

## Computation of fluid–solid and fluid–fluid interfaces with the CIP method based on adaptive Soroban grids—An overview

Takashi Yabe<sup>1,\*</sup>, Kenji Takizawa<sup>2</sup>, Tayfun E. Tezduyar<sup>3</sup> and Hyo-Nam Im<sup>1</sup>

<sup>1</sup>*Department of Mechanical Sciences and Engineering, Tokyo Institute of Technology, O-okayama, Meguro, Tokyo 152-8552, Japan*

<sup>2</sup>*National Maritime Research Institute, 6-38-1, Shinkawa, Mitaka-shi, Tokyo 181-0004, Japan*

<sup>3</sup>*Mechanical Engineering, Rice University-MS 321, 6100 Main Street, Houston, TX 77005, U.S.A.*

### SUMMARY

We provide an overview of how fluid–solid and fluid–fluid interfaces can be computed successfully with the constrained interpolation profile/cubic interpolated pseudo-particle (CIP) method (*J. Comput. Phys.* 1985; **61**:261–268; *Comput. Phys. Commun.* 1991; **66**:219–232; *Comput. Phys. Commun.* 1991; **66**:233–242; *J. Comput. Phys.* 2001; **169**:556–593) based on adaptive Soroban grids (*J. Comput. Phys.* 2004; **194**:57–77). In this approach, the CIP combined unified procedure (CCUP) technique (*J. Phys. Soc. Jpn* 1991; **60**:2105–2108), which is based on the CIP method, is combined with the adaptive Soroban grid technique. One of the superior features of the approach is that even though the grid system is unstructured, it still has a simple data structure that renders remarkable computational efficiency. Another superior feature is that despite the unstructured and collocated nature of the grid, high-order accuracy and computational robustness are maintained. In addition, because the Soroban grid technique does not have any elements or cells connecting the grid points, the approach does not involve mesh distortion limitations. While the details of the approach and several numerical examples were reported in (*Comput. Mech.* 2006; published online), our objective in this paper is to provide an easy-to-follow description of the key aspects of the approach. Copyright © 2007 John Wiley & Sons, Ltd.

Received 21 January 2007; Accepted 6 February 2007

KEY WORDS: fluid–solid interface; fluid–fluid interface; CIP method; Soroban grid; adaptive grid

### 1. INTRODUCTION

Computation of fluid–solid interfaces (including fluid–structure and fluid–particle interactions), which is one of the most challenging classes of problems in computational engineering and sciences,

\*Correspondence to: Takashi Yabe, Department of Mechanical Sciences and Engineering, Tokyo Institute of Technology, O-okayama, Meguro, Tokyo 152-8552, Japan.

†E-mail: yabe@mech.titech.ac.jp

has been receiving much attention in recent years. The numerical challenges include: (i) accurate representation of the flow field near the interface; (ii) accurate advection of the vortices; (iii) grid generation and motion; (iv) robustness and accuracy of the fluid–solid coupling technique; and (v) computational efficiency. The challenges increase further when the computation also involves fluid–fluid interfaces (including free-surface and multifluid flows). Robust and efficient computation of complex or very unsteady fluid–fluid interfaces may call for an algorithmic component different than the one used for the fluid–solid interfaces involved in the same problem. As examples of flow problems with both fluid–solid and fluid–fluid interfaces, we can mention solid–liquid–gas interactions and fluid–structure interactions with free surfaces, such as ship hydrodynamics.

In recent decades, a substantial number of finite-element interface tracking (moving grid) methods have been developed for computation of fluid–solid interfaces, including fluid–structure interactions (see, for example [1–17]). For computation of complex or very unsteady fluid–fluid interfaces, the preferred methods have typically fallen into the category of interface-capturing (non-moving grid) techniques. The fluid–fluid interface is captured within the resolution of the grid covering the area where the interface is. A consequence of the mesh not moving to track the interface is that, if applied to computation of fluid–solid interfaces, independent of how accurately the interface geometry is represented, the resolution of the boundary layer will be limited by the resolution of the fluid mesh where the interface is. The mixed interface-tracking/interface-capturing technique was introduced in [18] for computations involving both fluid–solid interfaces that need to be accurately tracked with a moving mesh method and fluid–fluid interfaces that are too complex or unsteady to be tracked and therefore require an interface-capturing technique.

Although the fluid–solid interfaces involved in such mixed problems can in most cases be successfully computed with interface-tracking techniques, in some cases the grid distortion rates may require more frequent remeshing than we are willing to live with. A method that could free us from mesh moving and distortion concerns without compromising the accuracy near the fluid–solid interfaces would of course be very desirable. The adaptive-grid method introduced recently in [19] accomplishes that goal by combining the constrained interpolation profile/cubic interpolated pseudo-particle (CIP) combined unified procedure (CCUP) technique [20], which is based on the CIP method developed by Yabe *et al.* [21–24] for solving hyperbolic equations, with the ‘Soroban<sup>‡</sup> grid’ technique [25], which is an unstructured and collocated grid technique.

The combined approach has a number of superior features originating from its advanced components. The CIP method has superior accuracy in dealing with the advection terms. It has minimal numerical diffusion and is known as a powerful multifluid solver. It is essentially a semi-Lagrangian scheme and has a third-order accuracy in space and time. It has also been extended to the computation of incompressible flows in the framework of compressible flows. The CIP method uses a primitive Euler method in solving all fluid-like equations, with separate treatment of the advection and non-advection parts, which is typical of a semi-Lagrangian approach. Therefore, a formulation that can solve incompressible and compressible flows simultaneously is easily obtained. The advection part is solved with the CIP method and non-advection part is calculated with the CCUP method [20], where a Poisson’s equation is solved for pressure to treat compressible and incompressible flows in a unified fashion. Although the solution technique is based on a collocated-grid approach, high-order accuracy and robustness are maintained. The Soroban grid technique does

---

<sup>‡</sup>‘Soroban’ is the Japanese word for abacus, reflecting the nature of the grid system with the grid points positioned along parallel lines.

not have any elements or cells connecting the grid points, and therefore the approach is free from mesh (or grid) distortion limitations. Furthermore, even though the grid system is unstructured and adaptive, it still has a simple data structure, rendering remarkable computational efficiency. A single interpolation process is used for both calculating the advection terms and estimating the values at the new grid points from the old grid points, and therefore there is no additional computational overhead associated with the grid adaptation.

The details of the combined approach and several numerical examples were reported in [19]. In this paper, we provide an easy-to-follow description of its key aspects. In Section 2, we briefly describe the CIP method and the Soroban grid technique. The pressure-based algorithm in the context of a primitive Euler scheme is described in Section 3. A numerical example is presented in Section 4, and the concluding remarks are given in Section 5.

## 2. CIP METHOD AND SOROBAN GRID

### 2.1. Calculation of the advection terms with the CIP method

With the Soroban grid, one-dimensional CIP is used extensively. Here, we explain the basics of the CIP method by using an advection equation:

$$\frac{\partial f}{\partial t} + u \frac{\partial f}{\partial x} = 0 \quad (1)$$

The CIP method uses the value  $f$  as well as its spatial derivative  $\partial f/\partial x$  ( $\equiv g$ ) to construct a cubic interpolation between two grid points. If we differentiate Equation (1) with respect to the spatial variable  $x$ , we obtain:

$$\frac{\partial g}{\partial t} + u \frac{\partial g}{\partial x} = -\frac{\partial u}{\partial x} g \quad (2)$$

In the simplest case where the velocity  $u$  is constant, Equation (2) coincides with Equation (1) because  $\partial u/\partial x = 0$ . Using these equations, we can trace the time evolution of  $f$  and  $g$ .

Given the values of  $f$  and  $g$  at the two grid points, the spatial profile between these points can be interpolated by a cubic polynomial  $F(x) = ax^3 + bx^2 + cx + d$ . Thus, the profile at time step  $n + 1$  can be obtained by transporting the profile by  $u\Delta t$  so that

$$\begin{aligned} f_i^{n+1} &= F(x_i - u_i \Delta t) = a_i \xi^3 + b_i \xi^2 + g_i^n \xi + f_i^n \\ g_i^{n+1} &= \frac{dF}{dx}(x_i - u_i \Delta t) = 3a_i \xi^2 + 2b_i \xi + g_i^n \\ a_i &= \frac{g_i + g_{iup}}{D^2} + \frac{2(f_i - f_{iup})}{D^3} \\ b_i &= \frac{3(f_{iup} - f_i)}{D^2} - \frac{2g_i + g_{iup}}{D} \end{aligned} \quad (3)$$

where  $\xi \equiv -u_i \Delta t$ ,  $D = -\Delta x$  and  $iup = i - 1$  for  $u > 0$ , and  $D = \Delta x$  and  $iup = i + 1$  for  $u < 0$ .

### 2.2. Difficulty with attaining high-order accuracy when using coordinate transformation

There have been many attempts to decrease the number of grid points in describing a complex object or surface. When the CIP scheme is applied in a curvilinear coordinate system, however, the third-order accuracy achieved with uniform grids severely deteriorates, and the accuracy is degraded to first order in the deformed mesh [25]. Therefore, the degradation of accuracy caused by introducing a curvilinear coordinate system would cancel the advantage of the CIP method that originally has a third-order accuracy in space and time.

To build a new accurate scheme, first we need a tool for estimating the accuracy. Although it is easy to analyse the accuracy for the Cartesian grids, it is not always possible for complex grid systems. If the accuracy deteriorates because of introducing an adaptive grid, such a grid system would not be valuable. The difficulty in estimating the accuracy can easily be illustrated with the following example. Let  $\xi$  be the adaptive grid coordinate. Then Equation (1) is transformed into a simple advection equation in  $\xi$  space:

$$\frac{\partial f}{\partial t} + U \frac{\partial f}{\partial \xi} = 0 \quad (4)$$

where the contravariant velocity  $U = u/(dx/d\xi)$ . Since the grid points are equally spaced in the  $\xi$  space, it would be easy to obtain high-order accuracy if the scheme stays only in the  $\xi$  space. In the calculation of  $U$ , however, the accuracy might deteriorate because of the  $x$ - $\xi$  transformation. There is some misunderstanding of the accuracy of the finite difference approximation as explained below. Even if the term  $dx/d\xi$  is approximated by the fourth-order finite differences

$$\frac{dx}{d\xi} = \frac{-x_{i+2} + 8(x_{i+1} - x_{i-1}) + x_{i-2}}{12\Delta\xi} \quad (5)$$

the numerical result does not necessarily attain fourth-order accuracy.

The reason behind this paradox can be illustrated with the following example. The discussion of the accuracy relies on the Taylor expansion,

$$x(\xi + \Delta\xi) = x(\xi) + \frac{dx}{d\xi} \Delta\xi + \frac{d^2x}{d\xi^2} \frac{\Delta\xi^2}{2} + \dots \quad (6)$$

However, this expansion is valid only if the function is analytical and there exists all orders of derivatives. If the function is in the form of a step function,  $dx/d\xi$  is infinite and the Taylor expansion given by Equation (6) is no longer valid and the accuracy based on this expansion has no meaning. Since it is very difficult to realize the grid arrangement so that the grid spacing is smooth enough to satisfy the continuities of all the derivatives, an accurate scheme based on the finite differences is hardly realized. Some examples of this type were illustrated with Figures 4 and 6 in [25].

### 2.3. Accurate adaptive-grid computation with the Soroban-grid CIP method

To resolve such difficulties, the CIP method has recently been upgraded to include adaptive grids with assurance of both high-order accuracy and robustness. The new grid system is called the Soroban grid [25]. The schematics of a Soroban grid is shown in Figure 1. The grid system consists of straight lines and grid points moving along those lines, like how it is in an abacus. The

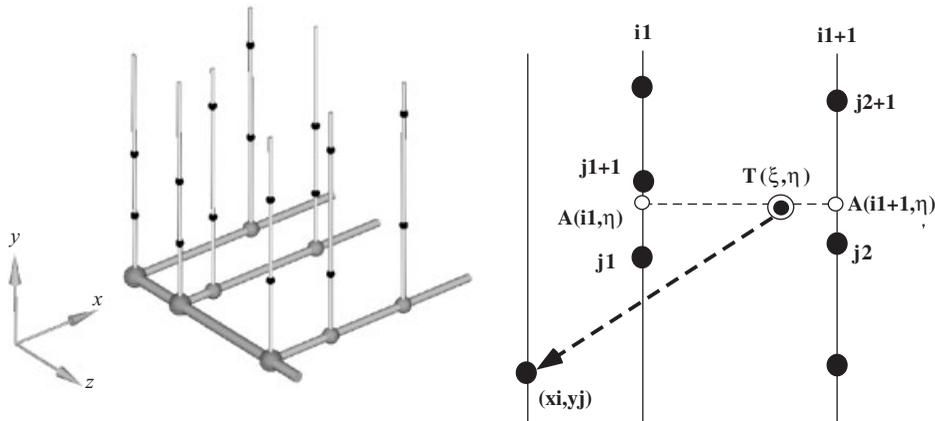


Figure 1. Soroban grid arrangement. Left: 3D view. Right: view on a plane.

lines on a plane move in parallel, and the planes also move in parallel. The length of each line and the number of grid points along each line can be variable.

To understand how the advection equations are solved with the Soroban grid, let us consider the grid lines and points on a plane shown in Figure 1 (right), where the vertical lines in the  $y$ -direction are spaced in the  $x$ -direction, and the grid points move along each line. Let  $(x_i, y_j)$  be the point of interest. If its upstream departure point  $T$  is given as  $(\xi, \eta) = (x_i - u\Delta t, y_j - v\Delta t)$ , the solution  $f^{n+1}$  of Equation (1) at  $(x_i, y_j)$  is simply given by the value at  $T$ . As in the CIP formulation in the Cartesian grid,  $T$  must be interpolated from the neighbouring points.

For such interpolations, first a pair of lines satisfying  $x_{i1} < \xi < x_{i1+1}$  are searched. We need to remember that  $x_i$  can be far from  $x_{i1}$  and  $x_{i1+1}$  so that large Courant–Friedrichs–Lewy (CFL) ( $u\Delta t/\Delta x, v\Delta t/\Delta y$ ) computations are possible. Next, two pairs of points satisfying  $y_{j1} < \eta < y_{j1+1}$  and  $y_{j2} < \eta < y_{j2+1}$  are searched along the two lines at  $x = x_{i1}$  and  $x = x_{i1+1}$ , respectively. The interpolation that uses the neighbouring grid points found here is performed as explained below.

- (1) Apply one-dimensional CIP to the vertical straight lines (along the  $y$ -direction), giving  $A_{i1, \eta}$  and  $A_{i1+1, \eta}$ .
- (2) Calculate  $T$  by one-dimensional CIP along the straight line connecting  $A_{i1, \eta}$  and  $A_{i1+1, \eta}$  (along the  $x$ -direction).

In Step (1),  $A$  and only  $\partial A/\partial y (= \partial_y A)$  are readily obtained by the CIP interpolation. In Step (2), however, we need  $A$  and  $\partial A/\partial x (= \partial_x A)$  in order to use the CIP interpolation to calculate  $T$ . Since  $\partial A/\partial x$  is not yet calculated in the first step, we must devise a way of estimating it. In a previous paper [26], we proposed to use a first-order scheme because the derivative in the direction perpendicular to the propagation direction is not sensitive and hence can be estimated only roughly by a linear interpolation. Such a splitting scheme was called the ‘Type-M’ scheme.

Although the Type-M scheme is sufficient for many applications, a little more accurate scheme is possible at the cost of additional memory requirement. This scheme was proposed by Aoki [27] and is called the ‘Type-C’ scheme. In this scheme, independent variables are  $f, g_x, g_y$  and  $\partial_{xy} f (= \partial_x g_y = \partial_y g_x)$  in two dimensions. Instead of using linear interpolation for  $g_x$  in the

$y$ -direction, the one-dimensional CIP scheme is applied to the advection of  $(\partial_x f)$  and  $\partial_y(\partial_x f)$ . The same can be said for the  $x$ -direction, that is, the one-dimensional CIP scheme is applied to the advection of  $(\partial_y f)$  and  $\partial_x(\partial_y f)$ .

Since this scheme uses only the one-dimensional CIP method without coordinate transformation, it is able to retain the third-order accuracy in space and time even for deformed grids such as the one demonstrated by Figure 10 in [25].

#### 2.4. Grid motion with the monitoring function

The grid movement along a line is determined by using a monitoring function defined as follows:

$$M(x, y, t) \equiv \left( 1 + \alpha \left( \frac{\partial f}{\partial y} \right)^2 \right)^{1/2} + \beta \left| \frac{\partial^2 f}{\partial y^2} \right| \quad (7)$$

where the two parameters  $\alpha$  and  $\beta$  can be chosen depending on the problem computed. The monitoring function  $M$  becomes large in regions with larger gradients. Since the Soroban grid points are on straight lines in the  $y$ -direction, it is much easier to generate the adaptive-grid points along those lines. The reorganization of the grid points can easily be performed by using an accumulated monitoring function  $I(x, y) = \int_0^y M(x, y) dy$ . If we divide the accumulated function into equal pieces, the boundaries of these pieces give the  $y$  coordinate of the new grid points. It is easy to see that the grid spacing  $\Delta y$  becomes small where the monitoring function  $M$  is large. Therefore, the grid points are concentrated in regions where the spatial gradient is large. This procedure is repeated for every time step.

To extend the procedure to multidimensional cases, we can define a monitoring function for each spatial direction, and move the planes, lines, and grid points independently. Alternatively, we can define, as was done in [19], a multidimensional monitoring function, which would simply be the multidimensional extension of the definition given by Equation (7).

The CIP method is suitable for this grid system because it uses only two stencils to construct a cubic-interpolation function, and the calculation of large CFL numbers ( $>10$ ) with locally refined grids can be easily performed. Grid generation and searching of the upstream departure point are very simple and the process is essentially mesh free.

#### 2.5. Handling the grid motion and advection in a single process

Remembering that the CIP method relies on accurate interpolation between the grid points, we can use the method to calculate a value at a new grid point from neighbouring old grid points. Let us consider the procedure shown in Figure 2. In a conventional method, we first calculate the advection terms in the old grid by locating the upstream departure point in that old grid, and then move the grid points to new positions. The values at the new grid points are determined by interpolation from the values updated after the advection calculations in the old grid.

In the Soroban grid CIP method, on the other hand, we can perform these two actions in a single process. We first generate the new grid points, which are completely independent from the old points. Then the upstream departure point for each grid point in the new grid is searched for in the old grid. It is easy to see that transporting the value from this upstream departure point takes care of both the calculation of the advection terms and the interpolation from the old grid points.

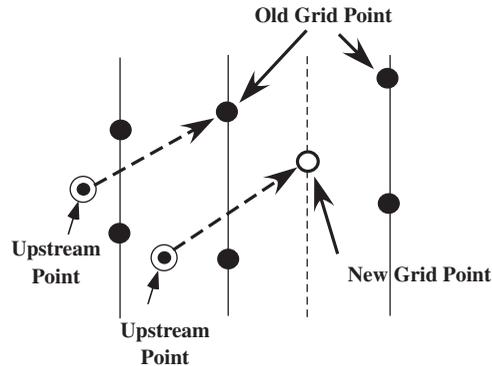


Figure 2. Advection calculation and grid motion can be handled in a single process by transporting the upstream value directly to a new grid point.

### 3. A PRESSURE-BASED ALGORITHM IN A PRIMITIVE EULER SCHEME

#### 3.1. Semi-Lagrangian formulation

To apply the Soroban grid CIP technique to fluid mechanics equations, we extend Equation (1) to an equation that includes a non-advection term:

$$\frac{\partial f}{\partial t} + u \frac{\partial f}{\partial x} = g \quad (8)$$

This equation can be re-written in terms of the Lagrangian derivative along the trajectory:

$$\frac{df}{dt} = g \quad (9)$$

The solution of Equation (9) can be obtained by integrating  $g$  along the trajectory:

$$f^{n+1} = f^* + \int g \, dt \quad (10)$$

where  $f^*$  represents the value at the upstream point given by Equation (3). The integral of  $g$  is approximated by  $g\Delta t$  using the value of  $g$  implicitly as will be explained in Section 3.2.

#### 3.2. Finite difference approximation in the Soroban grid

The non-advection terms of the fluid mechanics equations are implicitly solved as

$$\rho^{n+1} - \rho^* = -\rho^* \nabla \cdot \mathbf{u}^{n+1} \Delta t, \quad \rho^* C_v (T^{n+1} - T^*) = -P_{TH} \nabla \cdot \mathbf{u}^{n+1} \Delta t \quad (11)$$

where  $P_{TH} = T(\partial p / \partial T)_\rho$  and  $\mathbf{u}^{n+1}$  in this equation is given by the equation of motion:

$$\mathbf{u}^{n+1} - \mathbf{u}^* = -\frac{\nabla p^{n+1}}{\rho^*} \Delta t \quad (12)$$

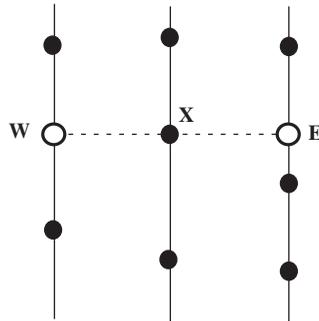


Figure 3. CCUP method in the Soroban grid. Solid circles are the grid points. The finite difference approximation at point  $X$  needs the values at  $E$  and  $W$ .

When the differential operators on the right-hand side are approximated by finite differences, we must take special care in the Soroban grid. As shown in the 2D example of Figure 3, the finite differences along the vertical lines are readily obtained. In the  $x$ -direction, however, we have no corresponding points  $E$  and  $W$ . We propose to use the CIP interpolation for obtaining  $E$  and  $W$  from the neighbouring points along the corresponding vertical line.

### 3.3. Pressure solver in the Soroban grid

In the CCUP method [20, 24], the pressure equation

$$\nabla \cdot \left( \frac{1}{\rho^*} \nabla p^{n+1} \right) = \frac{p^{n+1} - p^*}{\Delta t^2 \left( \rho^* C_s^2 + \frac{P_{TH}^2}{\rho^* C_v T^*} \right)} + \frac{\nabla \cdot \mathbf{u}^*}{\Delta t} \quad (13)$$

is solved first, then  $p^{n+1}$  is used in Equation (12) to obtain  $\mathbf{u}^{n+1}$ , and  $\rho^{n+1}$  and  $T^{n+1}$  are obtained from Equation (11).

The strategy described in Section 3.2 can be used for the finite difference approximation of all the derivatives. With those derivatives already obtained for the stage denoted by superscript ‘\*’, the finite difference forms can readily be constructed with  $\partial p^*/\partial x$  and  $\partial p^*/\partial y$  in the CIP interpolation. In calculation of the pressure,  $\partial p^{n+1}/\partial x$  and  $\partial p^{n+1}/\partial y$  needed for the CIP interpolation at  $E$  and  $W$  are not yet available. Therefore, the matrix system of Equation (13) would need to involve as unknowns not only  $p^{n+1}$  but also  $\partial p^{n+1}/\partial x$  and  $\partial p^{n+1}/\partial y$ .

To avoid that complication, we employ an iteration procedure as shown below

$$p^{n+1} = p^{[m-1]} + \delta p^{[m]} \quad (14)$$

where  $m$  is the iteration counter. Then Equation (13) is modified as follows:

$$\nabla \cdot \left( \frac{1}{\rho^*} \nabla \delta p^{[m]} \right) = - \nabla \cdot \left( \frac{1}{\rho^*} \nabla p^{[m-1]} \right) + \frac{p^{[m-1]} + \delta p^{[m]} - p^*}{\Delta t^2 \left( \rho^* C_s^2 + \frac{P_{TH}^2}{\rho^* C_v T^*} \right)} + \frac{\nabla \cdot \mathbf{u}^*}{\Delta t} \quad (15)$$

For calculation of the first term on the right-hand side involving  $p^{[m-1]}$ , the CIP interpolation is used. The matrix solver is applied only to the terms involving  $\delta p^{[m]}$ , where we use a linear interpolation to estimate the points  $E$  and  $W$  in Figure 3. We continue the iterations by incrementing the counter  $m$ .

#### 4. NUMERICAL EXAMPLE

In the test computation presented here, we apply the Soroban grid CIP technique to incompressible flow past a circular cylinder where Reynolds number is 100. The dimensions of the computational domain are  $60 \times 16$  and the cylinder is located at  $(8, 8)$ , where the values are normalized by the cylinder diameter. The location of the cylinder and the lateral dimension of the domain are the same as those reported in [28]. The initial Soroban grid is shown in Figure 4. The Soroban lines are parallel to the vertical ( $y$ ) axis, and the grid points are placed on each line in a way to refine the grid where it is needed. We also put grid points inside the cylinder, otherwise the interpolations become more complicated. The grids are generated from monitoring functions that depend on the vorticity and the distance from the cylinder surface:

$$M_1(x, y) = \frac{1}{0.005 + 0.1 \times \text{Minimum}(5, D(x, y))} \quad (16)$$

$$D(x, y) = \sqrt{(x - 8)^2 + (y - 8)^2} - 0.5 \quad (17)$$

$$M_2(x, y, t) = 1 + 50 \left| \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \right| \quad (18)$$

$$M(x, y, t) = \text{Minimum} \left( \sum_{k=1}^N M_k(x, y) / N, \Delta x_{\max} / \Delta x_{\min} \right) \quad (19)$$

The no-slip boundary condition is imposed on the cylinder surface, and inside the cylinder both velocity components are set as  $u = v = 0$ .

The drag and lift coefficients,  $C_D$  and  $C_L$ , are defined as follows:

$$C_D = \frac{F_D}{\frac{1}{2} \rho U_\infty^2 D} \quad (20)$$

$$C_L = \frac{F_L}{\frac{1}{2} \rho U_\infty^2 D} \quad (21)$$

where  $F_D$  and  $F_L$  are the drag and lift forces. The number of grid points is initially about 4000, eventually increased up to 9000. The computed values for the drag and lift coefficients and the Strouhal number are  $1.375 \pm 0.009$ ,  $\pm 0.27$  and 0.16. These values are in good agreement with those reported in [28].

Figure 5 is a snapshot of the adaptive grid. The number of grid points changes in time as shown in Figure 6. After  $t = 80$ , the number of grid points stays at about the same value, because the number of wake vortices in the computational domain remains almost constant.

We also tested the dependence of the solution on the interpolation described in Section 3.2. In Figure 7, we depict the result obtained with ‘linear interpolation’ as well as the result obtained with

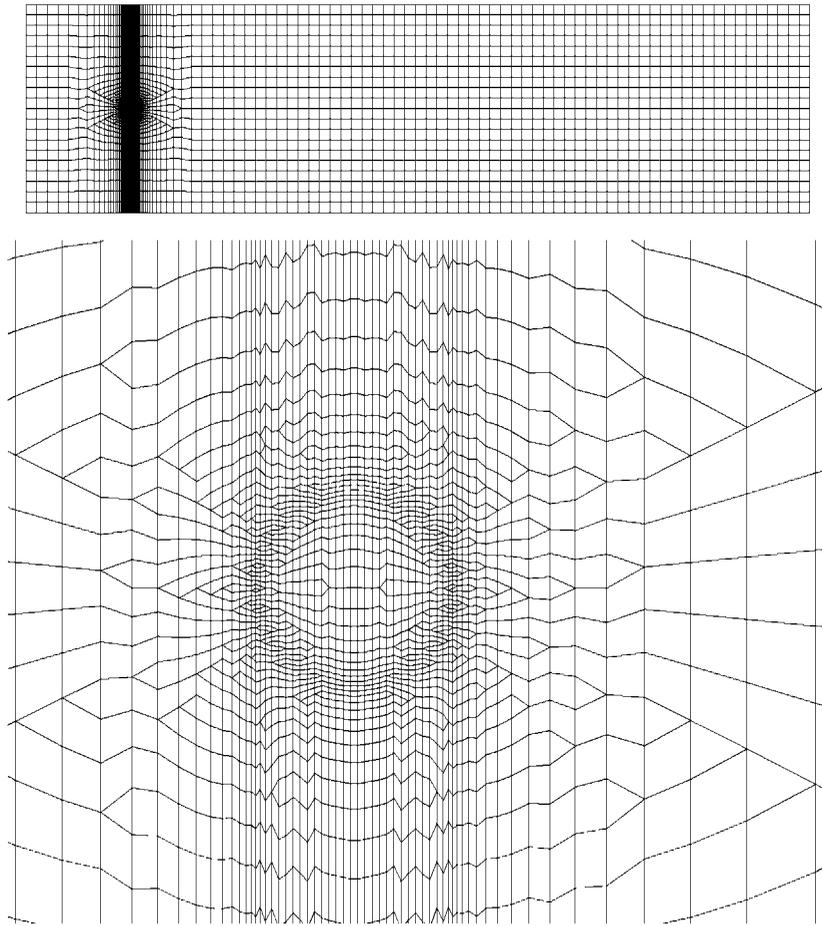


Figure 4. Flow past a cylinder. Initial grid arrangement. The Soroban lines are parallel to the vertical ( $y$ ) axis. Non-vertical lines connecting the grid points are for visualization purpose only and have no significance in the actual computation. Top: entire grid image; bottom: local view near the cylinder.

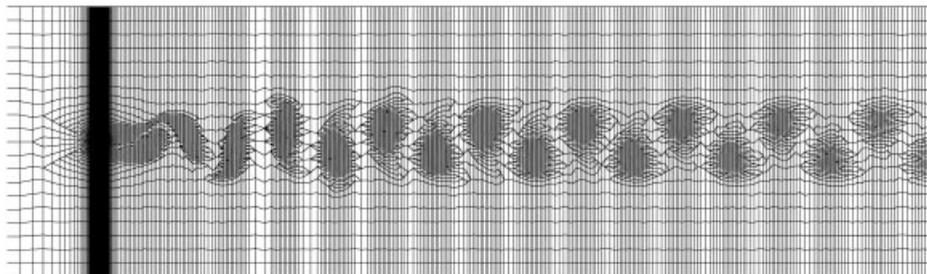


Figure 5. Snapshot of the adaptive grid.

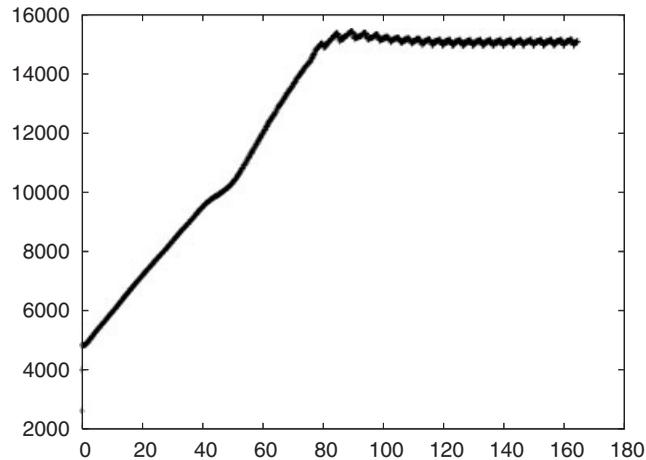


Figure 6. Time history of the number of grid points in the adaptive-grid calculation.

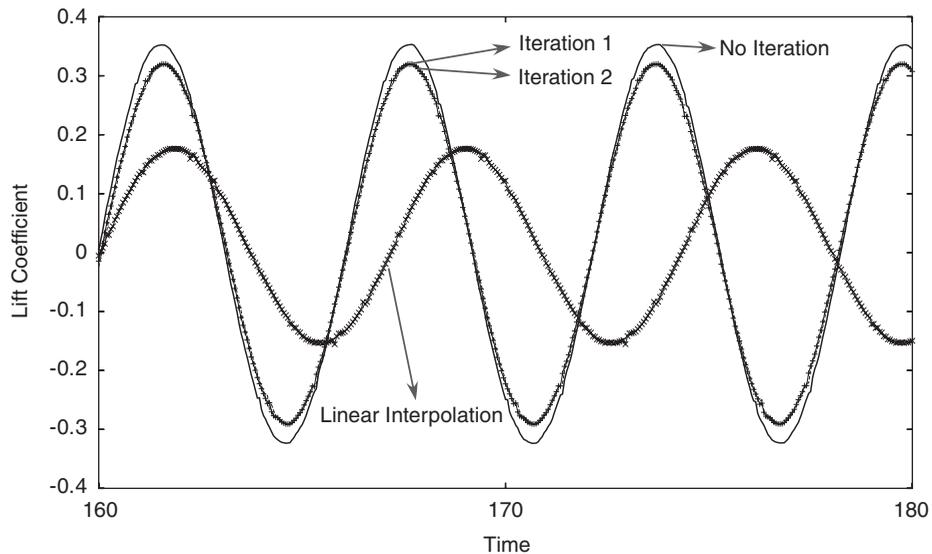


Figure 7. Time history of lift coefficient.

the CIP interpolation described in Section 3.2. The figure shows that the use of linear interpolation instead of the CIP interpolation gives rise to large dissipation as well as error in Strouhal number, which is 0.14 for linear interpolation, while it is 0.165 for the CIP interpolation. Thus, the accurate interpolation is a key issue in the Soroban grid system.

The semi-Lagrangian technique used for the advection terms and the implicit procedure used for the pressure and viscous terms make the overall technique free from CFL constraints, and

therefore we can use large time-step sizes. This becomes very important for locally refined grids, because such refinements place limits on the time-step sizes that can be used in conventional schemes. In our computations, the time-step size was set to  $\Delta t = 0.05$  regardless of grid size and the maximum CFL number was about 10.

## 5. CONCLUDING REMARKS

We provided an overview of the key aspects of a CIP-based adaptive grid approach for computing fluid–solid and fluid–fluid interfaces. In this approach, the CCUP technique, which is based on the CIP method, is combined with the adaptive Soroban grid technique. The approach has a number of superior features. Even though the grid system is unstructured, it still has a simple data structure, and that gives us very good computational efficiency. A single interpolation process is used for both calculating the advection terms and estimating the values at the new grid points from the old grid points, and therefore there is no additional computational overhead associated with the grid motion. Because the Soroban grid technique does not have any elements or cells connecting the grid points, the approach is free from mesh (or grid) distortion limitations. The CIP method represents the solution between the grid points very accurately, and therefore despite the unstructured nature of the grid, the technique has high-order accuracy (the importance of the interpolation procedure was demonstrated by comparing the linear and CIP interpolations). Although the solution technique is based on a collocated-grid approach, high-order accuracy and robustness are maintained.

## REFERENCES

1. Tezduyar TE. Stabilized finite element formulations for incompressible flow computations. *Advances in Applied Mechanics* 1992; **28**:1–44.
2. Tezduyar T, Aliabadi S, Behr M, Johnson A, Mittal S. Parallel finite-element computation of 3D flows. *Computer* 1993; **26**:27–36.
3. Tezduyar TE, Aliabadi SK, Behr M, Mittal S. Massively parallel finite element simulation of compressible and incompressible flows. *Computer Methods in Applied Mechanics and Engineering* 1994; **119**:157–177.
4. Mittal S, Tezduyar TE. Massively parallel finite element computation of incompressible flows involving fluid–body interactions. *Computer Methods in Applied Mechanics and Engineering* 1994; **112**:253–282.
5. Ohayon R. Reduced symmetric models for modal analysis of internal structural-acoustic and hydroelastic-sloshing systems. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**:3009–3019.
6. Tezduyar TE. Computation of moving boundaries and interfaces and stabilization parameters. *International Journal for Numerical Methods in Fluids* 2003; **43**:555–575.
7. Michler C, van Brummelen EH, Hulshoff SJ, de Borst R. The relevance of conservation for stability and accuracy of numerical methods for fluid–structure interaction. *Computer Methods in Applied Mechanics and Engineering* 2003; **192**:4195–4215.
8. van Brummelen EH, de Borst R. On the nonnormality of subiteration for a fluid–structure interaction problem. *SIAM Journal on Scientific Computing* 2005; **27**:599–621.
9. Tezduyar TE, Sathe S, Keedy R, Stein K. Space–time finite element techniques for computation of fluid–structure interactions. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:2002–2027.
10. Bazilevs Y, Calo VM, Zhang Y, Hughes TJR. Isogeometric fluid–structure interaction analysis with applications to arterial blood flow. *Computational Mechanics* 2006; **38**:310–322.
11. Khurram RA, Masud A. A multiscale/stabilized formulation of the incompressible Navier–Stokes equations for moving boundary flows and fluid–structure interaction. *Computational Mechanics* 2006; **38**:403–416.
12. Masud A. Effects of mesh motion on the stability and convergence of ALE based formulations for moving boundary flows. *Computational Mechanics* 2006; **38**:430–439.
13. Kuttler U, Forster C, Wall WA. A solution for the incompressibility dilemma in partitioned fluid–structure interaction with pure Dirichlet fluid domains. *Computational Mechanics* 2006; **38**:417–429.

14. Dettmer W, Peric D. A computational framework for fluid–rigid body interaction: finite element formulation and applications. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:1633–1666.
15. Masud A, Bhanabhagvanwala M, Khurram RA. An adaptive mesh rezoning scheme for moving boundary flows and fluid–structure interaction. *Computers and Fluids* 2007; **36**:77–91.
16. Sawada T, Hisada T. Fluid–structure interaction analysis of the two dimensional flag-in-wind problem by an interface tracking ALE finite element method. *Computers and Fluids* 2007; **36**:136–146.
17. Wall WA, Genkinger S, Ramm E. A strong coupling partitioned approach for fluid–structure interaction with free surfaces. *Computers and Fluids* 2007; **36**:169–183.
18. Tezduyar TE. Finite element methods for flow problems with moving boundaries and interfaces. *Archives of Computational Methods in Engineering* 2001; **8**:83–130.
19. Takizawa K, Yabe T, Tsugawa Y, Tezduyar TE, Mizoe H. Computation of free-surface flows and fluid–object interactions with the CIP method based on adaptive meshless Soroban grids. *Computational Mechanics* 2006; published online.
20. Yabe T, Wang PY. Unified numerical procedure for compressible and incompressible fluid. *Journal of Physical Society of Japan* 1991; **60**:2105–2108.
21. Takewaki H, Nishiguchi A, Yabe T. The cubic-interpolated pseudo-particle (CIP) method for solving hyperbolic-type equations. *Journal of Computational Physics* 1985; **61**:261–268.
22. Yabe T, Aoki T. A universal solver for hyperbolic-equations by cubic-polynomial interpolation I. One-dimensional solver. *Computer Physics Communications* 1991; **66**:219–232.
23. Yabe T, Ishikawa T, Wang PY, Aoki T, Kadota Y, Ikeda F. A universal solver for hyperbolic-equations by cubic-polynomial interpolation II. Two- and three-dimensional solvers. *Computer Physics Communications* 1991; **66**:233–242.
24. Yabe T, Xiao F, Utsumi T. Constrained interpolation profile method for multiphase analysis. *Journal of Computational Physics* 2001; **169**:556–593.
25. Yabe T, Mizoe H, Takizawa K, Moriki H, Im H, Ogata Y. Higher-order schemes with CIP method and adaptive Soroban grid towards mesh-free scheme. *Journal of Computational Physics* 2004; **194**:57–77.
26. Takewaki H, Yabe T. Cubic-interpolated pseudo particle (CIP) method—application to nonlinear or multi-dimensional problems. *Journal of Computational Physics* 1987; **70**:355–372.
27. Aoki T. Multi-dimensional advection of CIP (cubic-interpolate propagation) scheme. *CFD Journal* 1995; **4**: 279–291.
28. Tezduyar TE, Liou J, Ganjoo DK. Incompressible flow computations based on the vorticity–stream function and velocity–pressure formulations. *Computers and Structures* 1990; **35**:445–472.