

## A hybrid scheme based on finite element/volume methods for two immiscible fluid flows

Tian Wan<sup>1</sup>, Shahrouz Aliabadi<sup>1,\*</sup>,<sup>†</sup> and Christopher Bigler<sup>2</sup>

<sup>1</sup>*School of Engineering, Northrop Grumman Center for High Performance Computing, Jackson State University, Jackson, MS, U.S.A.*

<sup>2</sup>*Department of Naval Architecture and Marine Engineering, University of Michigan, Ann Arbor, MI, U.S.A.*

### SUMMARY

We have successfully extended our implicit hybrid finite element/volume (FE/FV) solver to flows involving two immiscible fluids. The solver is based on the segregated pressure correction or projection method on staggered unstructured hybrid meshes. An intermediate velocity field is first obtained by solving the momentum equations with the matrix-free implicit cell-centered FV method. The pressure Poisson equation is solved by the node-based Galerkin FE method for an auxiliary variable. The auxiliary variable is used to update the velocity field and the pressure field. The pressure field is carefully updated by taking into account the velocity divergence field. This updating strategy can be rigorously proven to be able to eliminate the unphysical pressure boundary layer and is crucial for the correct temporal convergence rate. Our current staggered-mesh scheme is distinct from other conventional ones in that we store the velocity components at cell centers and the auxiliary variable at vertices. The fluid interface is captured by solving an advection equation for the volume fraction of one of the fluids. The same matrix-free FV method, as the one used for momentum equations, is used to solve the advection equation. We will focus on the interface sharpening strategy to minimize the smearing of the interface over time. We have developed and implemented a global mass conservation algorithm that enforces the conservation of the mass for each fluid. Copyright © 2009 John Wiley & Sons, Ltd.

Received 26 August 2008; Revised 18 November 2008; Accepted 3 December 2008

**KEY WORDS:** finite element method; finite volume method; two-fluid flows; freesurface flows; incompressible flows; hybrid method

---

\*Correspondence to: Shahrouz Aliabadi, School of Engineering, Northrop Grumman Center for High Performance Computing, Jackson State University, Jackson, MS, U.S.A.

<sup>†</sup>E-mail: saliabadi@jsums.edu

Contract/grant sponsor: Northrop Grumman Ship Systems and National Science Foundation

## 1. INTRODUCTION

Recently, we developed a hybrid finite element/volume (FE/FV) solver [1] for incompressible flows. The hybrid solver is based on the well-known pressure correction (projection) method [2, 3]. The solution procedure follows a segregated approach to decouple the pressure from the velocity. The velocity field is updated by solving the momentum equation provided that a known pressure field is given as a source term, through a cell-centered FV discretization. The pressure does not directly enter the momentum equation. Instead, an auxiliary variable, which is closely related to the pressure, takes the place of pressure in the momentum equation, providing pressure gradient information. We put the auxiliary variable on the vertices of cells. This deployment provides a convenient way to evaluate the pressure gradient using the local FE basis functions. The incremental value of the auxiliary variable is computed by solving a Poisson equation using the Galerkin FE method. The auxiliary variable is then used to update the velocity field. After the final velocity field is determined, the pressure can be updated using the auxiliary variable and the velocity divergence field. The pressure is updated in such a way that the pressure field is free of unphysical conditions in the boundary layer.

Our hybrid FE/FV solver aims to take advantage of the merits of both the FV and the FE methods and avoid their shortcomings. For example, highly stretched cells (also known as high-aspect-ratio cells) are commonly used inside the boundary layer for high Reynolds number flows to resolve the boundary layer and reduce the number of cells. The stabilization parameters in the stabilized FE-based flow solvers [4, 5] are related to the characteristic element length that is not well defined for high-aspect-ratio mesh elements. Owing to this, it is very difficult to control the numerical dissipation of stabilized FE solvers. By contrast, the FV flow solver is very insensitive to the aspect ratio of the mesh cells. It is quite common for the FV solvers to handle cells with aspect ratios in the order of thousands [6, 7]. For this reason, we use the FV method to solve the momentum equation. On the other hand, the classic Galerkin FE method is very suitable for the elliptic-type equations like the pressure Poisson equation emerging from the segregated approach. Therefore, the combination of the FV method and the FE method is expected to perform well in the incompressible flow solvers based on the pressure projection method, which has been confirmed by our earlier work [1]. The numerical examples we presented in [1] are all about low Reynolds number flows. In our recent work, we have extended the hybrid flow solver to high Reynolds number flows using hybrid meshes with high-aspect-ratios [8].

We also incorporated the detached eddy simulation (DES) [9] turbulence model into the flow solver to compute the eddy viscosity. The DES model was originally proposed to be an affordable hybrid Reynolds-averaged Navier–Stokes (RANS) and large eddy simulation (LES) model for flows at realistic Reynolds numbers. In attached boundary layers, the DES model acts as a RANS model and in massively separated regions the DES functions as a LES model. A modified distance to the wall named the DES distance acts as a switch between the RANS mode and the LES mode [9]. There are two DES-based turbulence models. One is the one-equation Spalart–Allmaras (SA) DES model [9, 10]. The other is the two-equation shear stress transport DES model [11]. The SA–DES model is gaining more popularity [12–14] due to its simplicity and fair accuracy. In our hybrid flow solver, we incorporated the SA–DES turbulence model. For details of implementation, see [8].

Generally, there are two distinct approaches in the numerical simulation of two-fluid flows (excluding panel methods [15]). Depending on the physical characteristics of the problem, either ‘moving-mesh’ or ‘fixed-mesh’ techniques are used. In the moving-mesh techniques, the motion of the free surface is absorbed by moving the computational nodes located on the free surface [16, 17].

Most of the moving-mesh techniques are based on either the space–time FE formulations [18, 19] or the arbitrary Lagrangian–Eulerian [20, 21] formulations. In the applications where the deformation of the free surface is large, the moving-mesh techniques usually result in element distortions. As the element distortions grow and become unacceptable, the generation of a new mesh and the projection of the solution from the old mesh to the new mesh is essential [22]. In complex three-dimensional applications, this procedure is extremely difficult and time consuming. In such cases, computations using fixed-mesh techniques are more desirable.

The most common fixed-mesh techniques are based on the volume of fluid (VOF) [23], the level set [24, 25] and the interface-sharpening/global mass conservation (IS–GMC) methods [26, 27]. In these methods, the Navier–Stokes equations are solved over a non-moving mesh. A scalar function (or color function) acts as a marker to identify the location of the free surface. This function is transported throughout the computational domain with a transient advection equation.

In this paper we will describe the extension of our implicit hybrid FE/FV solver to flows involving two immiscible fluids based on the IS–GMC algorithm. In our implementation, we will focus on the free-surface sharpening strategy to minimize the smearing of the interface over time.

The rest of this paper is organized as follows. Section 2 presents the governing equations. Section 3 provides the solution strategy and a brief overview of our hybrid solver. Section 4 provides some implementation issues regarding the mass conservation. Some test cases are presented in Section 5 to demonstrate the performance of the current two-fluid-flow solver. Finally, in Section 6, we summarize this paper with concluding remarks.

## 2. GOVERNING EQUATIONS

The velocity and pressure fields of two isothermal incompressible and immiscible fluids, denoted by fluid<sup>A</sup> and fluid<sup>B</sup>, moving in the domain  $\mathbf{x} \in \Omega$  with boundary  $\partial\Omega$  during the time interval  $t \in (0, T)$  can be described by the incompressible Navier–Stokes equations for multiphase flows. Let  $\Omega = \Omega_t^A \cap \Omega_t^B$  where  $\Omega_t^A$  and  $\Omega_t^B$  represent volumes occupied by fluid<sup>A</sup> and fluid<sup>B</sup> at time  $t$ , respectively, and  $\phi$  denotes the VOF function such that:

$$\phi = \begin{cases} 1 & \text{on } \Omega_t^A \\ 0 & \text{on } \Omega_t^B \end{cases} \quad (1)$$

Using  $\phi$ , we can define the density,  $\rho$ , and the viscosity,  $\mu$ , as follows:

$$\rho = \rho_A + (\rho_B - \rho_A)\phi \quad (2)$$

$$\mu = \mu_A + (\mu_B - \mu_A)\phi \quad (3)$$

where the subscription ‘A’ and ‘B’ refer to fluid<sup>A</sup> and fluid<sup>B</sup>, respectively. For these two fluids, the Navier–Stokes equations can be written as:

$$\nabla \cdot \mathbf{u} = 0 \quad (4)$$

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0 \quad (5)$$

$$\rho \left[ \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right] = \rho \mathbf{g} + \nabla \cdot \boldsymbol{\sigma}, \quad \boldsymbol{\sigma} = -p \mathbf{I} + \mu (\nabla \mathbf{u} + \nabla \mathbf{u}^T) \quad (6)$$

Here,  $\mathbf{u}$ ,  $p$  and  $\mathbf{g}$  are the fluid velocity, pressure and gravitational force, respectively. The identity tensor is designated by  $\mathbf{I}$ . Equations (4)–(6) are completed by an appropriate set of boundary and initial conditions. Based on the velocity, the entire physical boundary can be generally categorized into  $\partial\Omega = \Gamma_D \cup \Gamma_n \cup \Gamma_N$  (the notation follows that in [28]) where

- $\Gamma_D: \mathbf{u} = \mathbf{u}_b$ , i.e. all components of velocity are specified, e.g. inflow boundary.
- $\Gamma_n: \mathbf{u} \cdot \mathbf{n} = (\mathbf{u} \cdot \mathbf{n})_b$ , i.e. only the velocity component normal to the boundary is specified, e.g. symmetry boundary, and
- $\Gamma_N: \boldsymbol{\sigma} \cdot \mathbf{n} = (\boldsymbol{\sigma} \cdot \mathbf{n})_b$ , e.g. outflow boundary.

where  $\mathbf{n}$  is the unit vector normal to the boundary. In addition to physical boundary conditions, at the moving interface of two fluids, the jump condition applies. For two immiscible fluids we can write

$$\begin{aligned} \mathbf{n} \cdot (\boldsymbol{\sigma} \cdot \mathbf{n}) &= \gamma \kappa \\ \mathbf{s} \cdot (\boldsymbol{\sigma} \cdot \mathbf{n}) &= 0 \end{aligned} \quad (7)$$

where  $\mathbf{s}$  is the unit vector tangent to the interface. Here,  $\gamma$  and  $\kappa$  are the coefficient of surface tension and radius of the curvature, respectively. Using Equation (7), one can obtain the following equivalent equation across the interface [29]:

$$p_A \mathbf{n} = p_B \mathbf{n} + \gamma \kappa \mathbf{n} \quad (8)$$

The momentum equation, Equation (6), is often referred to as the stress-divergence form [28], which is the dominantly adopted form in FE-based flow solvers [5, 26, 27], due to the convenience of enforcing the natural boundary conditions in the weak form of Equation (6). Alternatively, the incompressible momentum equations can also be written as

$$\rho \left[ \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right] = \rho \mathbf{g} - \nabla p + \mu \nabla^2 \mathbf{u} \quad (9)$$

which is the simplified form by taking into account  $\nabla \cdot \mathbf{u} = 0$  in Equation (6). The momentum equation in the form of Equation (9) is more commonly used in non-FE-based flow solvers [2, 3]. Our experience shows that Equations (6) and (9) yield almost identical results using our current solver. However, the outflow boundary condition should be handled properly to account for the different forms of the stress tensor.

We use the one-equation SA–DES model [9] that is derived from the SA one-equation eddy-viscosity RANS model. The equation can be written as

$$\frac{\partial \tilde{v}}{\partial t} + \mathbf{u} \cdot \nabla \tilde{v} = \frac{1}{\Psi Re} [\nabla \cdot ((v + \tilde{v}) \nabla \tilde{v}) + c_{b2} |\nabla \tilde{v}|^2] + c_{b1} \tilde{S} \tilde{v} - \frac{c_{w1} f_w}{Re} \left( \frac{\tilde{v}}{d_{DES}} \right)^2 \quad (10)$$

where  $\tilde{v}$  is the working variable in the order of the molecular viscosity  $\nu$  and  $Re$  is the Reynolds number. If the molecular viscosity is constant, then  $\nu = 1$ . In Equation (10), the second and the last term on the right-hand side are the production term and destruction term, respectively. The turbulent (eddy) viscosity  $\nu_t$  is then computed according to

$$\nu_t = \tilde{v} f_{v1}, \quad f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \quad \chi = \frac{\tilde{v}}{\nu} \quad (11)$$

In Equation (10),  $\tilde{S}$  is given by

$$\tilde{S} = S + \frac{\tilde{v}}{\kappa^2 d_{\text{DES}}^2 Re} f_{v2}, \quad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}} \quad (12)$$

where  $S$  is the magnitude of the vorticity. The wall destruction function  $f_w$  is given by:

$$f_w = g \left( \frac{1 + c_{w3}^6}{g^6 + c_{w3}^6} \right)^{1/6}, \quad g = r + c_{w2}(r^6 - r), \quad r = \frac{\tilde{v}}{\tilde{S} \kappa^2 d_{\text{DES}}^2 Re} \quad (13)$$

The DES distance to the wall is defined as

$$d_{\text{DES}} = \min(d, c_{\text{DES}} \Delta) \quad (14)$$

where  $d$  is the shortest distance to the wall and  $\Delta$  is the local grid spacing taken to be the longest distance between the cell center and all of its adjacent cell centers. Equation (14) acts as a switch between the RANS mode and the Smagorinski LES mode. When  $d \ll \Delta$ , the RANS mode is activated and when  $\Delta \ll d$  the LES mode is activated. The LES mode can be seen when the destruction term in Equation (10) is balanced with the production term, the eddy viscosity becomes proportional to  $\tilde{S} \Delta^2$ , which recovers the well-known fact that Smagorinski sub-grid scale (SGS) turbulence viscosity satisfies  $\nu_{\text{SGS}} \propto \tilde{S} \Delta^2$ . Therefore, Equation (14) plays the key role in the success of the SA-DES model. The constant  $c_{\text{DES}}$  in Equation (14) is usually chosen to be 0.65.

The constants in the equations are:

$$\begin{aligned} c_{b1} &= 0.1355, \quad \sigma = \frac{2}{3}, \quad c_{b2} = 0.622, \quad \kappa = 0.41 \\ c_{w1} &= \frac{c_{b1}}{\kappa^2} + \frac{1 + c_{b2}}{\sigma}, \quad c_{w2} = 0.3, \quad c_{w3} = 2, \quad c_{v1} = 7.1 \end{aligned} \quad (15)$$

Refer to [8] for implementation details and robust approach to solve Equation (10).

### 3. SOLUTION STRATEGY

Our hybrid incompressible flow solver is based on the pressure correction or projection method. A fractional step approach is employed to decouple the pressure from the velocity. A truly matrix-free [7, 30] (both the Generalized Minimal Residual method (GMRES) and preconditioning are matrix free) implicit cell-centered FV method is used to solve the original momentum equations to obtain an intermediate velocity field. The node-based Galerkin FE method is used to solve the Poisson equation derived from the fractional step approach for an auxiliary variable that is closely related to the real pressure. The auxiliary variable is used to update the velocity field and the pressure field. We store the velocity components at cell centers and the auxiliary variable at cell vertices, making the current hybrid solver a staggered-mesh scheme which is, however, distinct from other conventional staggered grid schemes where velocity components are usually stored at cell edges and the pressure at cell centers.

At the beginning of each time step, we are given an initial velocity field  $\mathbf{u}^n$  (not necessarily divergence free),  $\phi^n$  and  $\tilde{v}^n$  at cell centers and auxiliary variable field  $q^n$  at vertices, where the superscript ‘ $n$ ’ stands for the previous time step. At the end of the time step,  $\mathbf{u}^n$ ,  $\phi^n$ ,  $\tilde{v}^n$  and  $q^n$

will be updated to  $\mathbf{u}^{n+1}$ ,  $\phi^{n+1}$ ,  $\tilde{v}^{n+1}$  and  $q^{n+1}$ , respectively. At the same time the real pressure,  $p$ , will be updated as a by-product. To obtain  $\mathbf{u}^{n+1}$ ,  $\phi^{n+1}$ ,  $\tilde{v}^{n+1}$  and  $q^{n+1}$ , we perform the following:

*Step 1:* Set  $\mathbf{u} = \mathbf{u}^n$ ,  $\phi = \phi^n$ ,  $\tilde{v} = \tilde{v}^n$  and  $q = q^n$

*Step 2:* Solve the equation governing the motion of the interface function (Equation (16)) to update value of  $\phi$

$$\frac{\alpha_1 \phi + \alpha_0 \phi^n + \alpha_{-1} \phi^{n-1}}{\Delta t} + \mathbf{u} \cdot \nabla \phi = 0 \tag{16}$$

Note that in Equation (16), the time-dependent term has been discretized using the backward difference formula (BDF). For first-order time accurate scheme (BDF1),  $\alpha_1 = 1.0$ ,  $\alpha_0 = -1.0$  and  $\alpha_{-1} = 0.0$  and for second-order time accurate scheme (BDF2),  $\alpha_1 = 1.5$ ,  $\alpha_0 = -2.0$  and  $\alpha_{-1} = 0.5$ .

Equation (16) is solved by the implicit cell-centered FV method. The current FV solver is truly matrix free. The Jacobian-free version of the GMRES solver is implemented to solve the resulting linear system after implicit discretization. Moreover, the matrix-free lower-upper symmetric Gauss-Seidel (LU-SGS) preconditioner is adopted to accelerate the GMRES convergence. More details can be found in [1].

*Step 3:* Similarly, using the matrix-free FV method, solve the turbulent equation discretized in the BDF scheme (Equation (17))

$$\frac{\alpha_1 \tilde{v} + \alpha_0 \tilde{v}^n + \alpha_{-1} \tilde{v}^{n-1}}{\Delta t} + \mathbf{u} \cdot \nabla \tilde{v} = \frac{1}{\Psi Re} [\nabla \cdot ((v + \tilde{v}) \nabla \tilde{v}) + c_{b2} |\nabla \tilde{v}|^2] + c_{b1} \tilde{S} \tilde{v} - \frac{c_{w1} f_w}{Re} \left( \frac{\tilde{v}}{d_{DES}} \right)^2 \tag{17}$$

*Step 4:* Using the same matrix-free FV method, solve the time-discretized momentum equation (Equation (18)) to obtain the intermediate velocity field  $\tilde{\mathbf{u}}$ .

$$\rho \frac{\alpha_1 \tilde{\mathbf{u}} + \alpha_0 \mathbf{u}^n + \alpha_{-1} \mathbf{u}^{n-1}}{\Delta t} + \nabla q^n = \rho (\mathbf{g} - \tilde{\mathbf{u}} \cdot \nabla \tilde{\mathbf{u}}) + \mu \nabla^2 \tilde{\mathbf{u}} \tag{18}$$

It should also be noted that it is the auxiliary variable  $q$  instead of the real pressure  $p$  that enters the momentum equation.

The auxiliary variable is closely related to the pressure and stored at a different location from the velocity components. Because only pressure gradients, instead of the pressure itself, are needed in the momentum equation, we put the auxiliary variable on the vertices of cells. This deployment provides a convenient way to evaluate the pressure gradient using the local FE basis functions without the need of reconstruction as required for the velocity field. Refer to [1, 8] for more details.

*Step 5:* Solve the pressure correction equation to obtain the incremental auxiliary variable  $q'$  using the following equation:

$$\nabla^2 q' = \frac{\rho \alpha_1}{\Delta t} \nabla \cdot \tilde{\mathbf{u}} \tag{19}$$

Equation (19) is a Poisson equation (elliptic) that is solved by the continuous Galerkin FE method. Since the stiffness matrix is symmetric after the FE discretization, we are able to use the preconditioned conjugate gradient method to solve the system. The preconditioner is based on the SGS decomposition of the stiffness matrix that is also an incomplete LU decomposition (ILU(0)). The Eisenstat trick [31] is utilized in the implementation. With the Eisenstat trick, the preconditioner adds negligible extra cost compared with non-preconditioned solver.

*Step 6:* Update the auxiliary variable  $q$ . The auxiliary variable is updated according to:

$$q^{n+1} = q^n + q' \quad (20)$$

*Step 7:* Update the velocity  $\mathbf{u}$ . The velocity is updated according to

$$\mathbf{u}^{n+1} = \tilde{\mathbf{u}} - \frac{\Delta t}{\rho\alpha_1} \nabla q' \quad (21)$$

Since the incremental auxiliary variable  $q'$  is stored at the vertices of the mesh, and the velocity vector is stored at the cell centers, the gradient of  $q'$  can be computed via the local element interpolation functions at the cell center. Given the nodal values  $q'$  at vertices of each element, we can interpolate  $q'$  at the element center using FE base functions

$$q' = \sum_{i=1}^{\text{nen}} N_i q'_i \quad (22)$$

where 'nen' is the number of nodes per element and  $N_i$  is the interpolation function at each node. The gradient of  $q'$  at the element center can be calculate by taking gradient of  $q'$  in Equation (22) and evaluating the gradient of base functions at the element center

$$\nabla q'|_{xc} = \sum_{i=1}^{\text{nen}} (\nabla N_i)|_{xc} q'_i \quad (23)$$

where  $xc$  is the coordinate of the element center. Note we use first-order interpolation functions.

*Step 8:* In the Newton–Raphson nonlinear iteration, update and repeat Steps 1–7 until desirable convergence tolerance is obtained. In our computation, we reduce the nonlinear residual at least by three orders of magnitude.

*Step 9:* Update  $\mathbf{u}^{n+1} = \mathbf{u}$ ,  $\phi^{n+1} = \phi$ ,  $\tilde{v}^{n+1} = \tilde{v}$  and  $q^{n+1} = q$

*Step 10:* Update the real pressure  $p^{n+1}$ . The real pressure is updated according to

$$p^{n+1} = q^{n+1} - c\mu \nabla \cdot \tilde{\mathbf{u}} \quad (24)$$

where  $c=2$  if Equation (6) is used and  $c=1$  if Equation (9) is used. Note that in the current hybrid solver, the real pressure appears as a by-product in the solution procedure. The real pressure is updated utilizing the auxiliary variable field and the velocity-divergence field. The velocity divergence contains the velocity gradient information that has been obtained when solving the momentum equations. This pressure updating strategy leads to the correct physical pressure Neumann boundary condition (see [1] for detailed proof) on  $\Gamma_D$  and  $\Gamma_n$ .

As can be seen from the aforementioned ten steps, the solution procedure is very efficient. Step 5, solving a Poisson equation, is the most computational intensive step (approximately 60% of the computational cost).

#### 4. INTERFACE SHARPENING/GLOBAL MASS CONSERVATION (IS–GMC)

In Step 2, we solve purely a hyperbolic system to advance the front of the interface function with the fluid velocity. Since the interface function is a discontinuous function, its accurate representation over discrete domain is a major challenge. In fact, most of the algorithms developed for free

surface or two immiscible fluid flows suffer from global mass conservation. The error associated in solving the hyperbolic equation governing the advance of interface front results in fictitious mass transfer between the two fluids.

The numerical solution of Equation (16) will result in a diffusive solution with over shoots and undershoots in the vicinity of the interface. Since Equation (16) is solved using our advanced FV method [32], a slope limiting procedure (described in [32]) will eliminate the overshoots and undershoots. However, the diffusive interface still will introduce inaccuracies in mass conservation. To recapture the mass conservation, we sharpen the diffusive interface  $\phi$  with  $\phi_{\text{new}}$  as following:

$$\phi_{\text{new}} = \begin{cases} \beta^{1-\alpha} \phi^\alpha, & 0 \leq \phi \leq \beta \\ 1 - (1-\beta)^{1-\alpha} (1-\phi)^\alpha, & \beta \leq \phi \leq 1 \end{cases} \quad (25)$$

where  $1 \leq \alpha$  is a sharpening parameter [26, 27], and  $0 \leq \beta \leq 1$  (unknown) is a point satisfying the global conservation of mass for each fluid. To determine  $\beta$ , we satisfy the mass conservation over entire computational domain at a given time  $t$  for  $\phi_{\text{new}}$ . Note that the total mass of each fluid is the summation of the original mass plus the net amount of mass flux entering or exiting through the physical boundaries. Therefore,

$$\rho_A \int_{\Omega} \phi_{\text{new}} d\Omega = m_A + \rho_A \int_t \int_{\Gamma} \phi \mathbf{u} \cdot \mathbf{n} d\Gamma dt \quad (26)$$

$$\rho_B \int_{\Omega} (1 - \phi_{\text{new}}) d\Omega = m_B + \rho_B \int_t \int_{\Gamma} (1 - \phi) \mathbf{u} \cdot \mathbf{n} d\Gamma dt \quad (27)$$

where  $m_A$  and  $m_B$  are the initial mass of fluid<sup>A</sup> and fluid<sup>B</sup> respectively. Note that we only need to satisfy either Equation (26) or (27). Inserting Equation (25) into either equation, assuming that the parameter  $\alpha$  is given and constant, we can obtain

$$M\beta^{(1-\alpha)} + N(1-\beta)^{(1-\alpha)} = K \quad (28)$$

where  $M, N, K$  are all functions of  $\beta$ . This nonlinear equation is solved using a Newton–Raphson algorithm. Typically, with the initial guess of 0.5, the algorithm converges in two or three nonlinear iterations.

Once the parameter  $\beta$  is obtained,  $\phi$  will be replaced by  $\phi_{\text{new}}$ . It is worthy to mention that the parameter  $\beta$  will define where the true interface is at each time step.

## 5. NUMERICAL EXAMPLES

Three standard test problems for free-surface flows are chosen to validate our hybrid two-fluid flow solver. In all three problems, the non-dimensional properties of water and air are set as:

$$\frac{\rho_{\text{water}}}{\rho_{\text{air}}} = 813, \quad \frac{\mu_{\text{water}}}{\mu_{\text{air}}} = 55 \quad (29)$$

*Breaking dam:* The first test problem is the breaking dam problem [33], which is a classic test case for free-surface flows. In this problem, a column of the water is suddenly released and falls to the ground due to the gravitational force. The computational domain and the problem definition are shown in Figure 1.



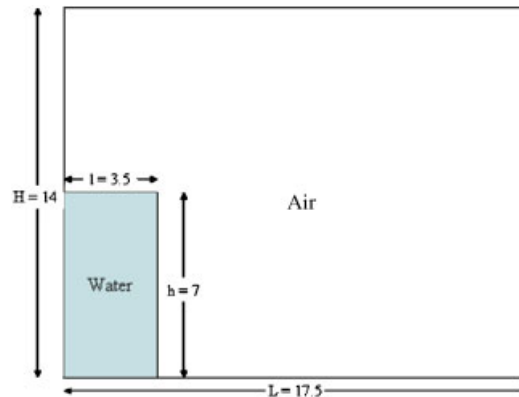


Figure 1. Breaking dam. Computational domain and the problem definition.

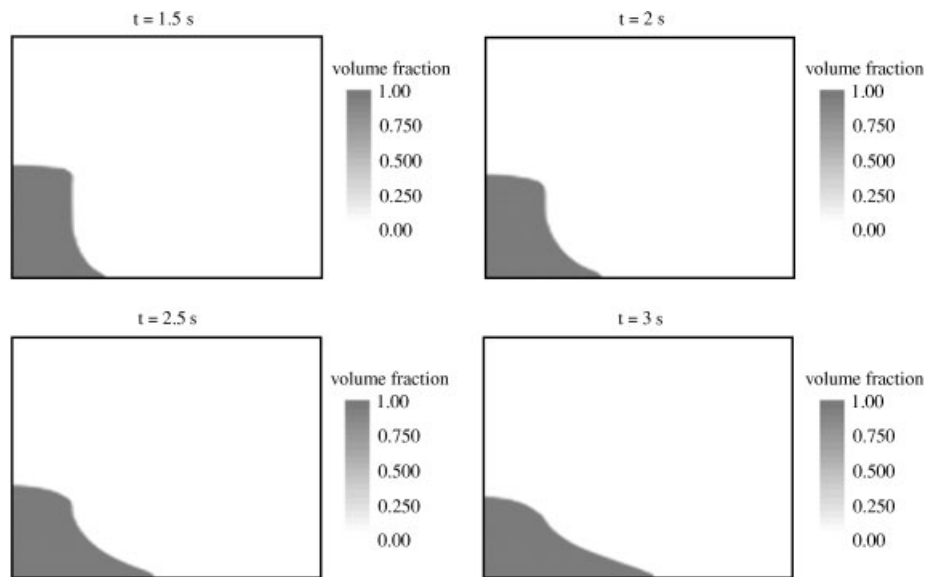


Figure 2. Breaking dam. The images show the volume of the fluid and the interface between air and water at four instances.

We used two structured meshes consisting of uniform quadrilateral elements to simulate this problem. The coarse and refined meshes are discretized with  $125 \times 100$  and  $260 \times 210$  elements in the horizontal and the vertical directions, respectively. Figure 2 shows the simulation results by plotting the VOF variable at four instances. The images in Figure 2 show that the interface between air and water is captured very well by our solver, and the transition region of the interface is very small in size. In Figure 3, the horizontal displacement of the interface front on the ground is compared with experimental data [33]. The agreement is very good for both coarse and refined

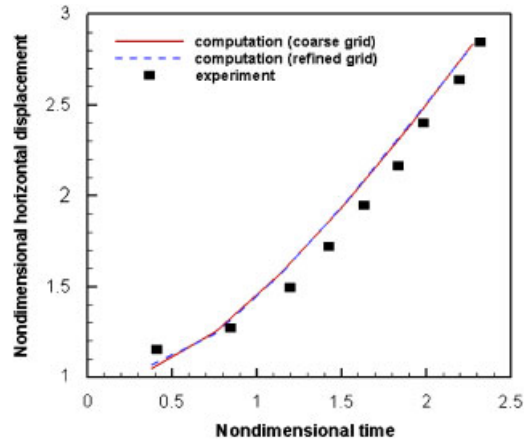


Figure 3. Breaking dam: horizontal displacement of the interface front. The symbols are experimental data, and the lines are our computed results.

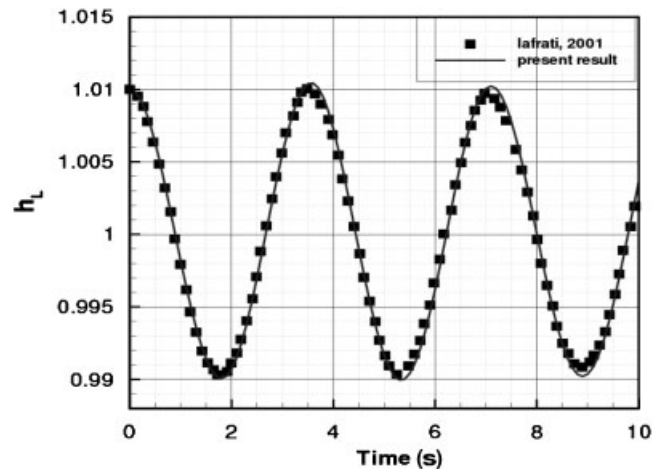


Figure 4. Oscillating flow in a tank. Time history of the air–water interface elevation on the left wall of the tank, compared with the results obtained by Iafrazi *et al.* [34].

meshes. In Figure 3, the dimensionless time and displacement are given by  $\tau = t\sqrt{2g/a}$  and  $\delta = x/a$ , where  $a$  is the initial width of the water column.

*Oscillating flow in a tank:* In our second test problem, the oscillating flow in a tank is studied. This problem further allows us to evaluate the capability of our method to capture interface dynamics (also referred to as free surface) in air–water interaction. Our computed results are compared with those obtained by Iafrazi *et al.* in [34]. The tank extends horizontally from  $x = -0.5$  to  $x = 0.5$  and from  $y = 0$  to  $y = 1.4$  in the vertical direction. The computational domain is non-dimensionalized with horizontal length. The initial free surface is described by the equation

$$y(x) = 1 - 0.01\sin(\pi x) \quad (30)$$

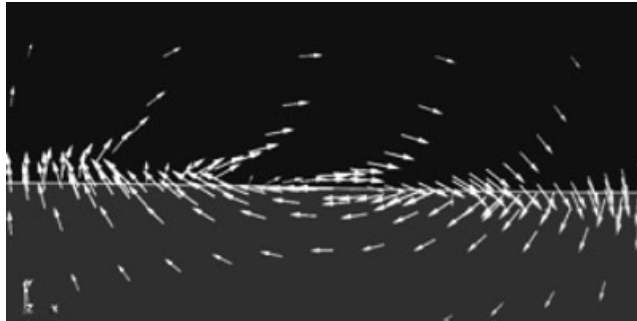


Figure 5. Oscillating flow in a tank. The interface and velocity vectors at  $t=6.75$  s. The image shows the smoothness of the interface between air and water.

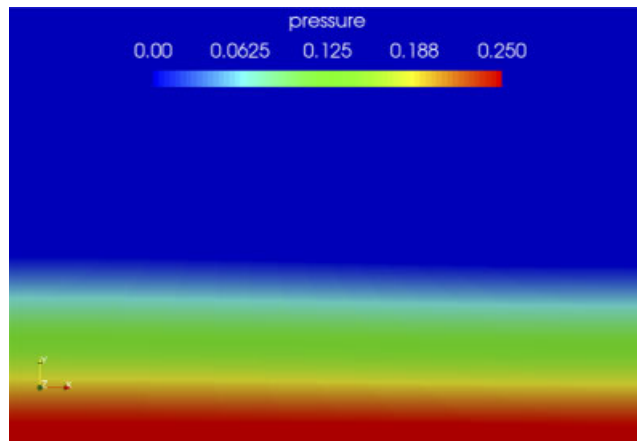


Figure 6. Oscillating flow in a tank. Dimensionless pressure contours at  $t=6.75$  s. The increase of pressure near the bottom is due to increasing hydro-pressure.

From the equation it is clear that initially the  $y$  value of the interface on the left wall is 1.01 and on the right wall is 0.99. For this flow,  $U = \sqrt{gL}$  is used as reference velocity. The Froude and Reynolds numbers are set to one and one million, respectively. The structured mesh for this test case has 40 equally spaced points in the horizontal direction and 280 points in the vertical direction, suitably clustered close to the interface region. In the region  $0.985 \leq y \leq 1.015$ , a uniform vertical spacing of  $\Delta y = 0.00025$  is adopted.

Figure 4 shows the time history of the free-surface height on the left wall of the tank for our computations and the results obtained in [34]. As we can see, the agreement between the two results can prove further that the dynamics of the free surface on the left wall is well captured. Figure 5 plots the velocity vectors around the interface between the air and water. It is clear that the velocity around the interface is smooth without any over-shoot or under-shoot oscillations. Figure 6 plots the pressure contours and it shows that toward the bottom of the tank, the pressure increases due to increasing hydro-pressure.

*NACA 0012 submerged hydrofoil:* The last test case is the NACA 0012 submerged hydrofoil at high Reynolds number and an angle of attack of  $5^\circ$ . The quantities are non-dimensionalized

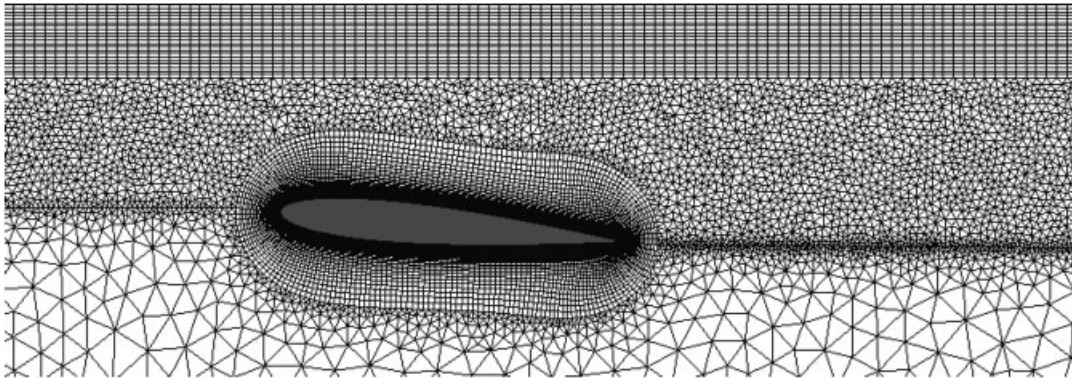


Figure 7. NACA 0012 submerged hydrofoil. The hybrid mesh near the hydrofoil.

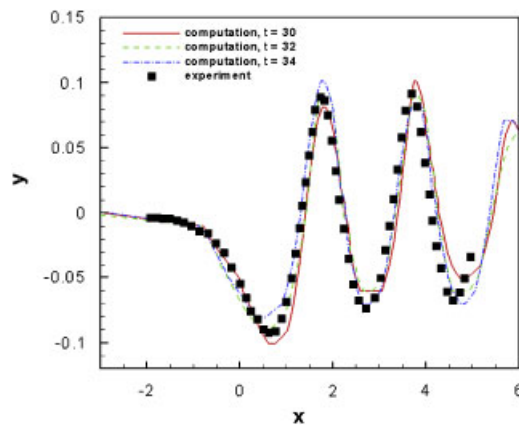


Figure 8. NACA 0012 submerged hydrofoil. The interface wave profile at three instances compared with experimental data.

with the chord length of the body and the uniform inflow velocity. Based on these scales, the Reynolds number is  $Re = 1.624 \times 10^5$  and the Froude number is  $Fr = 0.5672$ . The non-dimensional submergence at mid-chord is 1.034 in our simulation. The computational domain is 18 units in length (the upstream boundary is 6 units upstream of the leading edge) and 6 units in depth.

A hybrid computational mesh with 139 614 elements, which incorporates rectangular elements in the boundary layer near the hydrofoil surface and in the vicinity of the interface and triangular elements elsewhere are used in this computation. Figure 7 shows a close-up of the mesh near the hydrofoil.

Owing to the high Reynolds number flow in this problem, the transient flow turns turbulent very quickly. Hence, the DES turbulence model was turned on in the simulation. As we expected, the unsteady behavior in the flow was observed in our simulations. Figure 8 compares our computed interface profile with the experimental measurements reported in [35].

Figure 9 shows the velocity vectors near the interface between air and water. The continuity of the velocity field across the interface can be observed from this figure. The flow is attached to

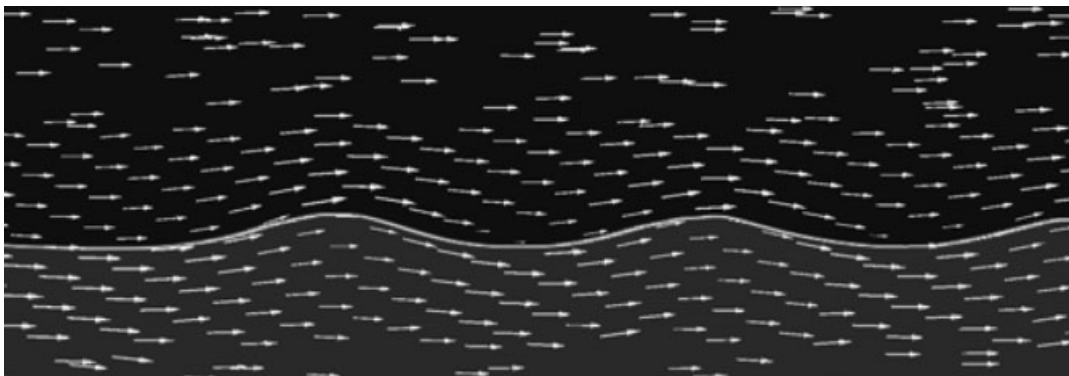


Figure 9. NACA 0012 submerged hydrofoil. Velocity vectors near the interface between the air and water. The plot is made at  $t=34$ s. The smoothness of the free surface and continuity of the velocity field can be seen from the figure.

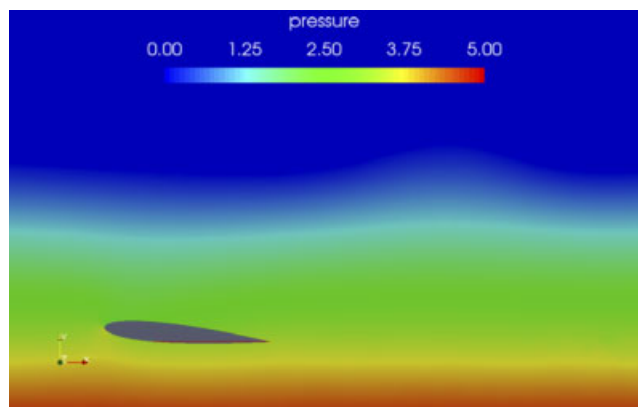


Figure 10. NACA 0012 submerged hydrofoil. The image shows the pressure distribution. The plot is made at  $t=34$ s. The smoothness of the solution and continuity of the pressure field with absent of the surface tension can be seen.

the hydrofoil, as flow is turbulent and separation is prevented. The image in Figure 9 also shows smooth velocity across the interface. Figure 10 depicts the computed pressure at  $t=34$ s. In this computation, the surface tension is ignored. Therefore, the pressure field should be continuous across the interface. The image in Figure 10 shows the continuous pressure field across the interface. It is clear that in the water phase, the hydro-pressure increases with water depth due to gravity.

## 6. CONCLUSIONS

In this paper, we have successfully extended our implicit hybrid FE/FV solver to flows involving two immiscible fluids. The solver is based on the segregated pressure correction or projection

method on staggered unstructured hybrid meshes. The GMRES matrix-free strategy was adapted to solve the governing equations using FE and FV methods. The fluid interface is captured by solving an advection equation for the volume fraction of one of the fluids. We implemented an interface sharpening strategy to minimize the smearing of the interface over time. We have developed and implemented a global mass conservation algorithm that enforces the conservation of mass for each fluid. We used LU–SGS preconditioning technique to solve the linear system of equations. Though the LU–SGS preconditioning for the momentum equations and the turbulence equation proved beneficial, a more efficient Poisson equation solver will be sought to improve the efficiency and reduce the overall computational cost of simulations.

#### ACKNOWLEDGEMENTS

Work is funded by Northrop Grumman Ship Systems and National Science Foundation.

#### REFERENCES

1. Tu S, Aliabadi S. Development of a hybrid finite volume/element solver for incompressible flows on unstructured meshes. *International Journal for Numerical Methods in Fluids* 2007; **55**(2):177–203.
2. Guermond JL, Mineev P, Shen J. An overview of projection methods for incompressible flows. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:6011–6045.
3. Timmermans LJP, Mineev PD, van de Vosse FN. An approximate projection scheme for incompressible flow using spectral elements. *International Journal for Numerical Methods in Fluids* 1996; **22**:673–688.
4. Aliabadi S, Johnson AA, Abedi J. Stabilized-finite-element/interface-capturing technique for parallel computation of unsteady flows with interfaces. *Computers and Fluids* 2003; **32**:535–545.
5. Aliabadi S, Tu S, Watts MD, Ji A, Johnson AA. Integrated high performance computational tools for simulations of transport and diffusion of contaminants in urban areas. *International Journal of Computational Fluid Dynamics* 2006; **20**(3–4):253–267.
6. Delanaye M. Polynomial reconstruction finite volume schemes for the compressible Euler and Navier–Stokes equations on unstructured adaptive grids. *Ph.D. Thesis*, Universite de Liege, 1996.
7. Tu S, Watts MD, Fuller A, Patel R, Aliabadi S. Development and performance of CaMEL\_Aero, a truly matrix-free, parallel and vectorized unstructured finite volume solver for compressible flows. *Proceedings of the 25th Army Science Conference*, Orlando, FL, 2006.
8. Tu S, Aliabadi S, Patel R, Watts M. An implementation of the Spalart–Allmaras DES model in an implicit unstructured hybrid finite volume/element solver for incompressible turbulent flow. *International Journal for Numerical Methods in Fluids*; to appear.
9. Spalart PR, Jou W-H, Strelets M, Allmaras SR. Comments on the feasibility of LES for wings and on hybrid RANS/LES approach. *Proceedings of the 1st AFOSR International Conference on DNS/LES*, Columbus, OH, 1997.
10. Nikitin NV, Nicoud F, Wasistho B, Squires KD, Spalart PR. An approach to wall modeling in large-eddy simulations. *Physics of Fluids* 2000; **12**(7):1629–1632.
11. Strelets M. Detached eddy simulation of massively separated flows. *The 39th AIAA Aerospace Sciences Meeting and Exhibits*, Reno, NV, 2001. AIAA Paper 2001-0879.
12. Scott AM, Matthew BS, Russell MC, James RF. DES grid resolution issues for vortical flows on a delta wing and an F-18C. *The 41st Aerospace Sciences Meeting and Exhibit*, Reno, NV, 2003. AIAA Paper 2003-1103.
13. Kapadia S, Roy S, Wurtzler K. Detached eddy simulation over a reference Ahmed car model. *The 41st Aerospace Sciences Meeting and Exhibit*, Reno, NV, 2003. AIAA Paper 2003-0857.
14. Forsythe JR, Hoffmann KA, Cummings RM, Squires KD. Detached eddy simulation with compressibility corrections applied to a supersonic axisymmetric base flow. *Journal of Fluids Engineering* 2002; **124**(4):911–923.
15. Rosen BS, Laiosa JP. Splash nonlinear and unsteady free-surface analysis code for grand prix yacht racing. *The Thirteenth Chesapeake Sailing Yacht Symposium*, Annapolis, MD, January 1997.
16. Aliabadi S, Tezduyar T. Stabilized-finite-element/interface-capturing technique for parallel computation of unsteady flows with interfaces. *Computer Methods in Applied Mechanics and Engineering* 2000; **190**:243–261.

17. Mikkola T. Testing of two FINFLO-based free-surface codes using Eulerian flow over a Gaussian ground elevation. *Report D-56*, Ship Laboratory, Helsinki University of Technology, 1999.
18. Aliabadi SK, Tezduyar TE. Space-time finite element computation of compressible flows involving moving boundaries and interfaces. *Computer Methods in Applied Mechanics and Engineering* 1993; **107**:209–223.
19. Hughes TJR, Hulbert GM. Space-time finite element methods for elastodynamics: formulations and error estimates. *Computer Methods in Applied Mechanics and Engineering* 1998; **66**:339–363.
20. Donea J. An arbitrary Lagrangian–Eulerian finite element method for transient fluid–structure interactions. *Computational Mechanics* 1982; **33**:689–723.
21. Farhat C, Lesoinne M, Maman N. Mixed explicit/implicit time integration of coupled aeroelastic problems: three-field formulation, geometric conservation and distributed solution. *International Journal for the Numerical Methods in Fluids* 1995; **21**:807–835.
22. Johnson A, Tezduyar T. Advanced mesh generation and update methods for 3D flow simulations. *Computational Mechanics* 1999; **23**:130–143.
23. Hirt W, Nichols BD. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics* 1981; **39**:201–225.
24. Sussman M, Smareka P, Osher S. A level set approach for computing incompressible two-phase flows. *Journal of Computational Physics* 1994; **114**:146–168.
25. Sethian JA. *Level Set Method*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press: Cambridge, 1996.
26. Aliabadi S, Johnson A, Zellars B, Abatan A, Berger C. Parallel simulation of flows in open channels. *Journal of Future Generation Computer Systems* 2002; **18**(5):627–637.
27. Aliabadi S, Abedi J, Zellars B. Parallel finite element simulation of Mooring forces on floating objects. *International Journal for Numerical Methods in Fluids* 2003; **41**:809–822.
28. Gresho PM, Sani RL. Isothermal laminar flow. *Incompressible Flow and the Finite Element Method*, vol. 2. Wiley: New York, 2000.
29. García JJ, Valls A. ODD level set: a new method for simulation of free-surface problems. *Journal of Maritime Research* 2006; **III**(1):27–40.
30. Tu S, Aliabadi S, Johnson AA, Watts MD. A robust parallel implicit finite volume solver for high-speed compressible flows. *The 43rd AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, 2005. AIAA Paper 2005-1396.
31. Eisenstat SC. Efficient implementation of a class of preconditioned conjugate gradient methods. *SIAM Journal on Scientific and Statistical Computing* 1981; **2**:1–4.
32. Tu S, Aliabadi S. A slope limiting procedure in discontinuous Galerkin finite element method for gas dynamics applications. *International Journal for Numerical Analysis and Modeling* 2005; **V2**:163–178.
33. Martin JC, Moyce WJ. An experimental study of the collapse of a liquid column on a rigid horizontal plane. *Philosophical Transactions of the Royal Society of London* 1952; **A244**:312–324.
34. Iafrati A, Mascio AD, Campana EF. A level set technique applied to unsteady free surface flows. *International Journal for Numerical Methods in Fluids* 2001; **35**:281–297.
35. Duncan JH. The breaking and non-breaking wave resistance of a two-dimensional hydrofoil. *Journal of Fluid Mechanics* 1983; **126**:507–520.