INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2006; **51**:617–633
Published online 20 December 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1137

Applications of the artificial compressibility method for turbulent open channel flows

J. W. Lee[1,*,†], M. D. Teubner[1,‡], J. B. Nixon[2,§] and P. M. Gill[1,¶]

[1]*Applied Mathematics, The University of Adelaide, Adelaide, SA 5005, Australia*
[2]*United Water International Pty Ltd., GPO Box 1875, Adelaide, SA 5001, Australia*

SUMMARY

A three-dimensional (3-D) numerical method for solving the Navier–Stokes equations with a standard k–ε turbulence model is presented. In order to couple pressure with velocity directly, the pressure is divided into hydrostatic and hydrodynamic parts and the artificial compressibility method (ACM) is employed for the hydrodynamic pressure. By introducing a pseudo-time derivative of the hydrodynamic pressure into the continuity equation, the incompressible Navier–Stokes equations are changed from elliptic-parabolic to hyperbolic-parabolic equations. In this paper, a third-order monotone upstream-centred scheme for conservation laws (MUSCL) method is used for the hyperbolic equations. A system of discrete equations is solved implicitly using the lower–upper symmetric Gauss–Seidel (LU-SGS) method. This newly developed numerical method is validated against experimental data with good agreement. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: artificial compressibility method; hydrodynamic pressure; open channel flows; k–ε turbulence model

1. INTRODUCTION

As computer capabilities have increased exponentially during the last decade, 3-D hydrodynamic (or non-hydrostatic) pressure models [1–4] for free surface flows have been developed extensively. Because of the incompressibility of water, the time variation of density is ignored, which makes solving for the pressure difficult. Most of these models determine the hydrodynamic pressure using a fractional-step method after decomposing the pressure

*Correspondence to: J. W. Lee, Applied Mathematics, The University of Adelaide, Adelaide, SA 5005, Australia.
†E-mail: jong.lee@adelaide.edu.au
‡E-mail: mteubner@maths.adelaide.edu.au
§E-mail: john.nixon@uwi.com.au
¶E-mail: pgill@maths.adelaide.edu.au

Contract/grant sponsor: University of Adelaide
Contract/grant sponsor: United Water International Pty Ltd.

Received 14 April 2005
Revised 14 October 2005
Accepted 18 October 2005

into hydrostatic and hydrodynamic parts [3]. Usually, the hydrodynamic pressure is unknown because it is a function of the velocity field. In the first step of the fractional-step method velocities are calculated using the hydrodynamic pressure from the previous time step, which may cause mass conservation errors in the incompressible continuity equation. In the second step these errors can be eliminated by substituting corrected velocities (the velocities from the first step plus a hydrodynamic pressure correction) into the continuity equation. This second step results in an elliptic-type equation, called the pressure-Poisson equation, which can be solved iteratively. Usually, the solution procedure for this pressure-Poisson equation method is central processor unit (CPU) time expensive. Moreover, it can cause grid-scale pressure oscillations, known as a checker-board pressure field [5], because of pressure–velocity decoupling in a traditional collocated grid arrangement. As an alternative, the artificial compressibility method (ACM) can be considered [6]. The main advantage of the ACM is that the pressure can be coupled directly with the velocity field; this is possible only by adding a time derivative of pressure into the incompressible continuity equation, leading to changing the governing equations from elliptic-parabolic to hyperbolic-parabolic equations. Many authors [7–9] have applied this method successfully. In addition, the ACM has been applied to free surface flows by a number of authors [10, 11]. In this paper, we decompose the pressure into its hydrostatic and hydrodynamic parts, and the ACM is applied only to the hydrodynamic pressure.

2. MATHEMATICAL FORMULATION

For free surface flows, the hydrodynamic pressure P_d can be considered using the pressure decomposition

$$P = P_{\mathrm{h}} + P_d = \rho g(H - z) + P_d \tag{1}$$

where P is the total pressure; P_{h} is the hydrostatic pressure, as a function of the water surface elevation H and the vertical location z; ρ is the water density; and g is gravitational acceleration. In order to couple the hydrodynamic pressure with the velocity field directly, we add a time derivative of the hydrodynamic pressure into the incompressible continuity equation, leading to

$$\frac{\partial}{\partial t^*}\left(\frac{p_d}{\beta}\right) + \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{2}$$

where $p_d = P_d/\rho$; t^* is pseudo-time; u, v, and w are mean velocity components in the Cartesian x-, y-, and z-coordinates, respectively; and β is a coefficient that represents the wave speed of the hydrodynamic pressure. Note that (2) is different from other ACM models [7–9] which involve a time derivative of the total pressure. By adding the time derivative of the hydrodynamic pressure into (2), the physical meaning of the original continuity equation is modified, but it is recovered when numerical solutions reach steady state.

In a generalized coordinate system (ξ, η, ζ, τ), the incompressible Navier–Stokes equations with (1) and (2) and the standard k–ε equations [12] can be written in vector form as

$$\frac{\partial \mathbf{Q}}{\partial \tau} + \frac{\partial}{\partial \xi}(\mathbf{E} - \mathbf{E}_{\mathrm{v}}) + \frac{\partial}{\partial \eta}(\mathbf{F} - \mathbf{F}_{\mathrm{v}}) + \frac{\partial}{\partial \zeta}(\mathbf{G} - \mathbf{G}_{\mathrm{v}}) = \mathbf{S} \tag{3}$$

where

$$\mathbf{E}=\frac{1}{J}\begin{pmatrix}\beta U\\ uU+\xi_x p_d\\ vU+\xi_y p_d\\ wU+\xi_z p_d\\ kU\\ \varepsilon U\end{pmatrix};\quad \mathbf{F}=\frac{1}{J}\begin{pmatrix}\beta V\\ uV+\eta_x p_d\\ vV+\eta_y p_d\\ wV+\eta_z p_d\\ kV\\ \varepsilon V\end{pmatrix};\quad \mathbf{G}=\frac{1}{J}\begin{pmatrix}\beta W+\zeta_t p_d\\ u(W+\zeta_t)+\zeta_x p_d\\ v(W+\zeta_t)+\zeta_y p_d\\ w(W+\zeta_t)+\zeta_z p_d\\ k(W+\zeta_t)\\ \varepsilon(W+\zeta_t)\end{pmatrix} \tag{4}$$

$$\mathbf{Q}=\frac{1}{J}\begin{pmatrix}p_d\\ u\\ v\\ w\\ k\\ \varepsilon\end{pmatrix};\quad \mathbf{S}=\begin{pmatrix}0\\ -g\left\{\left(\frac{\xi_x}{J}H\right)_\xi+\left(\frac{\eta_x}{J}H\right)_\eta+\left(\frac{\zeta_x}{J}H\right)_\zeta\right\}\\ -g\left\{\left(\frac{\xi_y}{J}H\right)_\xi+\left(\frac{\eta_y}{J}H\right)_\eta+\left(\frac{\zeta_y}{J}H\right)_\zeta\right\}\\ 0\\ (\mathscr{P}-\varepsilon)/J\\ \left(c_{1\varepsilon}\frac{\varepsilon}{k}\mathscr{P}-c_{2\varepsilon}\frac{\varepsilon^2}{k}\right)\Big/J\end{pmatrix};$$

$$\begin{aligned}
\mathbf{E}_\mathbf{v}&=\frac{\nabla\xi\cdot\nabla\xi_i}{J}\frac{\nu_\mathrm{T}}{c_1}\mathbf{I}_\mathbf{m}\frac{\partial\mathbf{Q}}{\partial\xi_i}\\
\mathbf{F}_\mathbf{v}&=\frac{\nabla\eta\cdot\nabla\xi_i}{J}\frac{\nu_\mathrm{T}}{c_1}\mathbf{I}_\mathbf{m}\frac{\partial\mathbf{Q}}{\partial\xi_i}\\
\mathbf{G}_\mathbf{v}&=\frac{\nabla\zeta\cdot\nabla\xi_i}{J}\frac{\nu_\mathrm{T}}{c_1}\mathbf{I}_\mathbf{m}\frac{\partial\mathbf{Q}}{\partial\xi_i}\\
\mathbf{I}_\mathbf{m}&=\mathrm{diag}(0,1,1,1,1,1)\\
U&=\xi_x u+\xi_y v+\xi_z w\\
V&=\eta_x u+\eta_y v+\eta_z w\\
W&=\zeta_x u+\zeta_y v+\zeta_z w
\end{aligned} \tag{5}$$

$J=|\partial(\xi,\eta,\zeta)/\partial(x,y,z)|$ is the Jacobian of the transformation; $\nabla\xi_i=(\nabla\xi,\nabla\eta,\nabla\zeta)$ for $i=1,2,3$; k is the turbulent kinetic energy (TKE); ε is the dissipation rate; $\mathscr{P}$ is the turbulent production; and $c_1=1$ for the Navier–Stokes equation, σ_k for the k equation, and σ_ε for the ε equation.

The turbulent eddy viscosity v_T is determined using

$$v_T = c_\mu \frac{k^2}{\varepsilon} \tag{6}$$

Finally, the turbulence constants have been found to be $c_\mu = 0.09$, $c_{1\varepsilon} = 1.44$, $c_{2\varepsilon} = 1.92$, $\sigma_k = 1.0$, and $\sigma_\varepsilon = 1.3$ [12]. In this transformation it is assumed the horizontal space coordinates (ξ, η) are independent of time, leading to $\xi_t = \eta_t = 0$.

For free surface evolution, the original incompressible continuity equation is integrated over the depth, leading to the water surface equation

$$\frac{\partial}{\partial \tau}\left(\frac{h}{J}\right) + \frac{\partial}{\partial \xi}\left(\overline{U}\frac{h}{J}\right) + \frac{\partial}{\partial \eta}\left(\overline{V}\frac{h}{J}\right) = 0 \tag{7}$$

where h is the depth $(= H - z_b)$; z_b is the bottom elevation; and the 2-D transformed velocities are

$$\overline{U} = \frac{1}{h}\left[\xi_x \int_{z_b}^{H} u\,\mathrm{d}z + \xi_y \int_{z_b}^{H} v\,\mathrm{d}z\right] \quad \text{and} \quad \overline{V} = \frac{1}{h}\left[\eta_x \int_{z_b}^{H} u\,\mathrm{d}z + \eta_y \int_{z_b}^{H} v\,\mathrm{d}z\right] \tag{8}$$

3. NUMERICAL FORMULATION

3.1. Spatial discretization

For the viscous fluxes ($\mathbf{E_v}$, $\mathbf{F_v}$, and $\mathbf{G_v}$) and the source term ($\mathbf{S}$), a central difference approximation is used. Since numerical representation of the inviscid fluxes ($\mathbf{E}$, $\mathbf{F}$, and $\mathbf{G}$) is crucial for overall stability and accuracy of the numerical model, we use the monotone upstream-centred scheme for conservation laws (MUSCL) [13] which has been applied for incompressible flow by other authors [9, 14]. As an example, the flux in the ξ-direction can be written as

$$\frac{\partial \mathbf{E}}{\partial \xi} \approx \frac{1}{\Delta \xi}(\mathbf{E}^*_{i+1/2} - \mathbf{E}^*_{i-1/2}) \tag{9}$$

where $\Delta\xi$ is the grid increment; and i denotes a grid point in the ξ-direction. Variations in the η- and ζ-directions are represented by j and k with grid spacing $\Delta\eta$ and $\Delta\zeta$, respectively. The starred variables are called the numerical fluxes, and these can be expressed as

$$\begin{aligned} \mathbf{E}^*_{i+1/2} &= \mathbf{E}(\mathbf{Q}^{\mathrm{L}}_{i+1/2}) + \mathbf{A}^-(\overline{\mathbf{Q}}_{i+1/2})(\mathbf{Q}^{\mathrm{R}}_{i+1/2} - \mathbf{Q}^{\mathrm{L}}_{i+1/2}) \\ &= \mathbf{E}(\mathbf{Q}^{\mathrm{R}}_{i+1/2}) - \mathbf{A}^+(\overline{\mathbf{Q}}_{i+1/2})(\mathbf{Q}^{\mathrm{R}}_{i+1/2} - \mathbf{Q}^{\mathrm{L}}_{i+1/2}) \end{aligned} \tag{10}$$

where $\mathbf{Q}^{\mathrm{L}}_{i+1/2}$ and $\mathbf{Q}^{\mathrm{R}}_{i+1/2}$ are the left and right states of $\mathbf{Q}_{i+1/2}$ [15], respectively. According to the MUSCL method, they are defined as

$$\begin{aligned} \mathbf{Q}^{\mathrm{L}}_{i+1/2} &= \mathbf{Q}_i + \tfrac{1}{4}[(1-\chi)(\mathbf{Q}_i - \mathbf{Q}_{i-1}) + (1+\chi)(\mathbf{Q}_{i+1} - \mathbf{Q}_i)] \\ \mathbf{Q}^{\mathrm{R}}_{i+1/2} &= \mathbf{Q}_{i+1} - \tfrac{1}{4}[(1-\chi)(\mathbf{Q}_{i+2} - \mathbf{Q}_{i+1}) + (1+\chi)(\mathbf{Q}_{i+1} - \mathbf{Q}_i)] \end{aligned} \tag{11}$$

where χ is chosen as 1/3, giving a third-order upwind interpolation. $\overline{\mathbf{Q}}_{i+1/2}$ in (10) is determined using Roe's mean [15]; here we have replaced this term by the arithmetic mean of $\mathbf{Q}^{\mathrm{L}}_{i+1/2}$ and $\mathbf{Q}^{\mathrm{R}}_{i+1/2}$. Using a flux difference method [15], the inviscid flux Jacobian $\mathbf{A}$ $(=\partial\mathbf{E}/\partial\mathbf{Q})$ can be divided into $\mathbf{A}^+$ and $\mathbf{A}^-$, where $\mathbf{A}^{\pm}=(\mathbf{P}\mathbf{\Lambda}^{\pm}\mathbf{P}^{-1})_{\xi}$; $\mathbf{\Lambda}^{\pm}_{\xi}=(\mathbf{\Lambda}_{\xi}\pm|\mathbf{\Lambda}_{\xi}|)/2$; and $\mathbf{P}_{\xi}$ and $\mathbf{\Lambda}_{\xi}$ are matrices of eigenvalues and eigenvectors of $\mathbf{A}$, respectively. In order to construct a non-singular matrix of $\mathbf{P}_{\xi}$ and $\mathbf{P}^{-1}_{\xi}$, a method similar to that of Rogers *et al.* [16] is applied, as briefly discussed in the Appendix.

3.2. Time discretization

For a 3-D problem, it is important to be able to develop a stable method using large time increments so that large storage and CPU time requirements can be alleviated. This is true especially for turbulent calculations because, for adequate accuracy, dense grid resolution is required near walls. To achieve this, (3) is discretized using the implicit formula

$$\frac{\delta\mathbf{Q}^n}{\Delta\tau}+\delta\left[\frac{\partial}{\partial\xi}(\mathbf{E}-\mathbf{E}_{\mathbf{v}})+\frac{\partial}{\partial\eta}(\mathbf{F}-\mathbf{F}_{\mathbf{v}})+\frac{\partial}{\partial\zeta}(\mathbf{G}-\mathbf{G}_{\mathbf{v}})\right]^n$$
$$=-\left[\frac{\partial}{\partial\xi}(\mathbf{E}-\mathbf{E}_{\mathbf{v}})+\frac{\partial}{\partial\eta}(\mathbf{F}-\mathbf{F}_{\mathbf{v}})+\frac{\partial}{\partial\zeta}(\mathbf{G}-\mathbf{G}_{\mathbf{v}})-\mathbf{S}\right]^n=\mathbf{R}^n \tag{12}$$

where $\delta(\cdot)^n=(\cdot)^{n+1}-(\cdot)^n$; n is the time level; $\Delta\tau$ is the time interval; and $\mathbf{R}^n$ is the right-hand side of (12). Applying (9) with the MUSCL formula, and (11) for the inviscid fluxes on the left-hand side of (12), leads to a large banded block matrix system of equations, which can be inverted—but not easily. To simplify this, these implicit inviscid fluxes are approximated, to first order, by dropping the higher-order terms in (11) while the explicit inviscid fluxes are approximated by (9) and (11), resulting in the diagonally dominant form

$$\left[\frac{\mathbf{I}}{\Delta\tau}+\left(\frac{\mathbf{A}^+}{\Delta\xi}+\frac{\mathbf{A}_{\mathbf{v}}}{\Delta\xi^2}\right)_{i-1/2}-\left(\frac{\mathbf{A}^-}{\Delta\xi}-\frac{\mathbf{A}_{\mathbf{v}}}{\Delta\xi^2}\right)_{i+1/2}+\left(\frac{\mathbf{B}^+}{\Delta\eta}+\frac{\mathbf{B}_{\mathbf{v}}}{\Delta\eta^2}\right)_{j-1/2}\right.$$
$$\left.-\left(\frac{\mathbf{B}^-}{\Delta\eta}-\frac{\mathbf{B}_{\mathbf{v}}}{\Delta\eta^2}\right)_{j+1/2}+\left(\frac{\mathbf{C}^+}{\Delta\zeta}+\frac{\mathbf{C}_{\mathbf{v}}}{\Delta\zeta^2}\right)_{k-1/2}-\left(\frac{\mathbf{C}^-}{\Delta\zeta}-\frac{\mathbf{C}_{\mathbf{v}}}{\Delta\zeta^2}\right)_{k+1/2}\right]\delta\mathbf{Q}^n_{ijk}$$
$$-\left(\frac{\mathbf{A}^+}{\Delta\xi}+\frac{\mathbf{A}_{\mathbf{v}}}{\Delta\xi^2}\right)_{i-1/2}\delta\mathbf{Q}^n_{i-1jk}+\left(\frac{\mathbf{A}^-}{\Delta\xi}-\frac{\mathbf{A}_{\mathbf{v}}}{\Delta\xi^2}\right)_{i+1/2}\delta\mathbf{Q}^n_{i+1jk}$$
$$-\left(\frac{\mathbf{B}^+}{\Delta\eta}+\frac{\mathbf{B}_{\mathbf{v}}}{\Delta\eta^2}\right)_{j-1/2}\delta\mathbf{Q}^n_{ij-1k}+\left(\frac{\mathbf{B}^-}{\Delta\eta}-\frac{\mathbf{B}_{\mathbf{v}}}{\Delta\eta^2}\right)_{j+1/2}\delta\mathbf{Q}^n_{ij+1k}$$
$$-\left(\frac{\mathbf{C}^+}{\Delta\zeta}+\frac{\mathbf{C}_{\mathbf{v}}}{\Delta\zeta^2}\right)_{k-1/2}\delta\mathbf{Q}^n_{ijk-1}+\left(\frac{\mathbf{C}^-}{\Delta\zeta}-\frac{\mathbf{C}_{\mathbf{v}}}{\Delta\zeta^2}\right)_{k+1/2}\delta\mathbf{Q}^n_{ijk+1}=\mathbf{R}^n \tag{13}$$

where $\mathbf{B}$ and $\mathbf{C}$ are the inviscid Jacobians of $\mathbf{F}$ and $\mathbf{G}$, respectively. In order to simplify the coefficient matrix, the source term is retained only in the explicit formulation and the viscous

Jacobians are approximated by dropping the off-diagonal elements, leading to

$$\mathbf{A}_{\mathbf{v}} = \frac{\nabla\xi\cdot\nabla\xi}{J}\frac{\nu_{\mathrm{T}}}{c_1}\mathbf{I}_{\mathbf{m}}, \quad \mathbf{B}_{\mathbf{v}} = \frac{\nabla\eta\cdot\nabla\eta}{J}\frac{\nu_{\mathrm{T}}}{c_1}\mathbf{I}_{\mathbf{m}}, \quad \mathbf{C}_{\mathbf{v}} = \frac{\nabla\zeta\cdot\nabla\zeta}{J}\frac{\nu_{\mathrm{T}}}{c_1}\mathbf{I}_{\mathbf{m}} \tag{14}$$

In this numerical scheme, it is assumed further that

$$\mathbf{A}^{+}_{i-1/2} - \mathbf{A}^{-}_{i+1/2} \approx \mathbf{A}^{+}_{ijk} - \mathbf{A}^{-}_{ijk} \tag{15}$$

and $\mathbf{A}^{\pm} = (\mathbf{A} \pm r_{\mathbf{A}}\mathbf{I})/2$ [17], where $r_{\mathbf{A}} = \max(|\lambda_{\mathbf{A}}|)$ is the spectral radius of $\mathbf{A}$. With similar approximations for $\mathbf{B}$ and $\mathbf{C}$, substituting into (13) yields the lower–upper symmetric Gauss–Seidel (LU-SGS) method [17]

$$(\mathbf{L} + \mathbf{D} + \mathbf{U})\delta\mathbf{Q}^n = \mathbf{R}^n \tag{16}$$

where

$$\begin{aligned}
\mathbf{D} &= \left[\frac{1}{\Delta\tau} + \frac{r_{\mathbf{A}}}{\Delta\xi} + \frac{r_{\mathbf{B}}}{\Delta\eta} + \frac{r_{\mathbf{C}}}{\Delta\zeta}\right]\mathbf{I} + \left(\frac{\mathbf{A}_{\mathbf{v}i-1/2}}{\Delta\xi^2} + \frac{\mathbf{A}_{\mathbf{v}i+1/2}}{\Delta\xi^2}\right) \\
&\quad + \left(\frac{\mathbf{B}_{\mathbf{v}j-1/2}}{\Delta\eta^2} + \frac{\mathbf{B}_{\mathbf{v}j+1/2}}{\Delta\eta^2}\right) + \left(\frac{\mathbf{C}_{\mathbf{v}k-1/2}}{\Delta\zeta^2} + \frac{\mathbf{C}_{\mathbf{v}k+1/2}}{\Delta\zeta^2}\right) \\
\mathbf{L} &= -\left(\frac{\mathbf{A}^{+}}{\Delta\xi} + \frac{\mathbf{A}_{\mathbf{v}}}{\Delta\xi^2}\right)_{i-1/2} - \left(\frac{\mathbf{B}^{+}}{\Delta\eta} + \frac{\mathbf{B}_{\mathbf{v}}}{\Delta\eta^2}\right)_{j-1/2} - \left(\frac{\mathbf{C}^{+}}{\Delta\zeta} + \frac{\mathbf{C}_{\mathbf{v}}}{\Delta\zeta^2}\right)_{k-1/2} \\
\mathbf{U} &= \left(\frac{\mathbf{A}^{-}}{\Delta\xi} + \frac{\mathbf{A}_{\mathbf{v}}}{\Delta\xi^2}\right)_{i+1/2} + \left(\frac{\mathbf{B}^{-}}{\Delta\eta} + \frac{\mathbf{B}_{\mathbf{v}}}{\Delta\eta^2}\right)_{j+1/2} + \left(\frac{\mathbf{C}^{-}}{\Delta\zeta} + \frac{\mathbf{C}_{\mathbf{v}}}{\Delta\zeta^2}\right)_{k+1/2}
\end{aligned} \tag{17}$$

The resulting LU-SGS formula is solved easily using two sweeps: one forward (18) and another backward (19)

$$(\mathbf{L} + \mathbf{D})\delta\mathbf{Q}^* = \mathbf{R}^n \implies \delta\mathbf{Q}^* = \mathbf{D}^{-1}(\mathbf{R}^n - \mathbf{L}\delta\mathbf{Q}^*) \tag{18}$$

$$(\mathbf{D} + \mathbf{U})\delta\mathbf{Q}^n = \mathbf{D}\delta\mathbf{Q}^* \implies \delta\mathbf{Q}^n = \delta\mathbf{Q}^* - \mathbf{D}^{-1}(\mathbf{U}\delta\mathbf{Q}^n) \tag{19}$$

It should be noted that matrix inversion in (18) and (19) is replaced by scalar division. The LU-SGS method has an advantage over factorized formulae such as the alternating direction implicit (ADI) method [18] in that factorization errors can be eliminated.

After obtaining $\mathbf{Q}^{n+1}$ $(= \delta\mathbf{Q}^n + \mathbf{Q}^n)$, the water surface elevation is determined by solving (7) implicitly using a method similar to that used for (3).

3.3. Boundary conditions

Near wall and bottom boundaries, all flow variables vary rapidly. In order to reduce intensive computation by imposing a large number of grid elements, a wall function is applied. Using this method, boundary conditions are specified at the first grid point $\overline{z_1}$ from the wall rather than at the boundary itself. The velocity normal to each wall is set to zero while the two tangential velocities are determined using a method similar to Stansby [3], which is based on

the standard formulae for open channel flows [19]:

$$\frac{u_1}{u_*} = \begin{cases} \dfrac{1}{\kappa} \ln\left(\dfrac{30\bar{z}_1}{k_s}\right) & \text{for rough walls} \\ \dfrac{1}{\kappa} \ln\left(\dfrac{9.05 u_* \bar{z}_1}{\nu}\right) & \text{for smooth walls} \end{cases} \tag{20}$$

where u_1 is the velocity parallel to the boundary—which is calculated from the momentum equations at $\bar{z}_1$, the normal distance of the first grid point from the boundary; the shear velocity u_* is related to the bottom shear stress τ_w as $u_* = \sqrt{\tau_w/\rho}$; $\kappa = 0.41$ is the von Karman constant; and k_s is the roughness height. Equation (20) is only applicable in the log-law region where $30 < u_* \bar{z}_1/\nu < 100$ [12]. Using this shear velocity, boundary conditions for k and ε at the wall [12] are

$$k = \frac{u_*^2}{\sqrt{c_\mu}} \quad \text{and} \quad \varepsilon = \frac{|u_*^3|}{\kappa \bar{z}_1} \tag{21}$$

The hydrodynamic pressure at each wall is obtained using extrapolation of two interior values.

At the free surface, $p_d = 0$ and a zero normal gradient condition for u and v is applied. The vertical velocity w is determined from the kinematic boundary condition,

$$\left[w = \frac{\partial H}{\partial t} + u \frac{\partial H}{\partial x} + v \frac{\partial H}{\partial y} \right]_{z=H} \tag{22}$$

Without wind stress, k and ε are obtained [12] from

$$\frac{\partial k}{\partial z} = 0 \quad \text{and} \quad \varepsilon = \frac{\left(k\sqrt{c_\mu}\right)^{1.5}}{0.07\kappa h} \tag{23}$$

At the inflow boundary, all dependent variables are specified using experimental data or assuming a uniform distribution, while at the outlet the normal gradients of these variables are set to zero. More details of inflow boundary conditions are described for each application.

3.4. Grid generation

In this paper, horizontal grids are distributed equally depending on the geometrical boundaries. In the vertical direction, a grid clustering method similar to Hoffmann and Chiang [20] is used to allow the concentration of grid points near the bottom and the surface boundary:

$$\zeta = \alpha_1 + (1 - \alpha_1) \frac{\ln[\{\alpha_2 + (2\alpha_1 + 1)(z - z_b)/h - 2\alpha_1\}/\{\alpha_2 - (2\alpha_1 + 1)(z - z_b)/h + 2\alpha_1\}]}{\ln[(\alpha_2 + 1)/(\alpha_2 - 1)]} \tag{24}$$

where α_1 and α_2 control clustering location and density, respectively.

The inverse of (24) is given by

$$z = z_b + h \frac{(2\alpha_1 + \alpha_2)\{(\alpha_2 + 1)/(\alpha_2 - 1)\}^{(\zeta - \alpha_1)/(1 - \alpha_1)} + 2\alpha_1 - \alpha_2}{(2\alpha_1 + 1)[1 + \{(\alpha_2 + 1)/(\alpha_2 - 1)\}^{(\zeta - \alpha_1)/(1 - \alpha_1)}]} \tag{25}$$

4. APPLICATION

In order to verify the accuracy and applicability of the current method, two test cases are considered, the first involving open channel flow with a dredged trench in the middle [21], and the second a meandering channel with a rectangular cross section [22].

4.1. Case 1: Trench channel flow

The trench channel flow problem was originally examined experimentally in a channel 17 m long and 0.5 m wide with 0.7 m high side walls [21]. A number of authors have examined this computationally: Alfrink and van Rijn [23] solve the problem using a rigid lid approximation; and solutions allowing a free surface have been obtained using a standard k–ε model [24] and simplified Reynolds stress models [25].

To reduce computational requirements, we have solved this problem by considering only 2.8 m in the horizontal direction around the trench (see Figure 1). Initially an equally spaced grid of 0.05 m is used in the horizontal and lateral directions with 30 layers in the vertical direction using $\alpha_1 = 0.5$ and $\alpha_2 = 1.1$ in (25). Figure 1 shows a cross section of the grid along the $y = 0.25$ m centre plane. The five locations identified in Figure 1 show where measured data are available [21]. To assess the sensitivity of the solution to variations in grid size, the initial grid arrangement has been refined using grid spacings of 0.025 m and 0.0125 m horizontally with 40 and 50 vertical layers. For the time increment we set the Courant–Friedrichs–Lewy (CFL) number to 30, which is approximately 0.05 s for the initial grid set-up; this is 20 times larger than that used by Stansby and Zhou [24], even though the grid step in the longitudinal ξ-direction is half of what they used. The simulation has been run for 200 s, and we consider steady state solutions when $|\delta p_d|_2 \approx 10^{-3}$ where $|\delta p_d|_2$ is the Euclidian norm of $\delta p_d = p_d^{n+1} - p_d^n$. Figure 2 shows the convergence history of $|\delta p_d|_2$ and $|\delta u|_2$ using only 0.05 m grid spacing. Convergence for the finer grids is slightly slower.

At the left-hand (inflow) boundary, the u velocity is defined using the rough wall condition in (20) and k and ε are specified using (21) with $u_* = 0.03$ m/s [24], while the other velocities are set to zero. At the right-hand (outflow) boundary, the radiation boundary conditions are applied. Along the side walls, u, w, k, and ε satisfy the zero flux condition across the wall, while v is zero.

A range of values of β from 0.5 to 10 was used to evaluate convergence rates of the numerical model. Similar to other authors [7, 8], a value of $\beta = 1$ produced the most acceptable convergence.

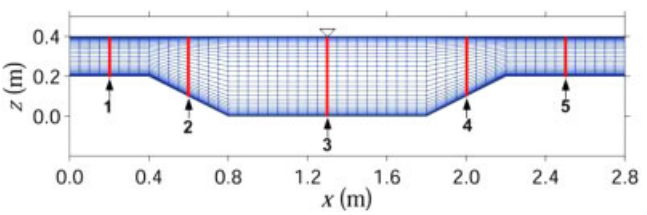

Figure 1. Cross section of trench channel geometry, with measurement sections (at 1–5) and grid distribution, for a vertical plane at $y = 0.25$ m.

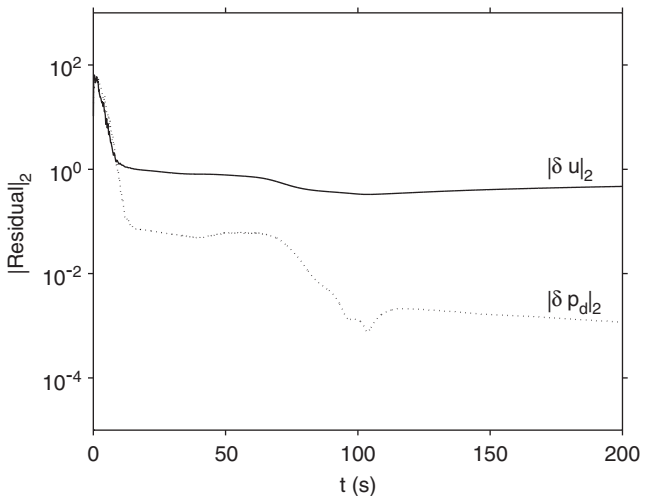

Figure 2. Convergence history of the Euclidian norms for the hydrodynamic pressure $|\delta p_d|_2$ (dashed line) and horizontal velocity $|\delta u|_2$ (solid line) using 0.05 m grid spacing.

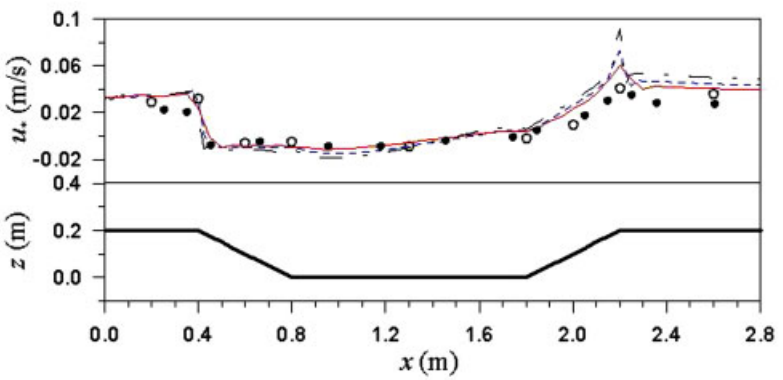

Figure 3. Shear velocities u_* from three different grids (solid line: 0.05 m grid space, dashed line: 0.025 m grid space, dash–dotted line: 0.0125 m grid space), experimental data (hollow circles) [21], and published results (filled circles) [24], along the centre line at $y = 0.25$ m.

In Figure 3, we compare the shear velocities calculated using (20) and the three different grid meshes along the centre line of $y = 0.25$ m with experimental data [22] and published results [24]. Even with the largest grid space (0.05 m), computed shear velocities agree reasonably well with the experimental data. Decreasing the grid spacing produces steeper changes at the beginning and end of the trench. In particular, results using the 0.0125 m grid spacing near $x = 2.2$ m show a sharp increase in the shear velocities. This is consistent with [25] and is probably highlighted by the small grid spacing.

In Figure 4, the computed horizontal velocity (u), TKE (k), and shear stress (τ_{xz}) for the 0.05 and 0.025 m grids are compared with experimental data [21], indicating that overall

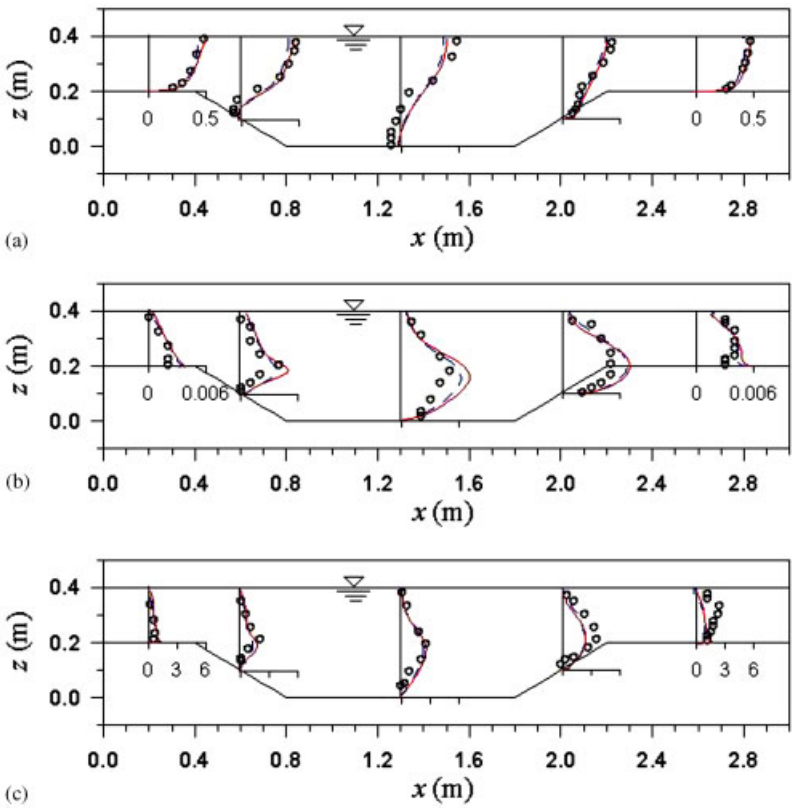

Figure 4. Comparison of numerical solutions from two different grids (solid line: 0.05 m grid space, dashed line: 0.025 m grid space) with experimental data (circles) [21], for the trench channel flow problem along the centre line at $y = 0.25$ m: (a) u (m/s); (b) k (m^2/s^2); and (c) τ_{xz} (N/m^2).

agreement is achieved; in addition, the model predicts circulation at locations 2 and 3 ($x = 0.6$ and 1.3 m, respectively), leading to negative velocities near the bottom. At these locations, the results show that, as the grid is refined, gradients of the horizontal velocities increase slightly. The largest discrepancy is observed at location 3. This may indicate the limitation of a standard k–ε model, in that only isotropic turbulence can be resolved. It is important to note that the solutions from this model are as good as those of Stansby and Zhou [24] even though much larger time steps have been used in our simulations. This indirectly indicates the accuracy and applicability of the current ACM.

4.2. Case 2: Meandering channel flow

The meandering channel flow problem has been examined experimentally by Chang [22], and numerically using a rigid lid approximation [26] and a free surface condition [4]. We have simulated this problem using the computational grid shown in Figure 5, where a variable grid has been used in the longitudinal direction, a constant grid in the transverse direction, and

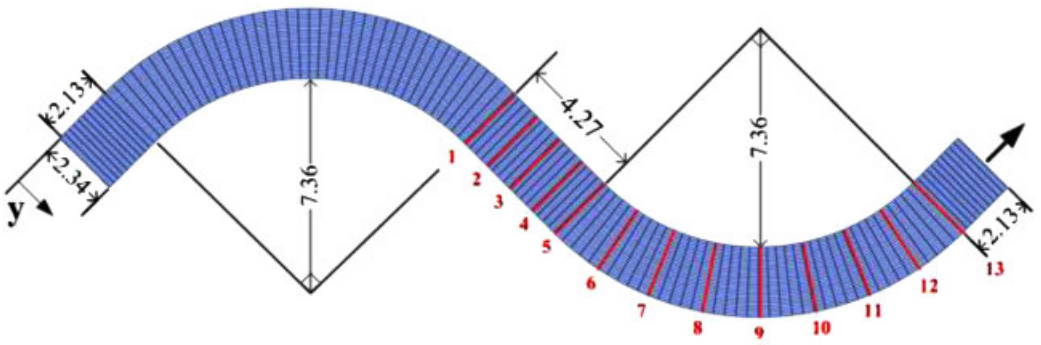

Figure 5. Meandering channel geometry, with measurement sections (1–13) [22] and grid distribution for the horizontal plane (lengths are shown in m).

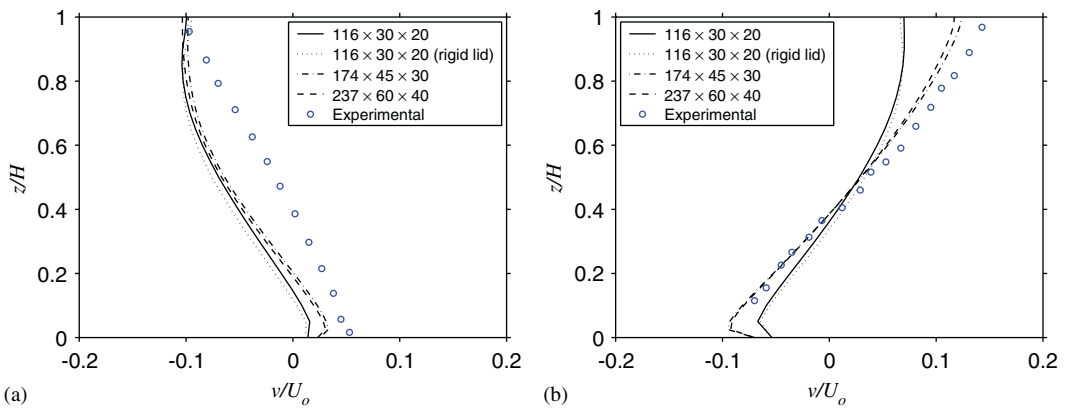

Figure 6. Transverse velocity v/U_o distributions over depth at Sections 1 and 9, with three different grid refinements and the rigid lid approximation (for only the $116 \times 30 \times 20$ grid): (a) $y/B = 0.83$ at section 1; and (b) $y/B = 0.17$ at section 9.

a variable grid—using (25) with $\alpha_1 = 0.5$ and $\alpha_2 = 1.2$—in the vertical direction; this gives a total of $116 \times 30 \times 20$ nodes. Figure 5 also shows the locations of the measurement sections (1–13) presented in Chang [22]. The inflow boundary conditions used are: $U = U_o = 0.355\,\text{m/s}$ [26]; and $V = W = 0\,\text{m/s}$. Similar to the previous test case, the radiation boundary conditions are applied at the outlet. The smooth wall boundary condition (20) is applied along the bottom and side walls.

In order to test the sensitivity of the solution to variation in grid size, the initial grid spacing (in all directions) has been increased by factors of 1.5 and 2.0. In all cases, the time increment has been kept at 0.2 s, and steady state solutions are considered after 500 s when $|\delta p_d|_2 \approx 10^{-3}$.

Figure 6 shows the vertical distribution of the normalized transverse velocity v/U_o at Section 9 (Figure 5) with the three different grids, where $B = 2.34\,\text{m}$ is the width of the

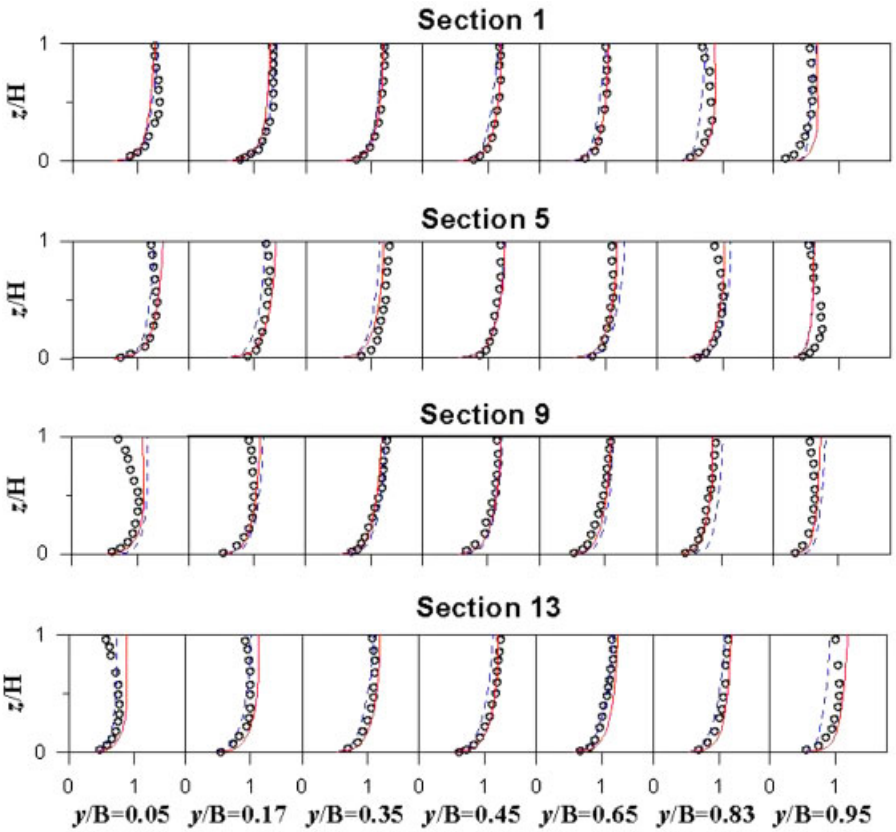

Figure 7. Comparison of longitudinal velocities u/U_o between computed values (solid line: $174 \times 45 \times 30$ grid, dashed line: $116 \times 30 \times 20$ grid) and experimental data (circles) [22] at selected sections (1, 5, 9, and 13) of the meandering channel.

channel. As the grids are refined, the vertical gradients tend to increase, although even doubling the initial grid refinement (i.e. halving the grid spacings) has little effect on the solution. Because of this, the $116 \times 30 \times 20$ and $174 \times 45 \times 30$ grids have been used to develop all subsequent results. An interesting aspect shown in Figure 6 is the different vertical flow regimes at similar points from each of the lateral boundaries, a result of the meandering channel. In this figure, numerical solutions using the rigid lid approximation (not allowing free surface changes) are included for the $116 \times 30 \times 20$ grid, and compared with solutions where free surface changes are allowed. The small changes between these results indicate that free surfaces changes are minimal for this problem.

In Figures 7 and 8, computed normalized velocities u/U_o and v/U_o are compared with experimental data [22] at Sections 1, 5, 9, and 13, showing strong secondary currents (lateral velocities that vary in direction with depth), except at Section 5, which is located at a straight section of the channel. In Figure 7, longitudinal velocities from the dense ($174 \times 45 \times 30$) grid seem to be slightly overestimated while solutions from the coarse ($116 \times 30 \times 20$) grid

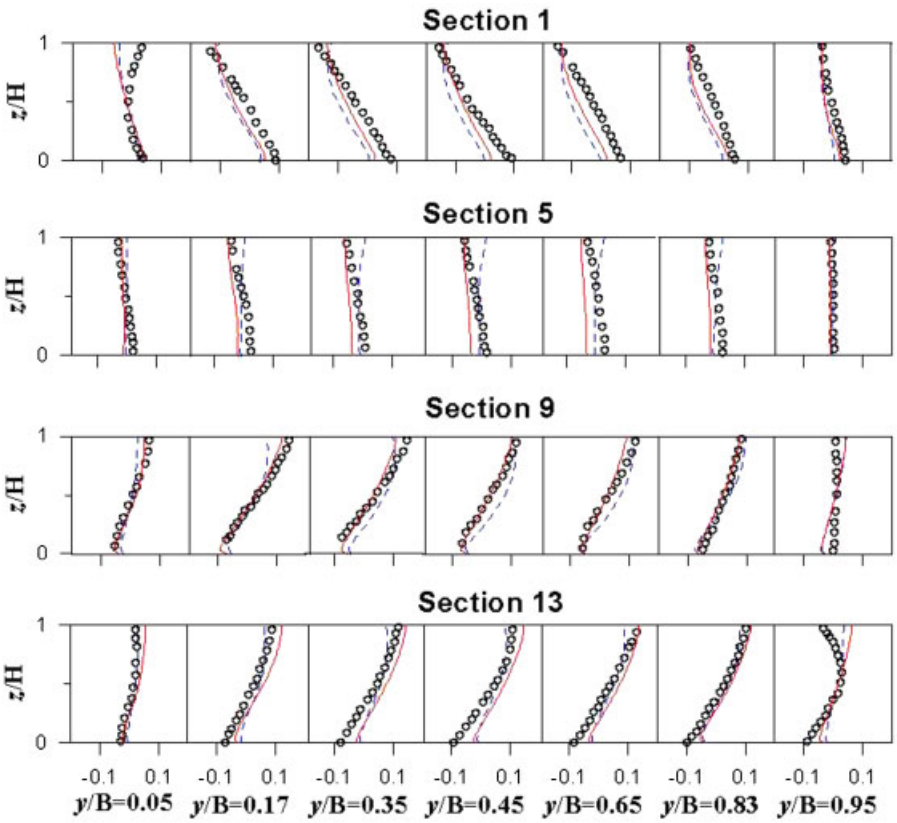

Figure 8. Comparison of transverse velocities v/U_o between computed values (solid line: $174 \times 45 \times 30$ grid, dashed line: $116 \times 30 \times 20$ grid) and experimental data (circles) [22] at selected sections (1, 5, 9, and 13) of the meandering channel.

are underestimated. As shown in Figure 8, computed velocities agree quite well with the experimental data for the dense grid. However, the method discussed in this paper does not accurately predict the small-scale circulation shown in the first ($y/B = 0.05$) image at Section 1 and the last ($y/B = 0.95$) image at Section 13. As noted by other researchers [4, 26], this deficiency may be rectified by using a higher-order turbulence model. Ye and McCorquodale [4] also mentioned difficulties that can arise in obtaining accurate measurement data in such situations, as secondary circulation velocities tend to be one or two orders of magnitude smaller than the longitudinal velocities.

5. CONCLUSION

In this paper, we have presented a numerical method for modelling turbulent open channel flows which has been used to solve for hydrodynamic pressure and velocities simultaneously

by using an artificial compressibility method. For turbulence closure, the standard k–ε model is employed. In this numerical method, the hyperbolic-like advection terms are approximated using Roe's flux difference method with a third-order MUSCL method. The model has been applied to two steady state test cases where data are available: the trench and meandering channel flow problems. In both cases, the model has been shown to be accurate using much larger time steps than used by previous researchers [24]. The second problem identified the need to consider higher-order turbulence models to attempt to better resolve small-scale motions.

APPENDIX A

The inviscid Jacobian matrix is given by

$$\mathbf{A}_i = \begin{pmatrix} a_\tau & a_x\beta & a_y\beta & a_z\beta & 0 & 0 \\ a_x & a_x u + \Theta + a_\tau & a_y u & a_z u & 0 & 0 \\ a_y & a_x v & a_y v + \Theta + a_\tau & a_z v & 0 & 0 \\ a_z & a_x w & a_y w & a_z w + \Theta + a_\tau & 0 & 0 \\ 0 & a_x k & a_y k & a_z k & \Theta + a_\tau & 0 \\ 0 & a_x \varepsilon & a_y \varepsilon & a_z \varepsilon & 0 & \Theta + a_\tau \end{pmatrix} \tag{A1}$$

where $\mathbf{A}_i = \mathbf{A}, \mathbf{B}, \mathbf{C}$ for $i = 1, 2, 3$, respectively;

$$\Theta = a_x u + a_y v + a_z w \tag{A2}$$

and

$$a_\tau = \frac{\partial \xi_i}{\partial \tau}, \quad a_x = \frac{\partial \xi_i}{\partial x}, \quad a_y = \frac{\partial \xi_i}{\partial y}, \quad a_z = \frac{\partial \xi_i}{\partial z}, \quad \text{with } \xi_i = \xi, \eta, \zeta \text{ for } i = 1, 2, 3 \tag{A3}$$

The Jacobian matrix can be diagonalized as

$$\mathbf{A}_i = \mathbf{P}_i \mathbf{\Lambda}_i \mathbf{P}_i^{-1} \tag{A4}$$

where $\mathbf{\Lambda}_i = \mathrm{diag}(\Theta + a_\tau - c, \Theta + a_\tau, \Theta + a_\tau, \Theta + a_\tau, \Theta + a_\tau, \Theta + a_\tau + c)$ is a diagonal matrix with eigenvalues of $\mathbf{A}_i$; and $c = \sqrt{\Theta^2 + \beta b}$, with $b = a_x^2 + a_y^2 + a_z^2$.

We found the matrix of eigenvectors to be

$$\mathbf{P}_i = \begin{pmatrix} (\Theta + c)c & 0 & 0 & 0 & 0 & -(\Theta - c)c \\ ub - (\Theta + c)a_x & x_{\xi_{i+1}} & 0 & 0 & x_{\xi_{i+2}} & ub - (\Theta - c)a_x \\ vb - (\Theta + c)a_y & y_{\xi_{i+1}} & 0 & 0 & y_{\xi_{i+2}} & vb - (\Theta - c)a_y \\ wb - (\Theta + c)a_z & z_{\xi_{i+1}} & 0 & 0 & z_{\xi_{i+2}} & wb - (\Theta - c)a_z \\ kb & 0 & 1 & 0 & 0 & kb \\ \varepsilon b & 0 & 0 & 1 & 0 & \varepsilon b \end{pmatrix} \tag{A5}$$

The inverse of this matrix is given by

$$\mathbf{P}_i^{-1} = \frac{1}{|\mathbf{P}|}(\mathbf{p}_1^{-1}, \mathbf{p}_2^{-1}, \mathbf{p}_3^{-1}, \mathbf{p}_4^{-1}, \mathbf{p}_5^{-1}, \mathbf{p}_6^{-1}) \tag{A6}$$

where

$$|\mathbf{P}| = 2c(\Theta^2 - c^2)(-a_x d_1 + a_y d_2 - a_z d_3) + 2c\Theta b(ud_1 - vd_2 + wd_3) \tag{A7}$$

$$d_1 = y_{\xi_{i+1}} z_{\xi_{i+2}} - y_{\xi_{i+2}} z_{\xi_{i+1}}; \quad d_2 = x_{\xi_{i+1}} z_{\xi_{i+2}} - x_{\xi_{i+2}} z_{\xi_{i+1}}; \quad d_3 = x_{\xi_{i+1}} y_{\xi_{i+2}} - x_{\xi_{i+2}} y_{\xi_{i+1}} \tag{A8}$$

$$\mathbf{p}_1^{-1} = \begin{pmatrix} -d_1[a_x(\Theta - c) - ub] + d_2[a_y(\Theta - c) - vb] - d_3[a_z(\Theta - c) - wb] \\ -2cb[-a_x(y_{\xi_{i+2}} w - z_{\xi_{i+2} v}) + a_y(x_{\xi_{i+2}} w - z_{\xi_{i+2} u}) - a_z(x_{\xi_{i+2}} v - y_{\xi_{i+2} u}) \\ -2cbk[a_x d_1 - a_y d_2 + a_z d_3] \\ -2cb\varepsilon[a_x d_1 - a_y d_2 + a_z d_3] \\ -2cb[-a_x(y_{\xi_{i+1}} w - z_{\xi_{i+1} v}) + a_y(x_{\xi_{i+1}} w - z_{\xi_{i+1} u}) - a_z(x_{\xi_{i+1}} v - y_{\xi_{i+1} u}) \\ d_1[a_x(\Theta + c) - ub] - d_2[a_y(\Theta + c) - vb] + d_3[a_z(\Theta + c) - wb] \end{pmatrix} \tag{A9}$$

$$\mathbf{p}_2^{-1} = \begin{pmatrix} c(\Theta - c)d_1 \\ 2c[(\Theta^2 - c^2)(a_y z_{\xi_{i+2}} - a_z y_{\xi_{i+2}}) + \Theta b(y_{\xi_{i+2}} w - z_{\xi_{i+2}} v)] \\ -2\Theta cbkd_1 \\ -2\Theta cb\varepsilon d_1 \\ -2c[(\Theta^2 - c^2)(a_y z_{\xi_{i+1}} - a_z y_{\xi_{i+1}}) + \Theta b(y_{\xi_{i+1}} w - z_{\xi_{i+1}} v)] \\ c(\Theta + c)d_1 \end{pmatrix} \tag{A10}$$

$$\mathbf{p}_3^{-1} = \begin{pmatrix} -c(\Theta - c)d_2 \\ -2c[(\Theta^2 - c^2)(a_x z_{\xi_{i+2}} - a_z x_{\xi_{i+2}}) + \Theta b(x_{\xi_{i+2}} w - z_{\xi_{i+2}} u)] \\ 2\Theta cbkd_2 \\ 2\Theta cb\varepsilon d_2 \\ 2c[(\Theta^2 - c^2)(a_x z_{\xi_{i+1}} - a_z x_{\xi_{i+1}}) + \Theta b(x_{\xi_{i+1}} w - z_{\xi_{i+1}} u)] \\ -c(\Theta + c)d_2 \end{pmatrix} \tag{A11}$$

$$\mathbf{p}_4^{-1} = \begin{pmatrix} c(\Theta - c)d_3 \\ 2c[(\Theta^2 - c^2)(a_x y_{\xi_{i+2}} - a_y x_{\xi_{i+2}}) + \Theta b(x_{\xi_{i+2}} v - y_{\xi_{i+2}} y)] \\ -2\Theta cbkd_3 \\ -2\Theta cb\varepsilon d_3 \\ -2c[(\Theta^2 - c^2)(a_x y_{\xi_{i+1}} - a_y x_{\xi_{i+1}}) + \Theta b(x_{\xi_{i+1}} v - y_{\xi_{i+1}} y)] \\ c(\Theta + c)d_3 \end{pmatrix} \tag{A12}$$

$$\mathbf{p}_5^{-1} = \begin{pmatrix} 0 \\ 0 \\ |\mathbf{P}| \\ 0 \\ 0 \\ 0 \end{pmatrix}; \quad \text{and} \quad \mathbf{p}_6^{-1} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ |\mathbf{P}| \\ 0 \\ 0 \end{pmatrix} \tag{A13}$$

From (A7) and (A8), d_1, d_2, and d_3 are non-zero real numbers, which ensure the existence of $|\mathbf{P}|$. When the subindex ξ_i in x_{ξ_i}, y_{ξ_i}, and z_{ξ_i} is greater than 3, the cyclic notation is applied, that is $\xi_4 = \xi_1 = \xi$, etc.

ACKNOWLEDGEMENTS

Jong Wook Lee has been supported by an international postgraduate research scholarship (IPRS) from the University of Adelaide and a supplementary scholarship from United Water International Pty Ltd., Adelaide.

REFERENCES

1. Casulli V. A semi-implicit finite difference method for non-hydrostatic, free-surface flows. *International Journal for Numerical Methods in Fluids* 1999; **30**:425–440.
2. Lee JW, Teubner MD, Nixon JB, Gill PM. A 3-D non-hydrostatic pressure model for free surface flows. *International Journal for Numerical Methods in Fluids* 2004, in press.
3. Stansby PK. Semi-implicit finite volume shallow-water flow and solute transport solver with k–ε turbulence model. *International Journal for Numerical Methods in Fluids* 1997; **25**:285–313.
4. Ye J, McCorquodale JA. Simulation of curved open channel flows by 3D hydrodynamic model. *Journal of Hydrologic Engineering* 1998; **124**:687–698.
5. Ferziger JH, Perič M. *Computational Methods for Fluid Dynamics* (3rd edn). Springer: Berlin, 2002.
6. Chorin AJ. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics* 1967; **2**:12–26.
7. Chen Y-N, Yang S-C, Yang J-Y. Implicit weighted essentially non-oscillatory schemes for the incompressible Navier–Stokes equations. *International Journal for Numerical Methods in Fluids* 1999; **31**:747–765.
8. Rogers SE, Kwak D, Kiris C. Steady and unsteady solutions of the incompressible Navier–Stokes equations. *AIAA Journal* 1991; **29**:603–610.
9. Yuan L. Comparison of implicit multigrid schemes for three-dimensional incompressible flows. *Journal of Computational Physics* 2002; **177**:134–155.

10. Beddhu M, Taylor LK, Whitfield DL. A time accurate calculation procedure for flow with a free surface using a modified artificial compressibility formulation. *Applied Mathematics and Computation* 1994; **65**:33–48.
11. Li T. Computation of turbulent free-surface flows around modern ships. *International Journal for Numerical Methods in Fluids* 2003; **43**:407–430.
12. Rodi W. *Turbulence Models and their Application in Hydraulics—A State of the Art Review.* IAHR, 1984.
13. van Leer B. Towards the ultimate conservative difference scheme, V. A second order sequel to Godunov's method. *Journal of Computational Physics* 1979; **32**:101–136.
14. Dubuc L, Cantariti F, Woodgate M, Gibben B, Badcock KJ, Richards BE. Solution of the unsteady Euler equations using an implicit dual-time method. *AIAA Journal* 1998; **36**:1417–1424.
15. Roe PL. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics* 1981; **43**:357–379.
16. Rogers SE, Chang JLC, Kwak D. A diagonal algorithm for the method of pesudocompressibility. *Journal of Computational Physics* 1987; **73**:364–379.
17. Yoon S, Jameson A. Lower-upper symmetric-Gauss–Seidel method for the Euler and Navier–Stokes equations. *AIAA Journal* 1988; **26**:1025–1026.
18. Douglas J Jr, Rachford HH Jr. On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American Mathematical Society* 1956; **82**:421–439.
19. Chow VT. *Open-channel Hydraulics*. McGraw-Hill: New York, 1959.
20. Hoffmann KA, Chiang ST. *Computational Fluid Dynamics for Engineers*, vol. II. Engineering Education System, 1993.
21. van Rijn LC. *The Computation of the Flow and Turbulence Field in Dredged Trenches*. Delft Hydraulics Laboratory, 1982.
22. Chang YC. Lateral mixing in meandering channels. *Ph.D. Thesis*, University of Iowa, Iowa.
23. Alfrink BJ, van Rijn LC. Two-equation turbulence model for flow in trenches. *Journal of Hydrologic Engineering* 1983; **109**:941–958.
24. Stansby PK, Zhou JG. Shallow-water flow solver with non-hydrostatic pressure: 2D vertical plane problems. *International Journal for Numerical Methods in Fluids* 1998; **28**:541–563.
25. Basara B, Younis BA. Prediction of turbulent flows in dredged trenches. *Journal of Hydraulic Research* 1995; **33**:813–824.
26. Demuren AO, Rodi W. Calculation of flow and pollutant dispersion in meandering channels. *Journal of Fluid Mechanics* 1986; **172**:63–92.

