stability.

## WAVE PROPAGATION

The channel flow simulation illustrates the effects of a non-uniform grid in the vertical direction, but with a negligible vertical velocity. Now the model is used to simulate inviscid, non-hydrostatic, wave propagation with a significant vertical velocity. However, in this case there are no strong gradients of flow variables in the vertical direction, so the only reason to employ a variable vertical grid is to reduce computational expense by refining the grid near the free surface. The channel is

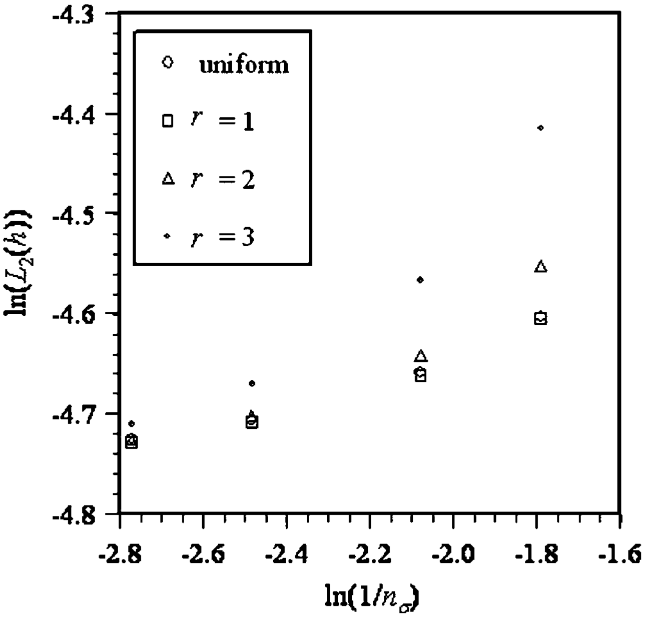

Figure 14. $L_2(h)$ as a function of $1/n_\sigma$ and $r$ for wave propagation.

100 m long, 1 m wide, and at the inflow boundary, fifth order stream function wave theory [22] is used to specify $h$, $u$ and $w$, while the other boundary is open. The initial depth is 1 m, the wave period is 5 s and the height is 0.2 m, which corresponds to a wavelength equal to 15.7 m. The model was run for 50 s with $\Delta t = 0.02$ s and $\Delta x = 0.1$ m, which corresponds to 250 timesteps per period and 157 cells per wavelength. Model predictions of $h$, $u$ at the bottom ($z = -1$ m), and $w$ at the surface ($z = h$) were compared to the analytical solution as a function of $t$ at the middle of the channel ($x = 50$ m).

Figure 14 compares $L_2(h)$ as a function of $1/n_\sigma$ for a uniform vertical grid as well as surface refined grids with $r = 1, 2$ and 3. In all cases, the errors decrease with increasing $n_\sigma$ but at different rates. For small $n_\sigma$, larger values of $r$ yield greater error. This is expected because of the previous analysis (Equations (29) and (32)) that showed the truncation error increases with grid stretching (increasing $r$). Also, unlike the previous case, there are no relatively large flow gradients near the free surface as compared to the bottom. Therefore, refining the grid in that region does not reduce the global truncation error. For larger $n_\sigma$, the differences become smaller, which was also demonstrated by Equations (29) and (32). $L_2(u)$ at the bottom and $L_2(w)$ at the surface showed similar trends and are therefore not presented.

Figure 15 compares the relative effort of the simulations as a function of $r$ and $n_\sigma$. In this case, relative effort is defined as the computational time required by the simulation normalized by the time for the corresponding uniform grid simulation. This figure shows a modest computational savings occurs when refining the grid near the surface. For $n_\sigma > 20$, there is a reduction in computational effort by about 10% for $r = 2$ and 15% for $r = 3$. Figure 15 also shows that there is a big computational penalty for refining the grid near the bed. For $n_\sigma > 20$, there is a 30% increase in computational time for $r = 2$ and a 70% increase for $r = 3$. Also shown in Figure 15 is the

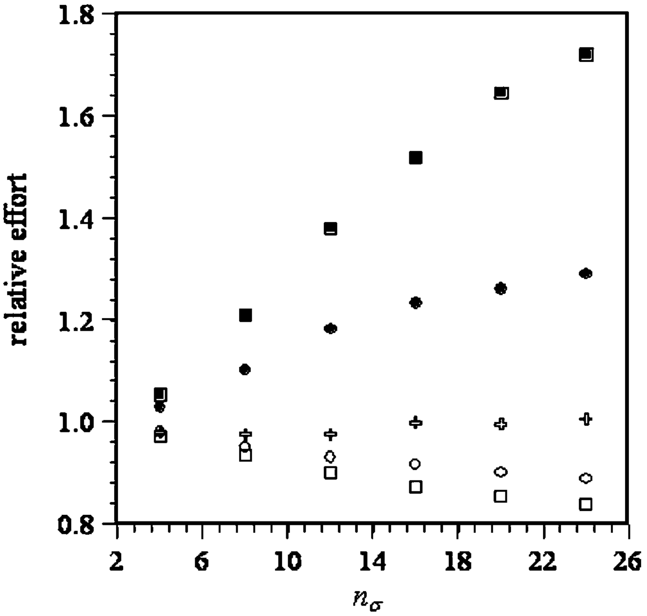

Figure 15. Plot of computational effort as a function of $n_\sigma$ and $r$ for wave propagation. Open symbols denote grids refined at the free surface while closed symbols denote grids refined at the bed. The circles represent $r = 2$ and squares $r = 3$. The crosses represent the grid simultaneously refined at the bed and free surface with $r = 2$.

relative effort when simultaneously refining the grid at the bed and free surface with $r = 2$. Using $r = 3$ yielded nearly identical results and is therefore not presented. The figure shows that there is virtually no difference in computational effort when compared to the uniform grid simulation using the same $n_\sigma$. However, for a given $n_\sigma$, simultaneous refinement will not yield the same resolution at the bed as bed grid refinement alone. Figure 16 shows plots of the required $n_\sigma$ for simultaneous refinement that yields the same grid spacing at the bed as bed refinement alone. For $r = 2$ and $n_\sigma = 10$, simultaneous refinement requires approximately $n_\sigma = 15$, while using $r = 3$ requires $n_\sigma = 23$. However, Figure 15 shows that despite using a larger $n_\sigma$, there would still be a 15% computational savings for $r = 2$ and a 33% savings for $r = 3$.

Figure 15 shows that increasing the surface grid refinement increases the computational savings. However, reducing the grid spacing at the surface must be done cautiously. Unlike the previous case in which the velocity was zero at the location of greatest grid refinement, in this case the greatest grid refinement occurs at the free surface where the vertical velocity is often the greatest. As the grid spacing is reduced, the Courant number increases (for fixed $\Delta t$) and may cause numerical instability. In addition, the Von Neumann analysis showed that an increasing Courant number reduces $\varepsilon_\mathrm{a}$ (increases numerical dissipation). Figure 17 shows plots of the ratio of the uniform grid spacing to the minimum grid spacing as a function of $n_\sigma$ and $r$. For $r = 1$ the ratio is approximately 1.2 for $n_\sigma > 5$, while for $r = 2$ it is 1.8 and for $r = 3$ it is 3.3. Therefore, the maximum vertical Courant number (assumed to occur at the free surface) for a uniform grid simulation will be more than tripled when using $r = 3$ for large $n_\sigma$.

DOI: 10.1002/fld

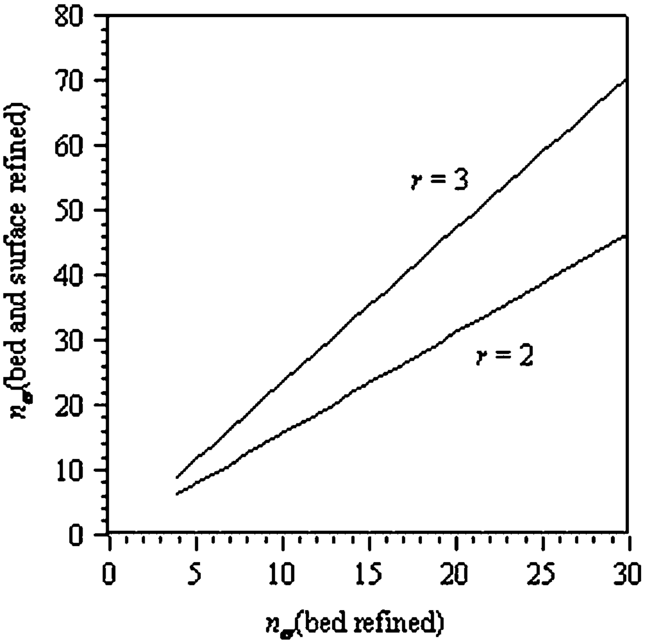

Figure 16. Plot of the required $n_\sigma$ when simultaneously refining the grid at the bed and surface that yields the same grid size at the bed when refining the grid near the bed only.

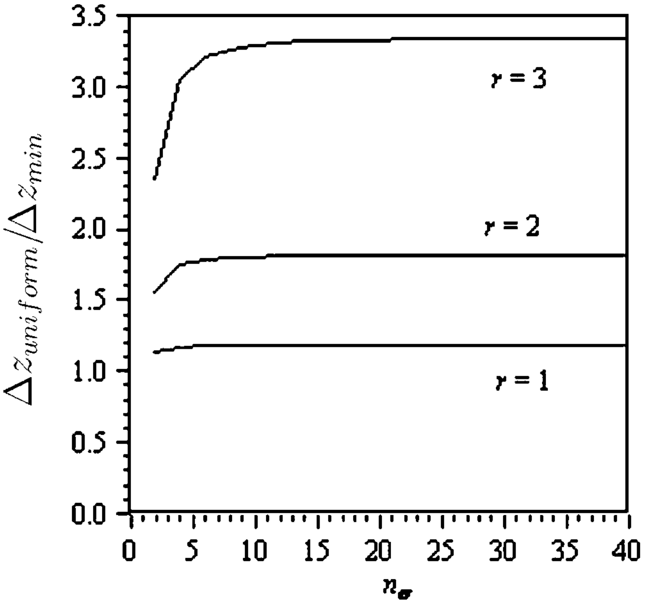

Figure 17. Plot of the ratio of the uniform grid spacing, $\Delta z_{\text{uniform}}$, to the minimum grid spacing, $\Delta z_{\text{min}}$, as a function of $n_\sigma$ and $r$.

## SUMMARY AND CONCLUSIONS

Refining the grid in regions of sharp flow gradients yields a more accurate solution than a uniform grid solution using the same number of cells. Alternatively, a variable grid model can attain the same accuracy as a uniform grid model, but using fewer cells and therefore less computational effort. In this paper it was shown by analysis and example that the hyperbolic sine stretching function preserves second-order accuracy in the sense that the truncation error is proportional to $N^{-2}$. However, even the slightest grid stretch introduces a dissipative error and stretching in the flow direction may introduce numerical instability. It was also shown that selecting $r \ll 2n_\sigma$ mitigates these problems, while the Von Neumann analysis showed little impact of grid stretching on phase error. The smooth pulse scalar transport simulations confirmed these results, while the sharp pulse simulations converged at a much slower rate regardless of the grid uniformity.

It was also demonstrated that as $n_\sigma$ increases, the condition number of the pressure Poisson matrix grows rapidly regardless of the vertical discretization. This reduces the convergence rate of the BCG method and therefore increases computational cost. The use of a Neumann free surface BC and preconditioning the Poisson matrix reduces the required iterations for convergence by 10–20% for $n_\sigma<10$, but the reduction is less than 5% for $n_\sigma=30$. It was shown that refining the grid near the free surface also reduces the condition number of the pressure Poisson matrix. For $r=1$, the reduction was approximately 15%, while for $r=3$ it was more than 50%. These reductions are substantial, but more importantly they remain even for large $n_\sigma$. The wave propagation simulations confirmed this finding by showing that the computational cost reductions grew with increasing $n_\sigma$ and increasing $r$. However, overly aggressive grid refinement at the free surface may cause a violation of the Courant stability condition and therefore must be done with caution if large surface velocities are expected.

Refining the grid at the bed has the opposite effect and greatly increases the Poisson matrix condition number and computational effort. However, it was shown that refining the grid at the bottom greatly improves the accuracy of the model when simulating a turbulent current. For example, the accuracy of a simulation using 30 uniform cells could be matched using $r=3$ and $n_\sigma=10$. For hydrostatic flow, there is no Poisson equation to solve and therefore no computational penalty for refining the grid near the bed. Also, there is no concern about violating the vertical Courant stability condition since the vertical velocity at the bed is zero. However, vertical diffusion must be implicitly integrated to preserve model stability when using extreme grid stretching. For non-hydrostatic flow, it was shown that simultaneously refining the grid at the bed and free surface can provide the same near bed resolution as refining near the bed only, and still yield a computational savings over the corresponding uniform grid simulation. Therefore when simulating hydrostatic currents, it is recommended that near bed refinement only is used. For non-hydrostatic flow, simultaneous near bed and near free surface grid refinement is recommended. Stretching with $r=2$–3 yielded good results, but the Courant condition must be closely monitored.

Although the analysis was presented for a specific model that uses the BCG method to invert the pressure Poisson matrix, the results should at least qualitatively apply to any fractional step, non-hydrostatic model that uses an iterative matrix inversion method. For example, popular methods like Gauss–Seidel converge faster for matrices with a relatively smaller condition number.

## APPENDIX A

The coefficients of the Poisson pressure equation, Equation (14), are

$$a_1 = -s_{\xi\sigma(j-1/2,k,l)} - (1-\omega_{l-1/2})s_{\xi\sigma(j,k,l-1/2)}, \quad a_2 = -s_{\eta\sigma(j,k-1/2,l)} - (1-\omega_{l-1/2})s_{\eta\sigma(j,k,l-1/2)}$$

$$a_4 = s_{\eta\sigma(j,k+1/2,l)} + (1-\omega_{l-1/2})s_{\eta\sigma(j,k,l-1/2)}, \quad a_5 = s_{\xi\sigma(j+1/2,k,l)} + (1-\omega_{l-1/2})s_{\xi\sigma(j,k,l-1/2)}$$

$$a_6 = -s_{\xi\eta(j-1/2,k)} - s_{\xi\eta(j,k-1/2)}, \quad a_8 = s_{\xi\eta(j-1/2,k)} + s_{\xi\eta(j,k+1/2)}$$

$$a_{12} = s_{\xi\eta(j+1/2,k)} + s_{\xi\eta(j,k-1/2)}, \quad a_{14} = -s_{\xi\eta(j+1/2,k)} - s_{\xi\eta(j,k+1/2)}$$

$$a_{15} = s_{\xi\sigma(j-1/2,k,l)} + \omega_{l+1/2}s_{\xi\sigma(j,k,l+1/2)}, \quad a_{16} = s_{\eta\sigma(j,k-1/2,l)} + \omega_{l+1/2}s_{\eta\sigma(j,k,l+1/2)}$$

$$a_{18} = -s_{\eta\sigma(j,k+1/2,l)} - \omega_{l+1/2}s_{\eta\sigma(j,k,l+1/2)}, \quad a_{19} = -s_{\xi\sigma(j+1/2,k,l)} - \omega_{l+1/2}s_{\xi\sigma(j,k,l+1/2)}$$

$$a_3 = -(s_{\sigma\sigma(j,k,l-1/2)} + s_{\xi\sigma(j-1/2,k,l)} - s_{\xi\sigma(j+1/2,k,l)} + s_{\eta\sigma(j,k-1/2,l)} - s_{\eta\sigma(j,k+1/2,l)}) \tag{A1}$$

$$a_7 = -(s_{\xi\xi(j-1/2,k)} + s_{\xi\eta(j,k-1/2)} - s_{\xi\eta(j,k+1/2)} + \omega_{l-1/2}s_{\xi\sigma(j,k,l-1/2)} - (1-\omega_{l+1/2})s_{\xi\sigma(j,k,l+1/2)})$$

$$a_9 = -(s_{\eta\eta(j,k-1/2)} + s_{\xi\eta(j-1/2,k)} - s_{\xi\eta(j+1/2,k)} + \omega_{l-1/2}s_{\eta\sigma(j,k,l-1/2)} - (1-\omega_{l+1/2})s_{\eta\sigma(j,k,l+1/2)})$$

$$a_{11} = -(s_{\eta\eta(j,k+1/2)} + s_{\xi\eta(j+1/2,k)} - s_{\xi\eta(j-1/2,k)} + (1-\omega_{l+1/2})s_{\eta\sigma(j,k,l+1/2)} - \omega_{l-1/2}s_{\eta\sigma(j,k,l-1/2)})$$

$$a_{13} = -(s_{\xi\xi(j+1/2,k)} + s_{\xi\eta(j,k+1/2)} - s_{\xi\eta(j,k-1/2)} + (1-\omega_{l+1/2})s_{\xi\sigma(j,k,l+1/2)} - \omega_{l-1/2}s_{\xi\sigma(j,k,l-1/2)})$$

$$a_{17} = -(s_{\sigma\sigma(j,k,l+1/2)} + s_{\xi\sigma(j+1/2,k,l)} - s_{\xi\sigma(j-1/2,k,l)} + s_{\eta\sigma(j,k+1/2,l)} - s_{\eta\sigma(j,k-1/2,l)})$$

$$a_{10} = s_{\xi\xi(j-1/2,k)} + s_{\xi\xi(j+1/2,k)} + s_{\eta\eta(j,k-1/2)} + s_{\eta\eta(j,k+1/2)} + s_{\sigma\sigma(j,k,l-1/2)} + s_{\sigma\sigma(j,k,l+1/2)}$$

where

$$s_{\xi\xi} = D(\xi_x^2 + \xi_y^2), \quad s_{\xi\eta} = \frac{D}{4}(\xi_x\eta_x + \xi_y\eta_y), \quad s_{\eta\eta} = D(\eta_x^2 + \eta_y^2)$$

$$s_{\xi\sigma} = \frac{D}{2\Delta\sigma_t}(\xi_x\sigma_x + \xi_y\sigma_y), \quad s_{\eta\sigma} = \frac{D}{2\Delta\sigma_t}(\eta_x\sigma_x + \eta_y\sigma_y) \tag{A2}$$

$$s_{\sigma\sigma(l\pm1/2)} = \frac{D}{\Delta\sigma_l\Delta\sigma_{l\pm1/2}}(\sigma_x^2 + \sigma_y^2 + 1/D^2)$$

and $\Delta\sigma_t = \Delta\sigma_l + (\sigma_{l+1/2} + \sigma_{l-1/2})/2$, $\omega_{l-1/2} = \Delta\sigma_l/(\Delta\sigma_l + \Delta\sigma_{l-1/2})$ and $\omega_{l+1/2} = \Delta\sigma_l/(\Delta\sigma_l + \Delta\sigma_{l+1/2})$.

# APPENDIX B

The coefficients in Equation (22) are

$$
\begin{aligned}
C_1 &= -\frac{\mathbf{D}_{j+1/2}}{2}\\
C_2 &= 1+\frac{\mathbf{D}_{j+1/2}+\mathbf{D}_{j-1/2}}{2}\\
C_3 &= -\frac{\mathbf{D}_{j-1/2}}{2}\\
C_4 &= \tfrac{1}{2}[\mathbf{D}_{j+1/2}-\mathbf{C}_j(1-\mathbf{C}_j)\phi_1]\\
C_5 &= 1-\mathbf{C}_j-\tfrac{1}{2}\{\mathbf{D}_{j+1/2}+\mathbf{D}_{j-1/2}+\mathbf{C}_j[(1-\mathbf{C}_j)\phi_2-(1-\mathbf{C}_{j-1})\phi_4]\}\\
C_6 &= \mathbf{C}_j+\tfrac{1}{2}\{\mathbf{D}_{j-1/2}-\mathbf{C}_j[(1-\mathbf{C}_j)\phi_3-(1-\mathbf{C}_{j-1})\phi_5]\}\\
C_7 &= \frac{\mathbf{C}_j}{2}(1-\mathbf{C}_{j-1})\phi_6
\end{aligned}
\tag{B1}
$$

$\mathbf{D}_{j+1/2}=v\Delta t/(\Delta x_j\Delta x_{j+1/2})$ and $\mathbf{D}_{j-1/2}=v\Delta t/(\Delta x_j\Delta x_{j-1/2})$ are the diffusion numbers and the $\phi$ coefficients result from the data reconstruction in Equation (19) and are defined as

$$
\begin{aligned}
\phi_1 &= \frac{\Delta x_{j-1/2}\Delta x_j}{\Delta x_{j+1/2}(\Delta x_{j+1/2}+\Delta x_{j-1/2})}\\
\phi_2 &= \frac{(\Delta x_{j+1/2}-\Delta x_{j-1/2})\Delta x_j}{\Delta x_{j+1/2}\Delta x_{j-1/2}}\\
\phi_3 &= -\frac{\Delta x_{j+1/2}\Delta x_j}{\Delta x_{j-1/2}(\Delta x_{j+1/2}+\Delta x_{j-1/2})}\\
\phi_4 &= \frac{\Delta x_{j-3/2}\Delta x_{j-1}}{\Delta x_{j-1/2}(\Delta x_{j-1/2}+\Delta x_{j-3/2})}\\
\phi_5 &= \frac{(\Delta x_{j-1/2}-\Delta x_{j-3/2})\Delta x_{j-1}}{\Delta x_{j-1/2}\Delta x_{j-3/2}}\\
\phi_6 &= -\frac{\Delta x_{j-1/2}\Delta x_{j-1}}{\Delta x_{j-3/2}(\Delta x_{j-1/2}+\Delta x_{j-3/2})}
\end{aligned}
\tag{B2}
$$

## REFERENCES

1. Lin P, Liu PLF. A numerical study of breaking waves in the surf zone. *Journal of Fluid Mechanics* 1998; **359**:239–264.
2. Bradford SF. Numerical simulation of surf zone dynamics. *Journal of Waterway Port*, *Coastal*, *and Ocean Engineering* (ASCE) 2000; **126**(1):1–13.
3. Harlow FH, Welch JE. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids* 1965; **8**(12):2182–2189.
4. Hirt CW, Nichols BD. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics* 1981; **39**(1):201–225.
5. Phillips NA. A coordinate system having some special advantages for numerical forecasting. *Journal of Meteorology* 1957; **14**(2):184–185.
6. Mahadevan A, Oliger J, Street R. A nonhydrostatic mesoscale ocean model. Part II: Numerical implementation. *Journal of Physical Oceanography* 1996; **26**(9):1881–1900.
7. Lin P, Li CW. A $\sigma$-coordinate three-dimensional numerical model for surface wave propagation. *International Journal for Numerical Methods in Fluids* 2002; **38**(11):1045–1068.
8. Bradford SF. Godunov-based model for nonhydrostatic wave dynamics. *Journal of Waterway*, *Port*, *Coastal*, *and Ocean Engineering* (ASCE) 2005; **131**(5):226–238.
9. Li B, Fleming CA. Three-dimensional model of Navier–Stokes equations for water waves. *Journal of Waterway*, *Port*, *Coastal*, *and Ocean Engineering* (ASCE) 2001; **127**(1):16–25.
10. MacCracken MC, Bornstein RD. On the treatment of advection in flux formulations for variable grid models, with application to two models of the atmosphere. *Journal of Computational Physics* 1977; **23**:135–149.
11. Brown PS, Pandolfo JP. Numerical stability of the combined advection–diffusion equation with nonuniform spatial grid. *Monthly Weather Review* 1979; **107**(8):959–962.
12. Thompson JF, Mastin CW. Order of difference expressions in curvilinear coordinate systems. *Journal of Fluids Engineering* 1985; **107**:241–250.
13. Yin FL, Fung IY. Net diffusivity in ocean circulation models with nonuniform grids. *Journal of Geophysical Research* 1991; **96**(C6):10773–10776.
14. Marti O, Madec G, Delecluse P. Comment on net diffusivity in ocean circulation models with nonuniform grids by F. L. Yin and I. Y. Fung. *Journal of Geophysical Research* 1992; **97**(C8):12763–12766.
15. Treguier AM, Dukowicz JK, Bryan K. Properties of nonuniform grids used in ocean general circulation models. *Journal of Geophysical Research* 1996; **101**(C9):20877–20881.
16. Bradford SF. Stability and accuracy of a semi-implicit Godunov scheme for mass transport. *International Journal for Numerical Methods in Fluids* 2004; **45**:365–389.
17. Press WH, Teukolsky SA, Vetterling WT, Flannery BP. *Numerical Recipes in Fortran*. Cambridge University Press: New York, 1992.
18. Saad Y. *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company: Boston, 1996.
19. Strang G. *Introduction to Applied Mathematics*. Wellesley-Cambridge Press: Wellesley, 1986.
20. Golub GH, Van Loan CF. *Matrix Computations*. Johns Hopkins University Press: Baltimore, London.
21. Wilcox DC. *Turbulence Modeling for CFD*. Griffin Printing: Glendale, CA.
22. Dalrymple RA. A finite amplitude wave on a linear shear current. *Journal of Geophysical Research* 1974; **79**(30):4498–4504.

DOI: 10.1002/fld