INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2006; **52**:263–295
Published online 10 February 2006 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1177

Finite volume method with zonal-embedded grids for cylindrical coordinates

Yong Kweon Suh[*,†] and Chang Ho Yeo[‡]

Division of Mechanical Engineering, Dong-A University, 840 Hadan-dong, Saha-gu, Busan 604-714, Republic of Korea

SUMMARY

A zonal-embedded-grid technique has been developed for computation of the two-dimensional Navier–Stokes equations with cylindrical coordinates. As is well known, the conventional regular grid system gives very small grid spacings in the azimuthal direction so it requires a very small time step for a stable numerical solution when the explicit method is used. The fundamental idea of the zonal-embedded-grid technique is that the number of azimuthal grids can be made small near the origin of the coordinates so that the grid size is more uniformly distributed over the domain than with the conventional regular-grid system. The code developed using this technique combined with the explicit, finite-volume method was then applied to calculation of the asymmetric swirl flows and Lamb's multi-polar vortex flows within a full circle and the spin-up flows within a semi-circle. It was shown that the zonal-embedded grids allow a time step far larger than the conventional regular grids. For the case of the Lamb's multi-polar vortex flows, the code was validated by comparing the calculated results with the exact solutions. For the case of the semi-circle spin-up flows, the experimental results were used for the verification. It was seen that the numerical results were in good agreement with the experimental results both qualitatively and quantitatively. Copyright © 2006 John Wiley & Sons, Ltd.

KEY WORDS: zonal-embedded grid; cylindrical coordinates; explicit method; Lamb multi-pole flows; spin-up flows; semi-circular boundary

1. INTRODUCTION

Problems of fluid flow bounded by circular geometry are ubiquitous in engineering applications. For instance, a circular pipe becomes a first choice when we consider conveying gas or liquid from one place to another, and it is more natural to use a circular container, not a rectangular box, in designing a mixer in a chemical process.

[*] Correspondence to: Y. K. Suh, Division of Mechanical Engineering, Dong-A University, 840 Hadan-dong, Saha-gu, Busan 604-714, Republic of Korea.
[†] E-mail: yksuh@dau.ac.kr
[‡] E-mail: chyeo@hyosung.com

Contract/grant sponsor: Korea Research Foundation; contract/grant number: KRF-2002-D00080
Contract/grant sponsor: Korea Science and Engineering Foundation

Received 21 September 2004
Revised 22 November 2005
Accepted 14 December 2005

In such geometry, use of the cylindrical coordinate system, if possible, is the most relevant and desirable in the formulation and discretization of the Navier–Stokes equations in numerical simulations. Then, as a specific tool one may suggest employing a spectral method of some kind because the problem is periodic in the azimuthal direction, e.g. References [1–7]. However, the spectral method has a serious drawback associated with the Gibbs phenomenon for the case in which the first or second derivative of the flow field tends to be spatially discontinuous, which is not exceptional for high-Reynolds number flows.

Use of finite difference or finite volume method with the cylindrical coordinates [8, 9] also reveals a fundamental problem when the regular grid system is employed; in this paper, the 'regular grid system' denotes the grid system which is built by simply drawing both radial lines and concentric circles as shown in Figure 1(a). As is well known, under the regular grid system extremely small time steps must be taken for the sake of numerical stability, when an explicit method is employed, because the grid spacing in the azimuthal direction is proportional to the radial distance from the origin, and therefore the grid spacing at the centre is usually some ten or hundred times smaller than that at the outer boundary. One remedy to overcome such restriction in the time step is to use an implicit method at least in the azimuthal direction as suggested by Akselvoll and Moin [10]. The implicit (or semi-implicit) method is indeed preferable in that a larger time step can be taken without causing the numerical instability. On the other hand, however, the recent development of the parallel-computation technology makes the computational fluid dynamicists revisit the explicit method, because it is intrinsically more suitable for the parallel computation especially for the domain-decomposition method [11–15]. In addition, the programming is simple and flexible with the explicit method.

There are lots of computational techniques that can resolve some of the fundamental problems encountered in the cylindrical geometry, e.g. the spectral element method proposed by Patera [16] and employed by others [17–20]. However, as usual, most of the numerical techniques cannot resolve the problems completely, and each has its own merit focusing on resolving any one or some part of the problems.

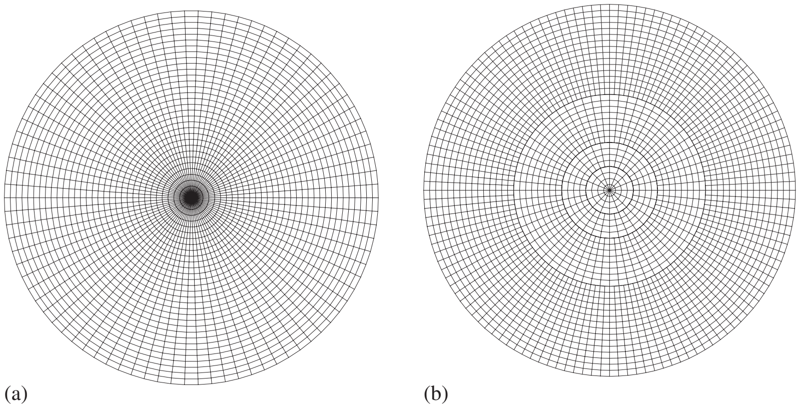

Figure 1. Regular and zonal-embedded grids for the domain within a circle. $I \times J = 32 \times 86$ for the regular grids and $M = 4$, $I_0 = 4$, $J_0 = 16$ ($I \times \bar{J} = 32 \times 86$) for the zonal-embedded grids: (a) regular grid; and (b) zonal-embedded grid.

We propose in this study to use the zonal-embedded-grid system shown in Figure 1(b) rather than the simple regular grid system, Figure 1(a), for the computation of fluid flow using an explicit method within a domain surrounded by a wall of circular geometry. With zonal-embedded grids, we can control the azimuthal grid spacing independent of the radial distance from the origin, and thus the time step can be made significantly larger than that with the regular grids. In particular, it may be useful when an increased resolution in the azimuthal direction is required near the wall of a circular pipe in order to catch small vortical structures of turbulent flows at high Reynolds numbers. In this case, the azimuthal grid spacing near the wall can be set arbitrarily small while that near the pipe centre is kept still large so that the critical time step occurs not at the centre but near the wall itself, which is a clear indication of flexibility in the grid design.

Use of zonal-embedded grids in computational fluid dynamics is not new. Kravchenko *et al.* [21] introduced zonal-embedded grids in a numerical study on the turbulent flow above a flat plate. However, no studies have been found in the literature that employed the zonal-embedded-grid system in solving a fluid flow problem with the cylindrical coordinates. On the other hand, the method of dropping the number of modes as the origin is approached with a Fourier spectral method can be considered as an alternative way to overcome the numerical instability near the origin, as pointed out by a referee.

In Section 2 the governing equations for two-dimensional incompressible flows are introduced specifically for asymmetric swirl flows and Lamb's multi-polar vortical flows within a full circular and spin-up flows within a semi-circular boundary. The method of building a zonal-embedded-grid system together with the overall numerical methods is then described in Section 3. We present the numerical results in Section 4, and finally, Section 5 will summarize the conclusions.

2. FORMULATION OF THE GOVERNING EQUATIONS

We consider an incompressible fluid of density ρ and kinematic viscosity v within a circular or a semi-circular container of radius R. With a suitable choice of the reference length, time, velocity and pressure, which is dependent on the particular model flow, we can write the governing equations, in terms of the cylindrical coordinates, in a dimensionless form as follows:

$$\frac{\partial ru}{r\partial r}+\frac{\partial v}{r\partial\theta}=w \tag{1}$$

$$\frac{\partial u}{\partial t}+\frac{\partial ru^2}{r\partial r}+\frac{\partial uv}{r\partial\theta}-\frac{v^2}{r}=-\frac{\partial p}{\partial r}+\frac{1}{Re}\left[\frac{\partial}{r\partial r}\left(\frac{r\partial u}{\partial r}\right)+\frac{\partial^2 u}{r^2\partial\theta^2}-\frac{u}{r^2}-\frac{2}{r^2}\frac{\partial v}{\partial\theta}\right]+f \tag{2a}$$

$$\frac{\partial v}{\partial t}+\frac{\partial ruv}{r\partial r}+\frac{\partial v^2}{r\partial\theta}+\frac{uv}{r}=-\frac{\partial p}{r\partial\theta}+\frac{1}{Re}\left[\frac{\partial}{r\partial r}\left(\frac{r\partial v}{\partial r}\right)+\frac{\partial^2 v}{r^2\partial\theta^2}-\frac{v}{r^2}+\frac{2}{r^2}\frac{\partial u}{\partial\theta}\right]+g \tag{2b}$$

where all the variables are dimensionless; t is the time, r and θ the radial and azimuthal coordinates, respectively, u and v the velocity components along each direction, and p the pressure. The mass-source term w and the body-force terms f and g on the right-hand sides of the above equations take different forms depending on the flow models. Table I shows the specific forms of these terms for three kinds of flow models treated in this study. This table

Table I. The three model flows treated in this study, and the corresponding reference length, velocity and time used in deriving the dimensionless governing equations.

Flow models	Asymm. swirl flow, decaying in time	Lamb's multi-pole flows periodic in time	Spin-up flows decaying in time
Ref. length	R	R	R
Ref. velocity	$u^*_{\max}$	$u^*_{\max}$	$R\Delta\Omega$
Ref. time	$R/u^*_{\max}$	$R/u^*_{\max}$	$1/\Delta\Omega$
Re	$Ru^*_{\max}/v$	$Ru^*_{\max}/v$	$R^2\Delta\Omega/v$
w	0	0	W_{E}/h
f	0	$nB(t)[J_n(ar)/r]\cos n\theta$	$2(q(t)+1/\varepsilon)v+uW_{\mathrm{E}}/h$
g	0	$-aB(t)J'_n(ar)\sin n\theta$	$-2(q(t)+1/\varepsilon)u+vW_{\mathrm{E}}/h-rq'(t)$
B.C.	$u=v=0$ at $r=1$	$u=\partial(rv)/\partial r=0$ at $r=1$	$u=v=0$ at $r=1$ $u=v=0$ at $\theta=0,\ \pi$
I.C.	Equation (3)	$u=nA(0)[J_n(ar)/r]\cos n\theta$ $v=-aA(0)J'_n(ar)\sin n\theta$	$u=v=0$

Re, the Reynolds number, and w, f and g, appearing on the right-hand side of Equations (1), (2a) and (2b). B.C. and I.C. indicate the boundary and initial conditions, respectively. $u^*_{\max}$ denotes the maximum physical velocity observed over the whole spatial and temporal domain. Refer to text for the other nomenclature.

also presents the reference quantities used in the derivation of the dimensionless equations, the Reynolds numbers, boundary conditions and initial conditions relevant for each flow model.

The initial conditions for the first model flow, i.e. the swirl flow, are characterized by a radially parabolic distribution of the azimuthal velocity component up to $r=\rho_c$ and a zero radial velocity component as follows:

$$u=0, \quad v=\begin{cases} \dfrac{r(\rho_c-r)}{(\rho_c/2)^2} & \text{for } 0\leqslant r\leqslant\rho_c \\ 0 & \text{for } r>\rho_c \end{cases} \tag{3}$$

This swirl flow is axisymmetric and the variables are independent of the azimuthal coordinate θ. For a more general case, we break this symmetry by parallely moving this basis flow as much as r_c along the line $\theta=\pi/2$ so that the swirl's centre is now at the point $(r,\theta)=(r_c,\pi/2)$.

The Lamb's multi-pole flows are periodic in time due to time-periodic body-force terms. In Table I, $J_n(ar)$ stands for the Bessel function of the first kind of order n and a satisfies $J_n(a)=0$. The functions $B(t)$ and $A(t)$ are

$$B(t)=\frac{-2\pi r_c\sqrt{1+(a^2/2\pi Re)^2}}{nJ_n(ar_c)}\sin 2\pi t \tag{4a}$$

$$A(t)=\frac{2/a}{\sqrt{1+(a^2/2\pi Re)^2}}\left(\cos 2\pi t-\frac{a^2}{2\pi Re}\sin 2\pi t\right) \tag{4b}$$

where r_c satisfies $ar_cJ'_n(ar_c)-J_n(ar_c)=0$. This restriction corresponds to the requirement that u becomes 1 at the point given by $r=r_c$ and $\theta=0$ where $|u|$ becomes maximum. The exact solutions for the Lamb's flows are shown to be

$$u = \frac{nA(t)J_n(ar)}{r} \cos n\theta \tag{5a}$$

$$v = -aA(t)J_n'(ar) \sin n\theta \tag{5b}$$

Thus the vortices are stationary in space but their strength is changing periodically in time. The number of vortices (or poles) is $2n$, and the Lamb dipole corresponds to $n = 1$.

The third model flows are spin-up flows within a semi-circular container. The container is initially rotating with a constant angular velocity $\Omega - \Delta\Omega$ for a long time so that the fluid remains in a state of solid-body rotation. Then, at a certain instant of time the container's angular speed is increased abruptly but smoothly by as much as $\Delta\Omega$ and finally maintained at a constant value Ω. During this speed-up period, the so-called spin-up flow starts to develop inside the container. It is well known that when the ratio $\Delta\Omega/\Omega$, called the Rossby number and denoted as ε in this paper, is small enough the axial flow (flow along the axis of rotation) is inhibited and the flow tends to be basically two dimensional [22]. The Ekman-pumping velocity W_{E} shown in Table I is given by the linear Ekman-pumping law [23]

$$W_{\mathrm{E}} = \frac{\sqrt{\varepsilon/Re}}{2}\varsigma \tag{6}$$

where the vorticity ς is determined by

$$\varsigma = \frac{1}{r}\left(\frac{\partial rv}{\partial r} - \frac{\partial u}{\partial \theta}\right) \tag{7}$$

and h in the table is the dimensionless liquid depth in a quiescent state. This pumping velocity brings the non-zero mass source w in (1), which tends to be uniformly distributed over the liquid column at low Rossby numbers, in such a way that the two-dimensional assumption be valid. The function $q(t)$ controls the speed-up mode, and for a smooth operation we employ the harmonic function:

$$q = \begin{cases} -(1 + \cos \omega t)/2 & \text{for } 0 \leqslant t \leqslant t_{\mathrm{s}} \\ 0 & \text{for } t \geqslant t_{\mathrm{s}} \end{cases} \tag{8}$$

The speed-up time t_{s} is related to ω by $t_{\mathrm{s}} = \pi/\omega$.

3. IMPLEMENTATION OF ZONAL-EMBEDDED GRIDS AND NUMERICAL METHODS

3.1. Zonal-embedded grid

To construct a zonal-embedded-grid system, the entire domain is first divided into several sub-domains or zones by concentric circles. Figure 1(b) shows the case with four zones. In each zone, the grids are built in the same way as the regular grids are made. However, the number of grids in the azimuthal direction differs from one zone to another, and following the purpose of this study, we increase the number of azimuthal grids in the outer zone. The simplest way to do this is to double the number of grids each time one crosses the zones radially outward. Figure 1(b) is a typical example in that it starts from 16 grid lines in the innermost zone and 32 lines in the next, progressing to 128 lines in the outermost zone. To keep a constant azimuthal grid-spacing at the outer end position of each zone, we need to double the radial size of each zone as one crosses the zones outward, except for the second zone as illustrated below.

Suppose we have M zones. Then I_m, the number of radial grids, and J_m, the number of azimuthal grids, at the mth zone are given by

$$I_m = \begin{cases} I_0 & \text{for } m=1 \\ 2^{m-2} I_0 & \text{for } 2 \leqslant m \leqslant M \end{cases} \tag{9a}$$

$$J_m = 2^{m-1} J_0 \quad (m = 1, 2, \ldots, M) \tag{9b}$$

where I_0 and J_0 are the number of grids in the first zone. The reason of taking the same I_m for the first and second zones is related to the implementation of the extrapolation scheme $\partial^3 \phi / \partial r^3 = 0$ in evaluation of a flow variable ϕ near the centre point. In order to use only the first zone's variables in this extrapolation, so that the algorithm becomes simple, at least four radial grids must be established in the first zone, $I_0 \geqslant 4$. For instance, with $M = 4$, $I_0 = 4$, if we use the formula $I_m = 2^{m-1} I_0$ $(1 \leqslant m \leqslant M)$ instead of (9a), then $I_m = 4$, 8, 16, 32 and the total number of radial grids becomes $I = 60$. On the other hand, if we use (9a), then $I_m = 4$, 4, 8, 16 and we have $I = 32$. Thus Equation (9a) allows for implementation of a simple extrapolation scheme near the centre point at a lower I. Parameters I_0 and J_0 together with M control the overall grid spacings. The averaged number of the azimuthal grids, $\bar{J}$, is given by

$$\bar{J} = \frac{1}{I} \sum_{m=1}^{M} I_m J_m$$

Figure 1(b) is made with $M = 4$, $I_0 = 4$ and $J_0 = 16$, which gives the total number of radial grids within the entire domain, $I = 32$, the number of azimuthal grids in the outermost zone, $J = 128$, and the averaged number of azimuthal grids, $\bar{J} = 86$.

3.2. Discretization of the continuity and momentum equations

Associated with the allocation of pressure and velocity components in each cell, we employ the staggered-grid method; the pressure is defined at the centre of the p-cell, and the velocity component at the mid-point of each line normal to the component surrounding the p-cell. There are two kinds of p-cells; one is a regular cell composed of four edges (cells marked 'O', 'B', and 'C' in Figure 2) and the other is a special cell composed of five edges (marked 'A' in Figure 2).

The continuity equation (1) is first integrated over the regular p-cell shown in Figure 3(a). The result is

$$u_e \Delta s_e - u_w \Delta s_w + v_n \Delta r - v_s \Delta r = \Delta s_0 \Delta r \, w_0^o \tag{10}$$

where Δr and Δs $(= r\Delta\theta)$ are the radial and azimuthal grid spacings, respectively, and w_0^o represents the value of w at the cell's centre point '0' given at the previous time step. The superscript 'o' indicates that the variable is known and evaluated at the previous time step. Notice that, for the spin-up flows, the right-hand side also contains the unknown velocity

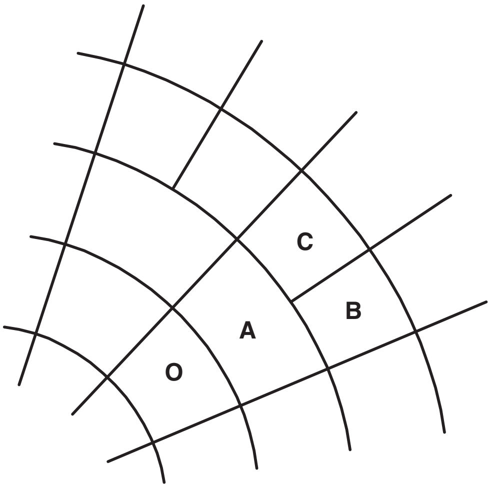

Figure 2. Four types of cells in the zonal-embedded-grid system.

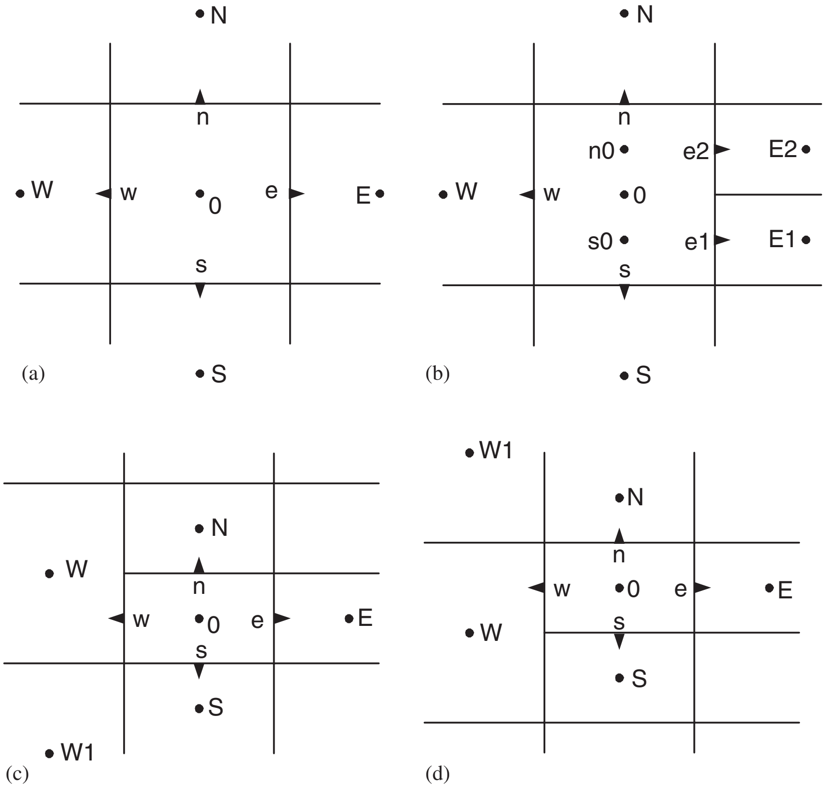

Figure 3. Four types of p-cells with various symbols to be used in discretization of the continuity equation and derivation of the discretized pressure equations: (a) regular p-cell of 'O' type; (b) special p-cell of 'A' type; (c) special p-cell of 'B' type; and (d) special p-cell of 'C' type.

components u and v (refer to Table I). Treating these variables really unknown, like the other velocity components on the left-hand side of (10), inevitably results in a much complex pressure equation. Therefore we treat these terms explicitly. The underlying principle for the relevance of treating the right-hand side of (10) known comes from the fact that at high Reynolds numbers the magnitude of w is in general much smaller than that of u or v.

The momentum equations (2a) and (2b) are then integrated over the regular u- and v-cells, respectively, to give

$$u_e = u_e^o + \Delta\tau\left(F_e - \frac{p_E - p_0}{\Delta r}\right), \quad u_w = u_w^o + \Delta\tau\left(F_w - \frac{p_0 - p_W}{\Delta r}\right) \tag{11a}$$

$$v_n = v_n^o + \Delta\tau\left(G_n - \frac{p_N - p_0}{\Delta s_0}\right), \quad v_s = v_s^o + \Delta\tau\left(G_s - \frac{p_0 - p_S}{\Delta s_0}\right) \tag{11b}$$

where F and G denote the momentum fluxes containing the convection, diffusion and source terms. Formulation for these quantities will be given in the next subsection. The time step $\Delta\tau$ takes Δt or $\Delta t/2$ depending on the stage number within one step of the fourth-order Runge–Kutta method (to be addressed in Section 3.6). The subscripts in (11a) and (11b) are still based on the p-cell. Substituting these results into the discretized continuity equation (10) gives us the pressure equation reading:

$$\begin{gathered} a_w p_W + a_s p_S + a_0 p_0 + a_e p_\mathrm{E} + a_n p_N = b \\ a_w = \Delta s_w/\Delta r^2, \quad a_e = \Delta s_e/\Delta r^2, \quad a_s = 1/\Delta s_0, \quad a_n = 1/\Delta s_0 \\ a_0 = -(a_w + a_e + a_s + a_n) \\ b = (F_e\Delta s_e - F_w\Delta s_w)/\Delta r + G_n - G_s - \Delta s_0(w_0^o - w_0^{oo})/\Delta t \end{gathered} \tag{12}$$

where w_0^{oo} stands for the value of w_0 at the previous-previous time step. Here too, the term $(w_0 - w_0^o)/\Delta t$ on the right-hand side of the equation for b has been replaced by $(w_0^o - w_0^{oo})/\Delta t$ to avoid complexity; in fact the former corresponds to a discretized form of $\partial w/\partial t$. An easy way of implementing the impermeable condition at the walls is to set both the corresponding coefficient a and flux F or G zero at the edge of the cell contacting each wall.

For the special p-cell of 'A' type shown in Figure 3(b), the continuity equation takes the form

$$(u_{e1} + u_{e2})\Delta s_e - u_w\Delta s_w + v_n\Delta r - v_s\Delta r = \Delta s_0\Delta r\, w_0^o \tag{13}$$

where the formula for u_{e1} and u_{e2} now contain p_S and p_N;

$$u_{e1} = u_{e1}^o + \Delta\tau\left(F_{e1} - \frac{p_{E1} - p_0}{\Delta r}\right) - \Delta\tau\frac{p_0 - p_S}{4\Delta r} \tag{14a}$$

$$u_{e2} = u_{e2}^o + \Delta\tau\left(F_{e2} - \frac{p_{E2} - p_0}{\Delta r}\right) - \Delta\tau\frac{p_0 - p_N}{4\Delta r} \tag{14b}$$

The last term on the right-hand side of each of these equations comes from discretization of $\partial p/\partial r$ at the point '$e1$' or '$e2$' while keeping the second-order accuracy. For instance at the point '$e1$' we have $\partial p/\partial r = (p_{E1} - p_{s0})/\Delta r$ and a linear interpolation gives $p_{s0} = (3p_0 + p_S)/4$, which results in $\partial p/\partial r = (p_{E1} - p_0)/\Delta r + (p_0 - p_S)/4\Delta r$. The first term on the right-hand side of this equation is included in the bracket on the right-hand side of (14a). It can be shown that the matrix of the discretized pressure equation system to be given by substituting (14a) and (14b) into the continuity equation keeps the symmetry property. Now, substituting (14a) and (14b) together with the formula for u_w, v_n and v_s into (11) yields

$$a_w p_W + a_s p_S + a_0 p_0 + a_{e1} p_{E1} + a_{e2} p_{E2} + a_n p_N = b \tag{15}$$

where the formula for the coefficients a_s, a_n, a_0 and b are

$$a_s = a_n = 1/\Delta s_0 - \Delta s_e/(4\Delta r^2)$$

$$a_0 = -(a_w + a_{e1} + a_{e2} + a_s + a_n)$$

$$b = [(F_{e1} + F_{e2})\Delta s_e - F_w \Delta s_w]/\Delta r + G_n - G_s - \Delta s_0(w_0^o - w_0^{oo})/\Delta t$$

and those for the other coefficients are the same as those shown under (12).

Special attention must be given to the cells adjacent to the walls. The coefficient a at the edge contacting the wall must be set at zero. It is also required to set F at zero at the edge contacting the wall at $r = 1$ or at the origin $r = 0$. However, when the cell is contacting the flat wall at $\theta = 0$ or at $\theta = \pi$ for the case of the semi-circular problem, the variable G should not be set at zero, and further the variable b takes the form

$$b = [(F_{e1} + F_{e2})\Delta s_e - F_w \Delta s_w]/\Delta r + G_n - G_{sw}\Delta s_e \Delta s_0/(4\Delta r^2) - \Delta s_0(w_0^o - w_0^{oo})/\Delta t$$

for the case where the cell faces the wall at the 's' edge ($\theta = 0$), and

$$b = [(F_{e1} + F_{e2})\Delta s_e - F_w \Delta s_w]/\Delta r + G_{nw}\Delta s_e \Delta s_0/(4\Delta r^2) - G_s - \Delta s_0(w_0^o - w_0^{oo})/\Delta t$$

for the case where the cell faces the wall at the 'n' edge ($\theta = \pi$). Here, G_{sw} and G_{nw} must be evaluated from the corresponding momentum equations at the facing walls. For instance, the formula for G_{sw} is as follows:

$$G_{sw} = -r_0 q'(t) + \frac{1}{Re}\left(\frac{1}{r_0^2}\frac{\delta^2 v}{\delta\theta^2} + \frac{2}{r_0^2}\frac{\delta u}{\delta\theta}\right)_s \tag{16}$$

The symbol 'δ' stands for the discretization of a derivative; in this study the simplest formulae are used for this, e.g.

$$\frac{\delta^2 v}{\delta\theta^2} = \frac{v_3 - 2v_2}{\Delta\theta^2} \tag{17a}$$

$$\frac{\delta u}{\delta\theta} = \frac{2u_{02}}{\Delta\theta} \tag{17b}$$

where v_2 and v_3 denote the v-value at the centre of each edge apart from the wall as much as $\Delta\theta$ and $2\Delta\theta$, respectively, and u_{02} is the u-value at the centre of the p-cell contacting the wall (Figure 4). Instead of using the one-sided difference formula (17a), we can employ the centred difference formula

$$\frac{\delta^2 v}{\delta\theta^2} = \frac{v_2 - 2v_1 + v_0}{\Delta\theta^2} = 0 \tag{18}$$

where the second equality comes from the impermeable condition $v_1 = 0$ and the condition $v_0 = -v_2$ (or $\delta v/\delta\theta = 0$) at the walls.

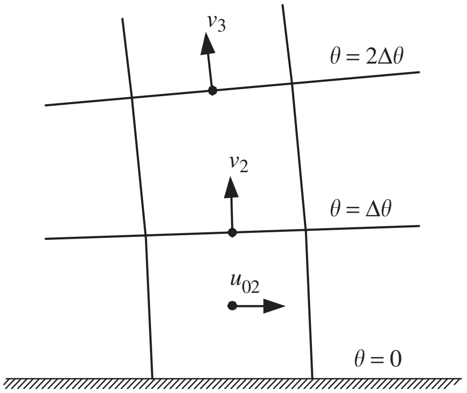

Figure 4. Definition of local variables for use in discretization of $\delta^2 v/\delta\theta^2$ and $\delta u/\delta\theta$, Equation (17), for the cell contacting the wall at the 's' edge.

For the p-cell of 'B' type shown in Figure 3(c), the discretized continuity equation takes the same form as (10). The formula for u_e, v_n and v_s are also the same as the corresponding ones in (11a) and (11b). The formula for u_w is

$$u_w = u_w^o + \Delta\tau\left(F_w - \frac{p_0 - p_W}{\Delta r}\right) - \Delta\tau\frac{p_W - p_{W1}}{4\Delta r} \tag{19}$$

Substituting these into (10), we get the result of (12) and all the coefficients are also the same as the corresponding ones shown under (12) except for b which in this case reads

$$b = (F_e\Delta s_e - F_w\Delta s_w)/\Delta r + G_n - G_s - \Delta s_0(w_0^o - w_0^{oo})/\Delta t + \Delta s_w(p_W^o - p_{W1}^o)/(4\Delta r^2) \tag{20}$$

Here, the last term is composed of the known pressures evaluated at the previous time step. These pressures are treated as known instead of unknown in order to make the coefficient matrix of the pressure equation symmetric so that we can use an efficient algorithm based on the conjugate gradient (CG) method; if the coefficient matrix were not symmetric we would need to use a solver such as Bi-Conjugate Gradient Stabilized (Bi-CGSTAB), and then the computational time needed in solving the discretized pressure equation would be almost double the time needed when the CG method is used.

For the 'B'-type cell too, we must pay a special attention to the evaluation of coefficients near the walls. For instance, when the wall contacts the cell at the 's' edge, the coefficient a_s should be taken as zero, and b becomes

$$b = [F_e\Delta s_e - F_w\Delta s_w]/\Delta r + G_n + G_{sww}\Delta s_w\Delta s_0/(4\Delta r^2) - \Delta s_0(w_0^o - w_0^{oo})/\Delta t$$

G_{sww} essentially takes the same form as (16), but in this case this value must be evaluated at the mid-point between W and $W1$.

For the p-cell of 'C' type shown in Figure 3(d), we can follow a very similar procedure as the case for the 'B'-type cell described above.

Special attention must be given to evaluating w_0 in terms of the velocity components u and v. Since w_0 is just proportional to the vorticity ς (refer to Table I and Equation (6)), one may try to obtain this by firstly computing each vorticity at the grid point where the solid lines intersect in Figure 3, and secondly averaging them to get the value at the point '0'. However this method does not guarantee the conservation property. For the continuity equation to keep the mass conservation, the vorticity (and therefore w_0) must be evaluated directly at the point '0'.

3.3. Discretization of the momentum flux

The momentum flux F and G appearing in the pressure equations can be obtained by the following discretized formula:

$$F = \frac{-F^c + F^d}{r_0} \tag{21a}$$

$$F^c = \frac{1}{\Delta r}(r_e u_e^2 - r_w u_w^2) + \left(\frac{\delta uv}{\delta\theta}\right)_0 - v_0^2 - r_0 f_0 \tag{21b}$$

$$F^d = \frac{1}{Re}\left[\frac{r_e(\delta u/\delta r)_e - r_w(\delta u/\delta r)_w}{\Delta r} + \frac{(\delta u/\delta\theta)_n - (\delta u/\delta\theta)_s}{\Delta s_0} - \frac{u_0}{r_0} - \frac{2}{r_0}\left(\frac{\delta v}{\delta\theta}\right)_0\right] \tag{21c}$$

$$G = \frac{-G^c + G^d}{r_0} \tag{21d}$$

$$G^c = \frac{1}{\Delta r}(r_e u_e v_e - r_w u_w v_w) + \left(\frac{\delta v^2}{\delta\theta}\right)_0 + u_0 v_0 - r_0 g_0 \tag{21e}$$

$$G^d = \frac{1}{Re}\left[\frac{r_e(\delta v/\delta r)_e - r_w(\delta v/\delta r)_w}{\Delta r} + \frac{(\delta v/\delta\theta)_n - (\delta v/\delta\theta)_s}{\Delta s_0} - \frac{v_0}{r_0} + \frac{2}{r_0}\left(\frac{\delta u}{\delta\theta}\right)_0\right] \tag{21f}$$

Here, two kinds of algorithms for the azimuthal derivatives are used. The first kind is the second-order algorithm

$$\left(\frac{\delta\phi}{\delta\theta}\right)_0 = \frac{\phi_+ - \phi_-}{\Delta\theta} \tag{22a}$$

and the second is the fourth-order algorithm

$$\left(\frac{\delta\phi}{\delta\theta}\right)_0 = \frac{27(\phi_+ - \phi_-) + \phi_{++} - \phi_{--}}{24\Delta\theta} \tag{22b}$$

where ϕ_+, ϕ_-, ϕ_{++} and ϕ_{--} denote the value of the variable ϕ at the points apart from the point '0' as much as $\Delta\theta/2$, $-\Delta\theta/2$, $3\Delta\theta/2$ and $-3\Delta\theta/2$, respectively, in the θ direction. In evaluating the ϕ-values at the points where the variable is not defined, we used two kinds of interpolation algorithms:

$$\phi_0 = \frac{\phi_+ + \phi_-}{2} \tag{23a}$$

as the second-order and

$$\phi_0 = \frac{9(\phi_+ + \phi_-) - (\phi_{++} + \phi_{--})}{16} \tag{23b}$$

as the fourth-order formula. As will be shown in the following section, application of the second-order formula brings a significant error in the numerical results when the flow near the origin of the coordinate system is characterized basically as a uniform flow. For instance, when the flow near the origin is uniform with the speed U and directed along the x direction, its velocity components can be described asymptotically as $u = U \cos\theta$, $v = -U \sin\theta$ and therefore the variation of these variables are significant because $\Delta\theta$ is in general large near the origin. Since making $\Delta\theta$ large near the origin corresponds in practice to the fundamental reason for the introduction of the zonal-embedded-grid method, employing higher-order algorithms such as (22b) and (23b) instead of (22a) and (23a) is very important for successful applications. When the flow is purely radial or azimuthal, we have $\delta\phi/\delta\theta = 0$, and there is no difference between the second- and fourth-order algorithms.

3.4. Treatment at the centre point

The mathematical singularity appearing in the governing equations written in terms of the cylindrical coordinates is in fact removed by adopting the staggered-grid system. However, we have to evaluate u_{1j} defined at the centre point and v_{1j} defined beyond the centre point in order to compute u_{2j} and v_{2j} (refer to Figure 5). The method of Fukagata and Sasaki (F&S hereafter) can be described as follows. First, from the components u_{2j} and v_{2j} obtained at the previous time step, we approximate the Cartesian velocity components $\bar{u}_x$ and $\bar{u}_y$ at the centre point by the formula

$$\bar{u}_x = \frac{1}{J_0 - 1} \sum_{j=2}^{J_0} (u_{2j} \cos\theta_{uj} - v_{2j} \sin\theta_{vj}) \tag{24a}$$

$$\bar{u}_y = \frac{1}{J_0 - 1} \sum_{j=2}^{J_0} (u_{2j} \sin\theta_{uj} + v_{2j} \cos\theta_{vj}) \tag{24b}$$

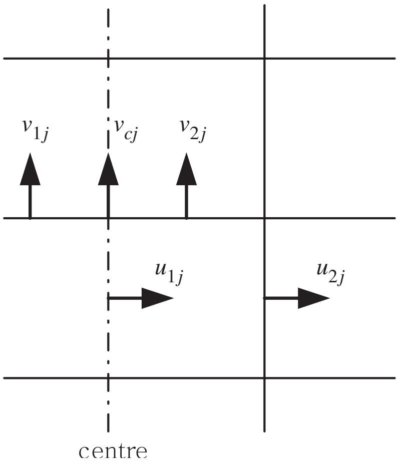

Figure 5. Definition of local velocity components near the centre point (denoted as a dashed line) for use in the centre-point treatment.

where θ_{uj} and θ_{vj} are θ coordinates at the u- and v-grid point, respectively. Then at the centre point we have

$$u_{1j} = \bar{u}_x \cos \theta_{uj} + \bar{u}_y \sin \theta_{uj} \tag{25a}$$

$$v_{cj} = \bar{u}_y \cos \theta_{vj} - \bar{u}_x \sin \theta_{vj} \tag{25b}$$

The variable v_{1j} is next obtained by a simple extrapolation formula

$$v_{1j} = 2v_{cj} - v_{2j} \tag{26}$$

The second treatment we employed in this study is simply using the extrapolation schemes

$$u_{1j} = 3u_{2j} - 3u_{3j} + u_{4j} \tag{27a}$$

$$v_{1j} = 3v_{2j} - 3v_{3j} + v_{4j} \tag{27b}$$

which is equivalent to applying $\partial^3\phi/\partial r^3 = 0$ at the centre point.

3.5. Conservation properties

The finite volume method described so far guarantees the mass conservation in a discrete sense.

Each component of the momentum is also conserved in most cells except for the cells near the interface between the neighbouring zones. For the u-momentum to be conserved at the interfacial u-cell of Figure 6(a), the term u_e^2 on the right-hand side of (21b) and the term $(\delta u/\delta r)_e$ on the right-hand side of (21c) must be replaced by $(u_{e1}^2 + u_{e2}^2)/2$ and $[(\delta u/\delta r)_{e1} + (\delta u/\delta r)_{e2}]/2$, respectively. Similarly, for the v-momentum to be conserved at the interfacial v-cell of Figure 6(b), the term $u_e v_e$ and on the right-hand side of (21e) and the

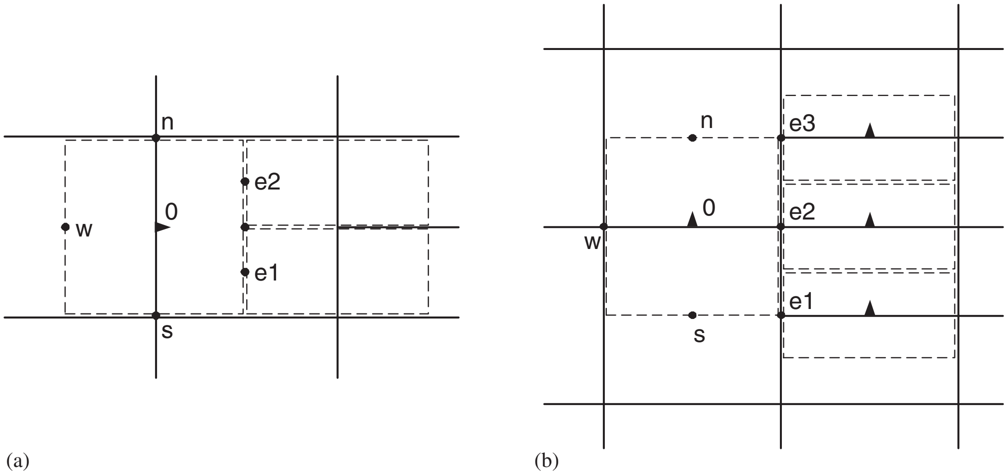

Figure 6. Interfacial u- and v-cells (surrounded by dashed lines around ‘0’) subjected to a special attention for deriving the discretized momentum flux which guarantees the conservation of momentum: (a) interfacial u-cell; and (b) interfacial v-cell.

term $(\delta v/\delta r)_e$ on the right-hand side of (21f) must be replaced by $(u_{e1}v_{e1}+2u_{e2}v_{e2}+u_{e3}v_{e3})/4$ and $[(\delta v/\delta r)_{e1}+2(\delta u/\delta r)_{e2}+(\delta v/\delta r)_{e3}]/4$, respectively.

Conservation of energy is relevant only for the convection terms. The energy-conservative scheme for the cylindrical coordinate system has been proposed by e.g. Fukagata and Sasaki. For instance for a regular u-cell, the term $r_e u_e^2 - r_w u_w^2$ on the right-hand side of (21b) must be replaced by $(\overline{ru})_e\bar{u}_e - (\overline{ru})_w\bar{u}_w$, where the over-bar indicates an arithmetic average like $\bar{\phi}_e = (\phi_0 + \phi_E)/2$. In this way we can construct easily the energy-conservative discrete equations for the regular cells. However it turns out to be very difficult to contrive the energy-conservative algorithm for the interfacial cells.

3.6. Fourth-order Runge–Kutta method and incomplete Cholesky conjugate gradient (ICCG)

After being discretized spatially and combined, the momentum equations (2a) and (2b) take the forms

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{f}(\mathbf{u}, p)$$

where the vector function $\mathbf{f}$ stands for the discretized form of the right-hand side of Equations (2a) and (2b). The algorithm of the fourth-order Runge–Kutta method for this system of equations can be written as

First stage:

$$\mathbf{k}_1 = \Delta t\,\mathbf{f}(\mathbf{u}^o, p^{*1}), \quad \mathbf{u}^{*1} = \mathbf{u}^o + \Delta t\,\mathbf{k}_1/2$$

Second stage:

$$\mathbf{k}_2 = \Delta t\,\mathbf{f}(\mathbf{u}^{*1}, p^{*2}), \quad \mathbf{u}^{*2} = \mathbf{u}^o + \Delta t\,\mathbf{k}_2/2$$

Third stage:

$$\mathbf{k}_3 = \Delta t\,\mathbf{f}(\mathbf{u}^{*2}, p^{*3}), \quad \mathbf{u}^{*3} = \mathbf{u}^o + \Delta t\,\mathbf{k}_3$$

Fourth stage:

$$\mathbf{k}_4 = \Delta t\,\mathbf{f}(\mathbf{u}^{*3}, p^{*4}), \quad \mathbf{u}^{n} = \mathbf{u}^o + (\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4)/6$$

Therefore the time step $\Delta\tau$ used in the formulation of the discretized pressure equations takes $\Delta t/2$ in the first and second stages, and Δt in the third stage. Further, the pressures p^{*1}, p^{*2}, p^{*3} and p^{*4} are determined by solving the pressure equations such that the velocities $\mathbf{u}^{*1}$, $\mathbf{u}^{*2}$, $\mathbf{u}^{*3}$ and $\mathbf{u}^{*4}$ satisfy the continuity equations, respectively.

The pressure equations are solved by employing the ICCG (incomplete Cholesky conjugate gradient) method. As described earlier, in order to keep the benefit of CG against Bi-CGSTAB, the coefficient matrix for the discretized pressure equation was kept symmetrical by taking some interfacial pressures known.

The initial pressure fields needed in the iteration of second and fourth stages were set by using a simple extrapolation scheme

$$p^{*2} = 2p^{*1} - p^{0*3}, \quad p^{*4} = 2p^{*3} - p^{*1} \tag{28}$$

where p^{0*3} denotes p^{*3} in the previous time level. It turns out that such extrapolation speeds up the convergence remarkably.

4. NUMERICAL RESULTS

The numerical algorithms developed in this study were tested for three kinds of two-dimensional model flows as shown in Table I; asymmetric swirl flows decaying in time within a full circle, Lamb's multi-pole vortical flows within a full circle, which is periodic in time, and spin-up flows decaying in time within a semi-circular container. In the first flow model, we show that the zonal-embedded-grid system provides a significant improvement in the stability limit as for the time step. We also show the need for the use of the $O(\Delta\theta^4)$-algorithm in the proposed method. The reason for selecting the Lamb's multi-pole flows as the second problem is that they have exact solutions so that we can measure the numerical accuracy of the proposed method in comparison with the conventional one. In the third flow model, we compare our numerical results with the experimental results.

4.1. Asymmetric swirl flows within a full circle

The algorithms used in this flow model follow Sections 3.2 and 3.3, and the simple extrapolation schemes, i.e. (27a) and (27b), are used in the centre-point treatment. The Reynolds number is fixed at $Re = 1000$. Table II shows the limit of the time step Δt_{lim} obtained from numerical experiments for an asymmetric swirl flow within a full circle described in Section 2. The results reveal that even at the smallest number of grids the zonal-embedded-grid system permits a time step 12 times larger than that for the regular grid system. When the number of grids is largest, then the embedded grid system allows a 500 times larger time step than that for the regular grid system. As seen from this table, the benefit of the zonal-embedded grids, as for the time step, is more pronounced as the number of grids is increased.

For the swirl flow model given above, the numerical accuracy was checked qualitatively and comparatively by applying the second-order algorithms (22a) and (23a) ($O(\Delta\theta^2)$-algorithm hereafter) and fourth-order algorithms (22b) and (23b) ($O(\Delta\theta^4)$-algorithm hereafter) in evaluation of the azimuthal derivatives and the variables at undefined points. Figure 7 shows distribution of the numerically obtained momentum flux F. We can judge the accuracy by the smoothness of the distribution. It is seen that the second-order algorithm gives a non-smooth, deteriorated pattern, especially near the origin of the coordinates. Since the zonal-embedded grid is characterized by a small number of azimuthal grids near the origin, applying a higher-order algorithm such as fourth-order is crucial in establishing numerical accuracy. A more detailed quantitative comparison for the numerical accuracy among the various algorithms will be addressed in the following subsection.

Table II. Limit of the time step Δt_{lim} of each grid system above which the numerical computation becomes unstable.

Grid system ($I \times \bar{J}; M, I_0, J_0$)	Regular grid	Embedded grid
$32 \times 86; 4, 4, 16$	0.0033	0.040
$64 \times 171; 5, 4, 16$	0.00020	0.019
$128 \times 342; 6, 4, 16$	0.000012	0.0062

These results are obtained from the numerical computation for a decaying motion of a swirl flow described in the text with $r_c = 0.25$, $\rho_c = 0.4$ at $Re = 1000$.

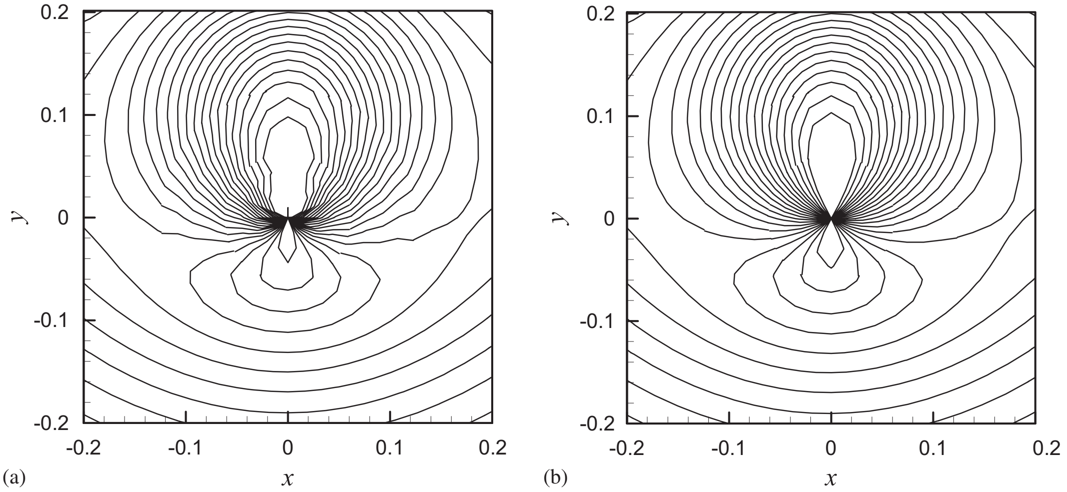

Figure 7. Contour lines of the function F, x component of the momentum flux, given by applying: (a) $O(\Delta\theta^2)$-algorithm; and (b) $O(\Delta\theta^4)$-algorithm in evaluation of the azimuthal derivatives and the undefined variables for the circular patch of swirl flow, described in the text, with $Re = 1000$, $r_c = 0.2$ and $\rho_c = 0.5$. The zonal-embedded grid is used with $I \times J = 65 \times 257$ ($M = 5$, $I_0 = 4$, $J_0 = 8$).

4.2. *Lamb's multi-pole flows*

The most important thing that must be done after a new algorithm or numerical method has been developed is to confirm the numerical accuracy and its performance. To do this, an appropriate flow model must be selected or designed. In this study, we selected the Lamb's multi-pole flows, which is periodic in time by introduction of time-periodic body forces in the momentum equations (refer to Table I). The exact solutions are given by (5a) and (5b), and typical flow fields are shown in Figure 8. Since the exact solutions are in a separable form, the vortices are stationary. In this subsection, we also show the numerical results obtained by using a semi-implicit fractional-step method with regular grids ('F-S-R' hereafter). The fractional-step method used in this computation is almost the same as that of Akselvoll and Moin [10], and the time integration is implicit only in the azimuthal direction.

Table III lists the numerical error ε_u and the run time obtained for the various combinations of algorithms, explained so far, for the Lamb multi-pole flows at $Re = 10$. Here, ε_u is defined as the time average of maximum error of u over the domain. At this Reynolds number, the flow becomes periodic from the second forcing period, that is at $t \geqslant 2$. Therefore the time average was performed for the data taken for the last period $2 \leqslant t \leqslant 3$. The run time shown at the last column in this table (and all for the other runs presented in this paper) is measured for the three forcing periods from a desk-top computer with 1.8 GHz CPU and 256 MB RAM.

The first case shown in Table IV follows the basic algorithms explained in Sections 3.2 and 3.3. Comparing this result with that given by F-S-R, we have found that this numerical error is far larger than F-S-R. After a careful investigation of the numerical data, we have designed to use governing equations with some viscous terms different from the original ones

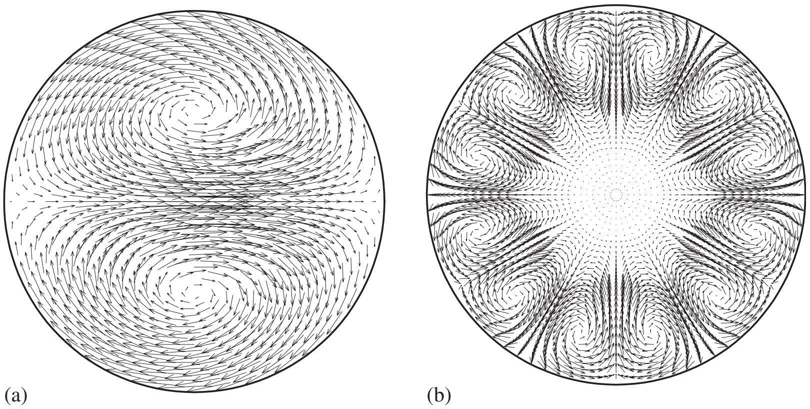

Figure 8. Velocity-vector plots of Lamb's dipole and 12-pole flows: (a) Lamb dipole; and (b) Lamb 12-pole.

Table III. Numerical error ε_u and the run time for the Lamb multi-pole flows at $Re = 10$ with the zonal-embedded grids; $M = 4$, $I_0 = 4$, $J_0 = 16$ $(I \times \bar{J} = 32 \times 86)$.

Case	No. of poles	Δt	Conserv. algorithm	Centre u, v	Diffusion terms	$\partial/\partial\theta^n$, interpo	ε_u	Run time
1	Dipole	1-03	Non	Extrapo	Old	4th	22.4-03	294 s
2	Dipole	1-03	Non	Extrapo	*New*	4th	3.04-03	341
3	Dipole	1-03	*Cons*	Extrapo	New	*2nd*	4.28-03	313
4	Dipole	1-03	Cons	*F&S*	New	2nd	4.59-03	315
5	Dipole	5-04	*Non*	F&S	New	*4th*	3.26-03	668
6	Dipole	5-04	Non	*Extrapo*	New	4th	3.04-03	656
7	*4-pole*	1-03	Non	Extrapo	New	4th	4.36-03	341
8	*6-pole*	1-03	Non	Extrapo	New	4th	2.80-03	339
9	*8-pole*	1-03	Non	Extrapo	New	4th	2.62-03	342
10	*10-pole*	1-03	Non	Extrapo	New	4th	3.05-03	346
11	*12-pole*	1-03	Non	Extrapo	New	4th	3.49-03	344
12	*16-pole*	1-03	Non	Extrapo	New	4th	4.72-03	344

The abbreviation 'non' in the fourth column indicates the non-conservative algorithm described in Sections 3.2 and 3.3 and 'cons' the conservative algorithm discussed in Section 3.5. Refer to the text for the 'diffusion terms' in the sixth column. The next column is associated with the order of accuracy in discretization of the azimuthal derivatives and interpolations described in Section 3.3. Refer to the text for the details of the numerical accuracy ε_u and the run time. Italic words stress the corresponding scheme change from the upper-row case.

Table IV. Effect of grid system (M, I_0, J_0) on the limit of the time step Δt_{lim} for a spin-up flow inside a semi-circular cylinder; $Re = 5000$, $\varepsilon = 1$, $h = 1$, $\omega = 10$.

Grid system $(I \times J; M, I_0, J_0)$	Δt_{lim}
$64 \times 256; 5, 4, 16$	0.012
$64 \times 256; 4, 8, 32$	0.0053
$64 \times 256; 3, 16, 64$	0.0017

(2a) and (2b). The low level of accuracy was found to arise from the terms

$$\text{viscous}_1 = \frac{\partial^2 u}{r^2 \partial\theta^2} - \frac{u}{r^2} - \frac{2}{r^2}\frac{\partial v}{\partial\theta} \tag{29a}$$

in (2a) and

$$\text{viscous}_2 = \frac{\partial^2 v}{r^2 \partial\theta^2} - \frac{v}{r^2} + \frac{2}{r^2}\frac{\partial u}{\partial\theta} \tag{29b}$$

in (2b), predominantly near the centre point of the domain. This can be explained for the case when the flow is uniform near the centre point. Suppose we have a uniform flow given by $u = a\cos\theta + b\sin\theta$ and $v = -a\sin\theta + b\cos\theta$. Then each of three terms in (29a) is evaluated to be $-(a\cos\theta + b\sin\theta)/r^2$, $-(a\cos\theta + b\sin\theta)/r^2$ and $2(a\cos\theta + b\sin\theta)/r^2$. Thus viscous_1 of (29a) becomes zero after summed up, but notice that each has a large value as the centre point is approached. Therefore even small relative errors contained in the constants a and/or b can result in significant errors in evaluation of the viscous terms; since errors in a and b should be larger when the grid resolution is poorer, the zonal-embedded-grid system is susceptible to such large errors compared with the regular grids. To solve this problem, we first introduce the continuity equation to transform (29a) and (29b) into the following:

$$\text{viscous}_1 = \left(\frac{\partial^2 u}{r^2 \partial\theta^2} + \frac{u}{r^2}\right) + \frac{2}{r}\frac{\partial u}{\partial r} \tag{30a}$$

$$\text{viscous}_2 = -\left(\frac{\partial^2 v}{r^2 \partial\theta^2} + \frac{v}{r^2}\right) - \frac{2}{r}\frac{\partial^2 u}{\partial\theta\partial r} \tag{30b}$$

In this form we still have the accuracy problem due to the terms within brackets, but the last term in each equation causes no significant problem. The next step is to assume locally

$$u = a_u\cos\theta + b_u\sin\theta + c_u, \quad v = a_v\cos\theta + b_v\sin\theta + c_v$$

and to substitute these into the terms within brackets. Then they become simply

$$\text{viscous}_1 = \frac{c_u}{r^2} + \frac{2}{r}\frac{\partial u}{\partial r} \tag{31a}$$

$$\text{viscous}_2 = -\frac{c_v}{r^2} - \frac{2}{r}\frac{\partial^2 u}{\partial\theta\partial r} \tag{31b}$$

The final step is representing the constants c_u and c_v in terms of the velocity components. It can be shown that by using the neighbouring grid points in the azimuthal direction, we end up with

$$c_u = \frac{u_N - 2u_0\cos\Delta\theta + u_S}{2(1-\cos\Delta\theta)} \tag{32a}$$

$$c_v = \frac{v_N - 2v_0\cos\Delta\theta + v_S}{2(1-\cos\Delta\theta)} \tag{32b}$$

where the subscripts 'N', '0' and 'S' are as defined in Figure 3. The test run of case 2 has been done with the 'new' scheme described above, and it is shown that the error is 7 times smaller than the conventional scheme, case 1. Therefore the numerical results to be presented in the followings were obtained only with the new scheme.

Notice that the form of (30a) and (30b) is exact as far as the incompressibility condition is satisfied. For a uniform flow defined as $u = \cos\theta$ and $v = -\sin\theta$, we can compare the error given from the application of the conventional central difference algorithm to (29a) and that given from application of (31a) with (32a) to the new form (30a). The exact value of the term viscous$_1$ in this case is zero. As a test cell, we take the u-cell very close to the origin with its centre being at $r = \Delta r$ and the azimuthal grid size of this cell being $\Delta\theta = 0.52$, corresponding to $J_0 = 12$. Now we apply the central difference algorithm to the original form (29a) (or equivalently to (30a)) to get

$$\text{viscous}_1 = \frac{\partial^2 u}{r^2 \partial\theta^2} - \frac{u}{r^2} - \frac{2}{r^2}\frac{\partial v}{\partial\theta} = \frac{1}{r^2}\left(\frac{2(\cos\Delta\theta - 1)}{\Delta\theta^2} - 1 + \frac{2\sin\Delta\theta}{\Delta\theta}\right) = -166$$

which is far from the exact value. On the other hand, the new form (30a) with (31a) and (32a) gives the exact value, zero.

The two test runs addressed above have been performed with the non-conservative algorithms. The word 'non-conservative' means that the corresponding scheme does not consider the energy conservation. In this category, we have two further branches of schemes; the first scheme is as described in Section 3.3 and the momentum is conserved for all the cells except for the interfacial cells, and the second branch of scheme considers the momentum conservation even for the interfacial cells. However, the numerical test revealed that the latter produces larger errors than the former. Therefore all the results produced by employing the non-conservative scheme were obtained by using the former one. On the other hand, the conservative scheme means that the energy conservation of the convection terms was considered, as presented in Section 3.5, but not at the interfacial cells. To apply the conservative algorithm, the fourth-order accurate scheme for the azimuthal derivatives and interpolations in the azimuthal direction cannot be adopted. The case 3 corresponds to such a set, and we can see that the numerical error is larger than the case with non-conservative but $O(\Delta\theta^4)$ schemes (case 2). When the F&S scheme was applied at the centre point with the conservative scheme (case 4), the situation becomes worse. In the case 5, the F&S scheme is combined with the non-conservative scheme and it turns out that the error becomes smaller. Again the centre-point treatment is changed from F&S to the extrapolation strategy, $\partial^3\phi/\partial r^3 = 0$, with all the other set unchanged, and then we have an improved accuracy (case 6). This then clearly indicates that as far as the numerical accuracy is concerned, both the non-conservative but $O(\Delta\theta^4)$-accurate scheme and the extrapolation scheme for the centre-point treatment are better than the conservative but $O(\Delta\theta^2)$-accurate scheme and the F&S method for the centre-point treatment.

The cases 7–12 test the effect of the flow types on the numerical accuracy. The result is not monotonous with the change of the number of poles. When the number of poles is 4 (case 7), the error is larger than that for the dipole flow. From the 4-pole's case the accuracy is improved up to 8-pole (cases 8 and 9). The error is then increased again when the number of poles is increased (cases 10–12). As shown in Figure 8, the vortex size becomes smaller and more concentrated at the outer region as the number of poles is increased. As will be shown below, such flow patterns are related to the optimal choice of the number of zones in the grid design.

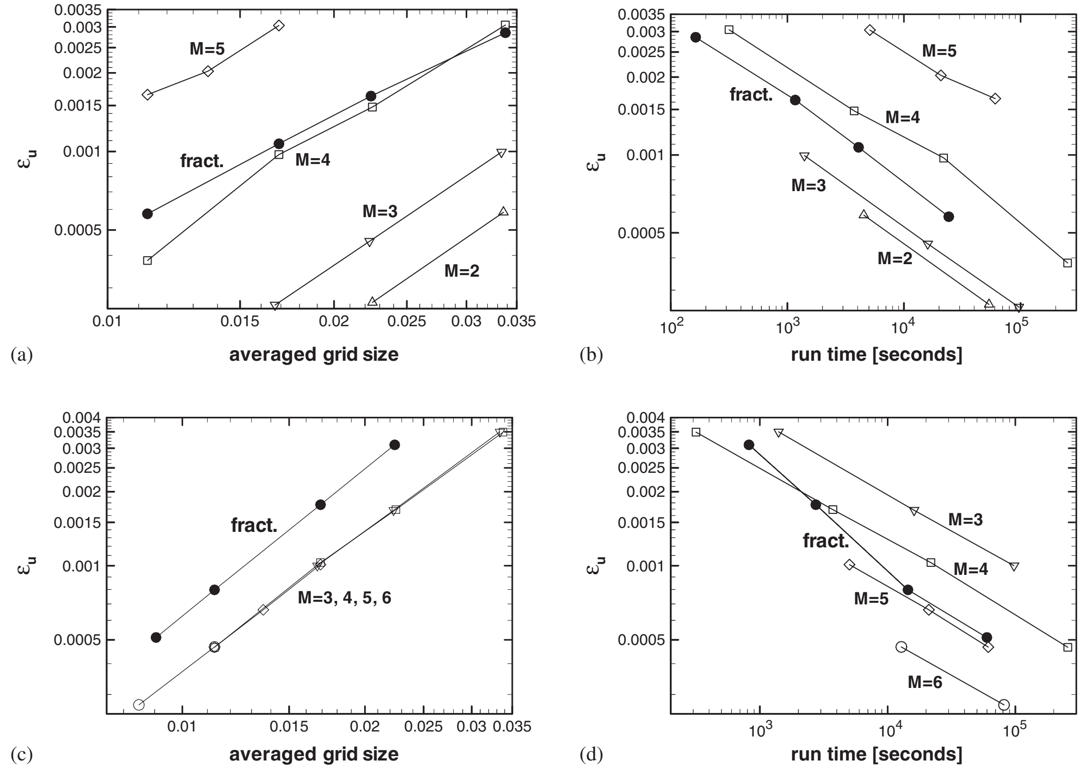

Figure 9. Numerical error ε_u for Lamb dipole and 12-pole flows. The lines with solid-circle symbols are given from the fractional-step method with regular grids. Open symbols are given from the R–K fourth algorithm with zonal-embedded grids with number of zones indicated by M: (a) error versus grid size for Lamb dipole; (b) error versus run time size for Lamb dipole; (c) error versus grid size for Lamb 12-pole; and (d) error versus run time size for Lamb 12-pole.

Figure 9 shows the numerical errors obtained by the embedded-grid method, in comparison with those given by F-S-R, for the Lamb dipole flows, (a) and (b), and for the Lamb 12-pole flows, (c) and (d) for various M values. Larger M indicates a smaller grid size. For the same M value the grid size can also be controlled by changing I_0 and J_0 (here, $J_0 = 4I_0$). The time step Δt taken in the numerics is given by $\Delta t = 0.9\Delta t_{\text{lim}}$ for both F-S-R and zonal-embedded-grid methods. Since the numerical errors are observed to be almost invariant to Δt, the shortest run time with a fixed grid resolution (and therefore with a fixed accuracy) would be attained by choosing $\Delta t = \Delta t_{\text{lim}}$. The number 0.9 corresponds to a safety factor. Figure 9(a) shows that the accuracy is improved by choosing a lower M value in the Lamb dipole flow. It is also remarkable to note that the zonal-embedded-grid system is superior to F-S-R (denoted as 'fract.' in this figure) for M less than 4. The second-order of accuracy upon the change of grid size is almost assured for both methods. The run time, as shown in Figure 9(b), is also in a lower level with the zonal-embedded-grid method than with F-S-R for M less than 4. For the case with larger poles, Figure 9(c) and (d), the second-order accuracy is also assured. In this flow model too, the zonal-embedded grids provide results better than

F-S-R if M is taken as 5 or 6. This implies that depending on the flow characteristics, we can choose in principle an optimal value of M which may give the best performance as for both the accuracy and the run time. For an example, when the turbulent pipe flow is concerned, if the flow pattern (e.g. Figures 4 and 5 of Reference [10] and Figure A2(a) in this paper) is assumed to be closer to Figure 8(b) than 8(a), we had better choose a higher M. However in practice determining an optimal M value should rely on the trial-and-error iterations for the most problems.

The superiority of the zonal-embedded-grid method in the overall numerical performance comes from a few factors. First, application of the $O\Delta\theta^4$ algorithm to discretization of the azimuthal derivatives and interpolations and use of the transformed diffusive terms bring a significant improvement in the numerical accuracy. Second, the extrapolation scheme (28) used in obtaining the initial pressure field in solving the pressure equation in the fourth-order Runge–Kutta method for the zonal-embedded grids provides very fast convergence property, which in part saves the run time. On the other hand, for the F-S-R, such scheme does not work with unknown reasons.

In fact, the solver of the pressure equation for the zonal-embedded grids is composed of inner and outer iterations. Within the inner iteration, some of the interfacial pressures are fixed with known values as explained in Section 3.2 (e.g. p^o_W and p^o_{W1} in Equation (20)) and they are updated every time the inner iteration has been converged and the next outer iteration starts. Of course one may expect that such process may affect the overall convergence property of the pressure solver. Figure 10 shows the convergence history of the pressure solvers for the regular grids and zonal-embedded grids. The velocity components for this run are given by the Lamb's dipole solutions (5a) and (5b) and the initial values of the pressure are set at zero everywhere; this is why the run needed iteration counters much larger than the usual situation. The zonal-embedded-grid solver converges after two outer iterations and the total number of iterations up to the final convergence is slightly larger than that with the regular grids. After some tens of the main iterations from the beginning of the run, however, the zonal-embedded-grid system needs only a few inner iterations due to the extrapolation scheme, (28), and does not need the second outer iteration. This means that the strategy of treating some pressures at the interfacial cells as known in the iteration process exerts almost no influence on the pressure-convergence property and the overall numerical accuracy. So, all the results presented in this study are obtained only with one outer iteration.

4.3. Spin-up in a semi-circular container

We next applied the code to the non-axisymmetric flows, i.e. spin-up flows inside a semi-circular boundary. In contrast to the spin-up within a full circular container, due to the existence of corners inside the domain, this configuration generates a boundary-layer separation from the solid walls. The separated flow then undergoes a vortex roll-up, vortex growth, and vortex merging. Such dynamical behaviour is most typical in the complex fluid flows and thus the spin-up flow inside a semi-circular container is thought to be suitable in verifying the code for a general purpose. This kind of flow was first studied experimentally by van Heijst [24] at the Reynolds number 1.6×10^5 and for various liquid depths. The three-dimensional numerical computation of this flow was then conducted by Andersson *et al.* [25]. However, their numerical results are not in good agreement with the experimental results. For instance, the experimentally observed phenomena such as vortex merging were

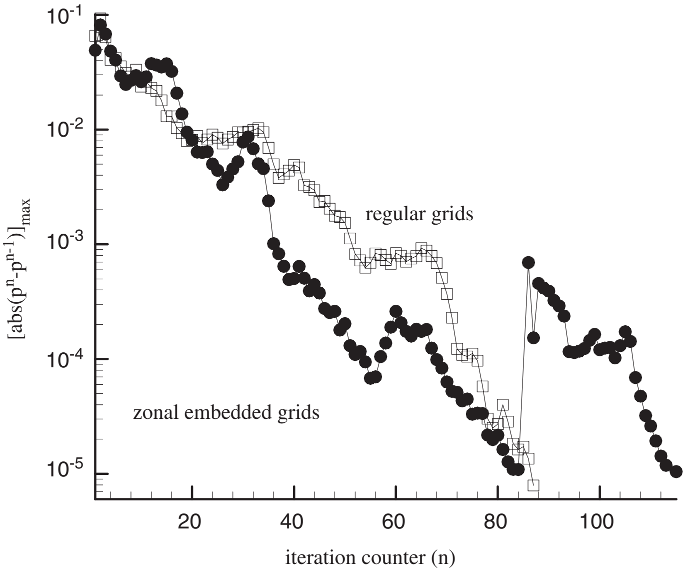

Figure 10. The convergence property of the ICCG method for solving the pressure equation for the Lamb dipole flow with the regular grids (open square) and that with the zonal-embedded grids (solid circle); $I \times \bar{J} = 64 \times 171$. From the iteration counter 86, ICCG with the zonal-embedded grids enters into the second outer iteration, and it stops at the beginning of the third outer iteration because it has converged.

not observed in the numerical experiment. The fundamental reason for such a discrepancy seems to be the low resolution of the grid system for such a high Reynolds number flow.

The numerical results presented in this subsection are obtained from the code with zonal-embedded grids employing the non-conservative $O(\Delta\theta^4)$ algorithm with the viscous terms in the original form.

First of all, the numerical stability was tested to this flow. Table IV shows the results for $Re = 5000$ and $I \times J = 64 \times 256$, which are fixed, and for three choices of (M, I_0, J_0). This result suggests that when decreasing M and increasing I_0 and J_0 simultaneously, we should make the time step smaller for a stable computation. Since increase of J_0 results in the decrease of the azimuthal-grid size near the origin, it again implies that the numerical stability is simply decided by the azimuthal-grid size at the origin, which is the smallest over the domain. Figure 11 shows the dependence of the limit time step Δt_{lim} on the Reynolds number for three grid systems. In each grid system, the value increases linearly with Reynolds number when the number is low. In this regime the dependence of Δt_{lim} on Re can be described by the empirical formula

$$\Delta t_{\mathrm{lim}} = (0.6\Delta s_c^2)Re \tag{33}$$

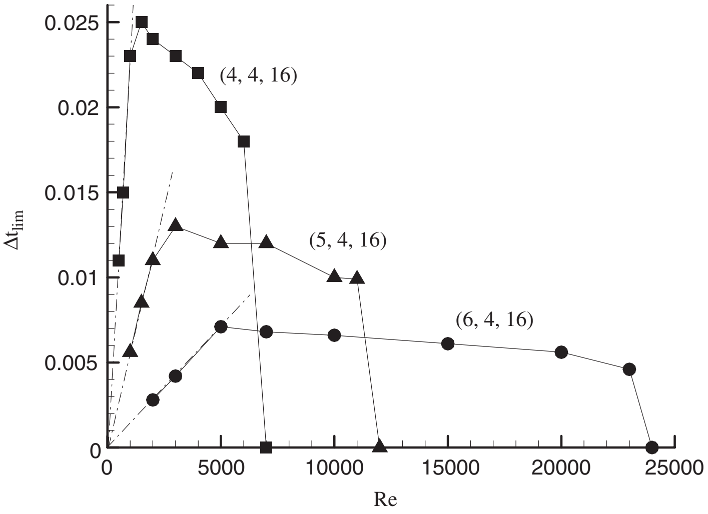

Figure 11. Stability diagram of the fourth-order Runge–Kutta method applied to the spin-up flow inside a semi-circular boundary. This diagram is obtained numerically at $\varepsilon = 1$, $h = 1$, $\omega = 10$ and with three grid systems identified by (M, I_0, J_0) shown. The dash–dot lines are drawn with formula (33) in the text.

where Δs_c denotes the azimuthal spacing of the smallest cell contacting the origin. This value can be obtained by

$$\Delta s_c = \frac{\pi}{J_0(I-1)}$$

It is shown in Figure 11 that three asymptotic lines given by (33) well fit the numerical results. Relationship (33) can be derived from the theoretical formula

$$\Delta t_{\text{lim}} = C \left[\frac{|u|}{\Delta r} + \frac{|v|}{\Delta s} + \frac{4}{Re} \left(\frac{1}{\Delta r^2} + \frac{1}{\Delta s^2} \right) \right]^{-1} \tag{34}$$

given by Akselvoll and Moin [10], by retaining only the last term within ( ). This figure also shows that there exists a Reynolds-number limit above which the given grid system cannot produce a stable solution, which suggests that as Reynolds number is increased the number of grids must also be increased for a stable solution.

Figure 12 shows the development of the velocity vectors, at $Re = 5000$ and $\varepsilon = 0.5$, given from the numerical solution in comparison with the experimental pathlines. In this flow, the speed-up of the container ends at $t = 0.48$ and its speed is maintained at this value thereafter. The background rotation is counterclockwise. Since most of the pathlines do not intersect with each other, they can be considered to be very close to streamlines. First of all, it is remarkable to see that the numerical solution exactly reproduces the experimentally observed one. At the very beginning of the development the flow is occupied by a large anticyclonic cell (to be referred to as ‘primary’ vortex hereafter), which is inviscid with a uniform vorticity $\varsigma = -2$ except for the thin boundary layers adjacent to the walls. Due to the existence of two corners,

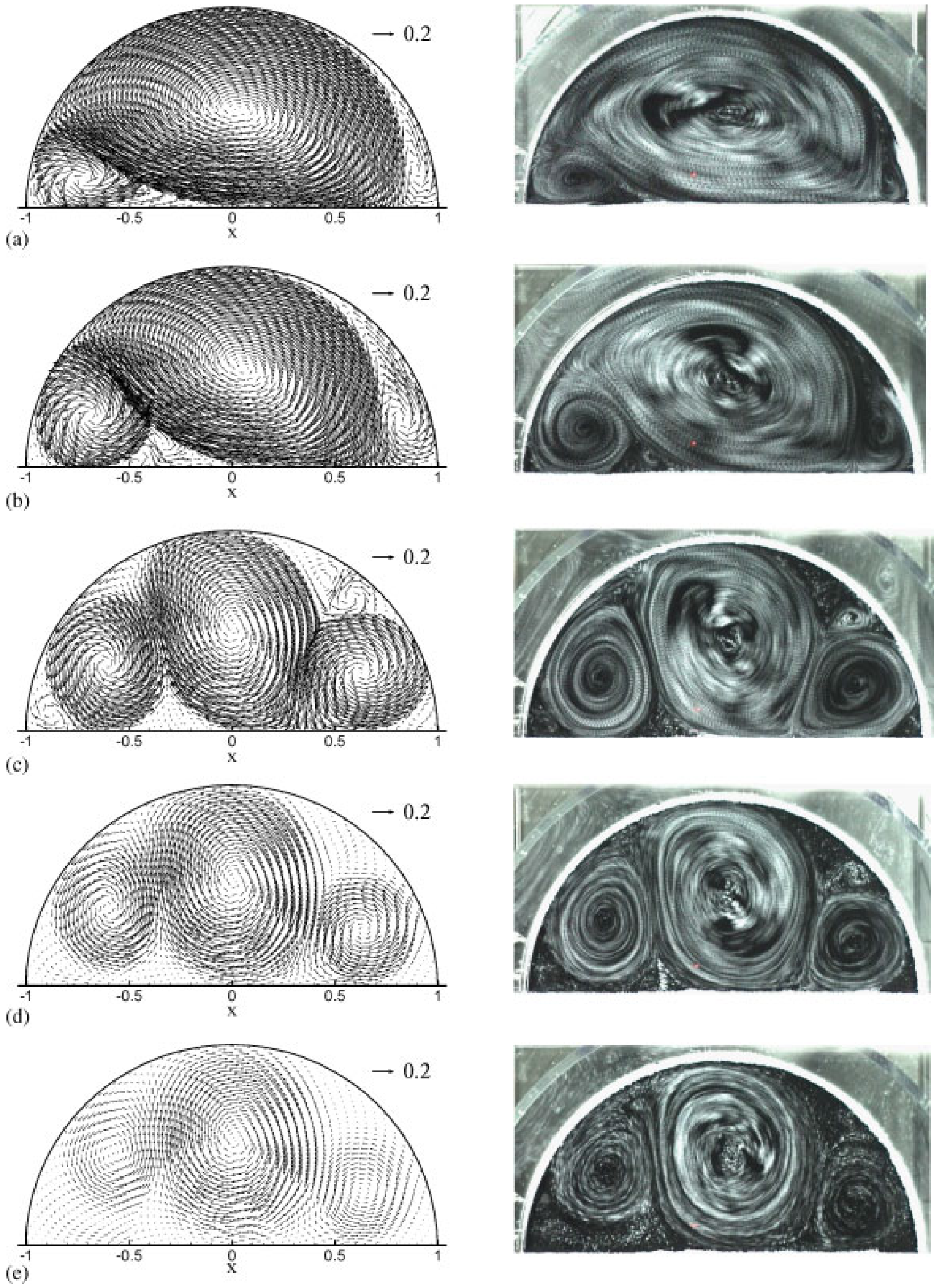

Figure 12. Development of the velocity-vector field obtained numerically (left-hand side) for $Re = 5000$, $\varepsilon = 0.5$, $h = 1$ and $\omega = 6.54$ with grids $(M, I_0, J_0) = (5, 4, 16)$ and development of the pathlines obtained from a flow-visualization experiment with acrylic powders floated on tap water: (a) $t = 4$; (b) $t = 6$; (c) $t = 12$; (d) $t = 24$; and (e) $t = 40$.

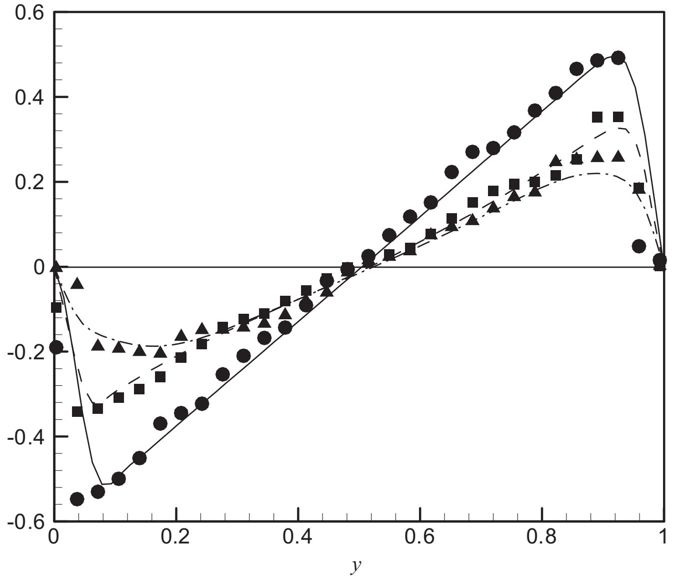

Figure 13. u-velocity profiles along the y-axis at $t=6$ (solid line and circles), $t=12$ (dashed line and squares) and $t=12$ (dash–dot line and deltas) obtained numerically (lines) and experimentally (symbols) for the spin-up flow in a semi-circular boundary with the same parameter set as Figure 12.

however, this vortex brings fourth boundary-layer separations from the surrounding walls and the separated layers roll up to form a cyclonic vortex near each corner (Figure 12(a)). The corner vortices (to be referred to as 'secondary' vortices hereafter) then grow to be almost comparable in size with the primary vortex (Figure 12(c)–(e)). During this development, the primary vortex imparts its material to the secondary vortices via the roll-up process discussed above and shrinks finally to a size approximately one-third of the domain. At the final stage (Figure 12(d) and (e)) the secondary vortex at the left-hand side is slightly shifted toward the circular boundary, but it stays there all the way until the flow vanishes completely. For this flow, we also tested the difference between the one-sided difference scheme (17a) and the centred difference scheme (18), and no difference was detected between the two results.

Figure 13 shows the distribution of the azimuthal velocity component along the line $\theta=\pi/2$ obtained numerically for the same parameter values as Figure 12 in comparison with the experimental results measured by the PIV method. Here the PIV results are obtained by using the MQD-PIV method developed by Suh [26]. Two results are in good agreement with each other except near the boundary, where due to a high velocity gradient it is generally difficult to measure the velocity accurately. Overall, the velocity decreases in time due to the damping caused by the pumping/suction of the fluid material from the Ekman boundary-layer on the bottom wall of the container; such pumping/suction is treated by the Ekman pumping velocity W_{E}, Equation (6).

The effects of the parameters, Re, ε and h on the flow pattern are expected to be significant as predicted by Suh and Choi [23] for the case of spin-up flows in a rectangular container. Figure 14 exhibits the decay of the spatially averaged kinetic energy $E(t)$ for three liquid

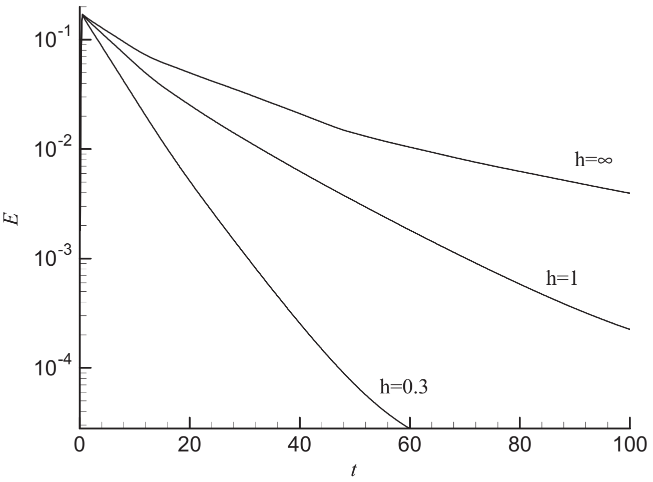

Figure 14. Temporal history of the spatially averaged kinetic $E(t)$ obtained numerically for the spin-up flow in a semi-circular boundary with three liquid depths shown; all the other parameter values are the same as Figure 12.

depths. It is seen that E is monotonously decreasing with time for any liquid depth. The decay rate is however strongly dependent on the liquid depth; a shallower liquid results in a higher decay rate. This can be explained in terms of the relation between the vorticity dynamics and the Ekman pumping velocity W_{E}. When the pressure is eliminated from the momentum equations (2a) and (2b) we can obtain the vorticity transport equation, and it can be shown that the vorticity decays in proportion to W_{E}/h; see also Reference [18]. As is seen from (6), W_{E} is independent of h, therefore the vorticity decays in inverse proportion to h.

The numerical results for $h=\infty$ are shown in Figure 15. Compared with the case of $h=0.3$, Figure 12, the velocity magnitudes are large; e.g. compare Figures 12(d) and 15(a). This is consistent with the theoretical analysis as for the effect of the liquid depth on damping discussed above. However, Figure 15 reveals a peculiar behaviour in that the secondary vortex on the left-hand side travels fast along the circular boundary and merges with the secondary vortex at the right-hand side region. This phenomenon can be explained in terms of the effect of the neighbouring vortices and the circular boundary. The flow pattern at $t=24$, Figure 15(a), is in fact very similar to that for the lower depth case, Figure 12(d). For the infinite-depth case, however, the secondary vortex on the left-hand side is strong enough to generate another vortex (hereafter to be referred to as 'third' vortex) near the left-hand side corner, Figure 15(b). On the other hand, the secondary vortex, located in between the primary and third vortices, resides close to the circular boundary, so it should have a tendency to move toward the left-hand side due to the image-vortex effect from outside the domain. On the contrary, the primary and third vortices have clockwise circulations and thus tend to bring the secondary vortex to the right-hand side. This then means that the observed transposition of the secondary vortex must have been driven dominantly by the primary and third vortices. Such transposition of the corner vortex is also typical at high Reynolds numbers, as will be shown shortly. On the other hand, the case with a lower depth at the

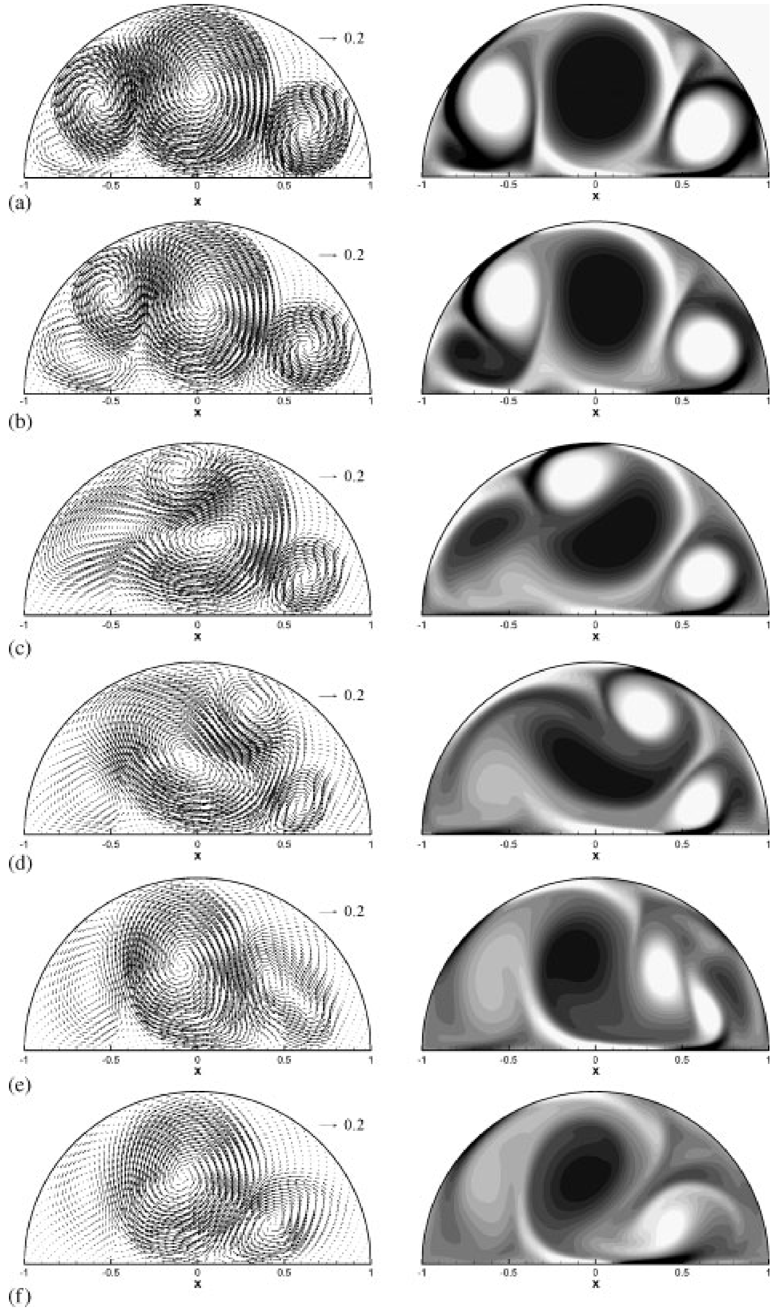

Figure 15. Development of velocity vectors (left-hand side) and vorticity contours with the same parameter set as Figure 12 except that this is for $h=\infty$. In the vorticity-contour plots, the vorticity values larger than 2 are coloured white, whereas the values less than -2 are black: (a) $t=24$; (b) $t=32$; (c) $t=40$; (d) $t=44$; (e) $t=48$; and (f) $t=52$.

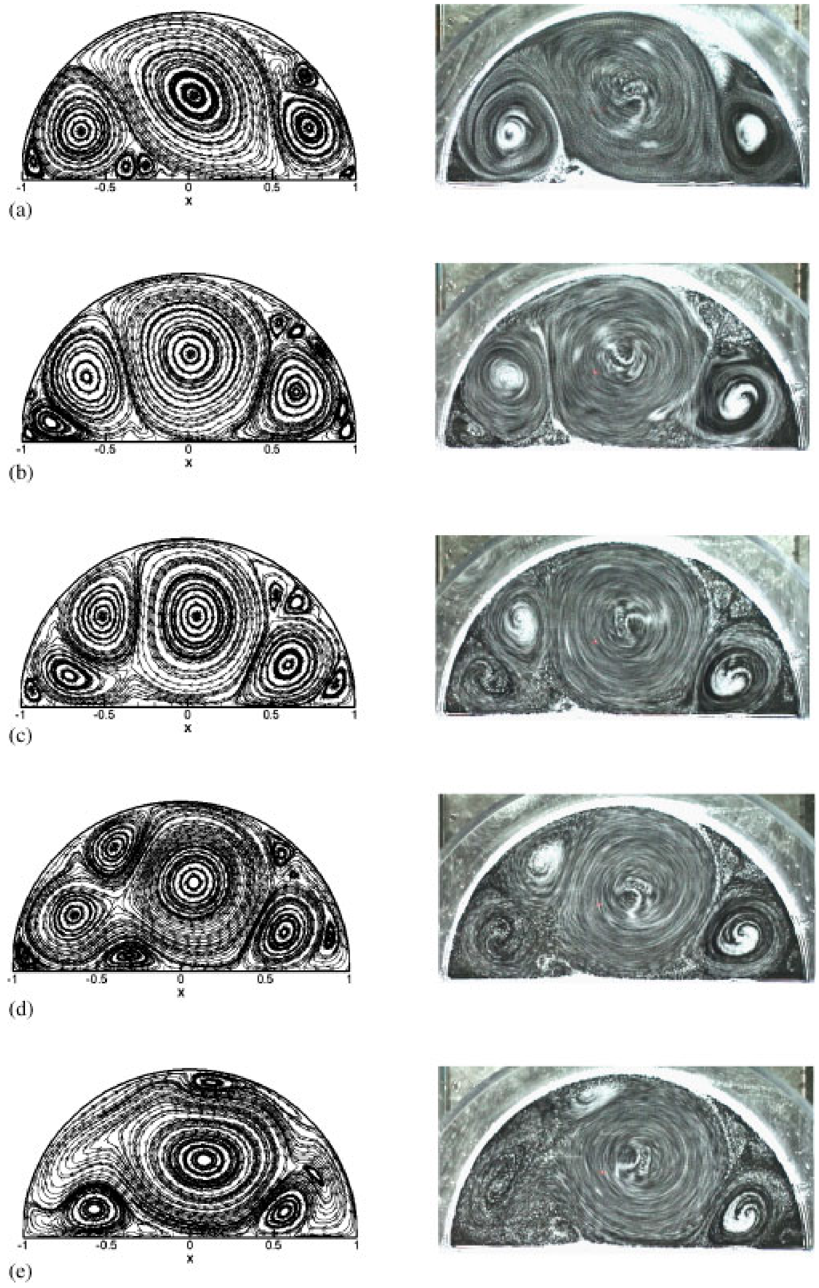

Figure 16. Development of the streamlines obtained numerically (left-hand side) for $Re = 10\,000$, $\varepsilon = 1$, $h = 1$ and $\omega = 6.54$ with grids $(M, I_0, J_0) = (5, 8, 32)$ in comparison with the pathlines obtained from a flow-visualization experiment (right-hand side): (a) $t = 8$; (b) $t = 24$; (c) $t = 40$; (d) $t = 48$; and (e) $t = 56$.

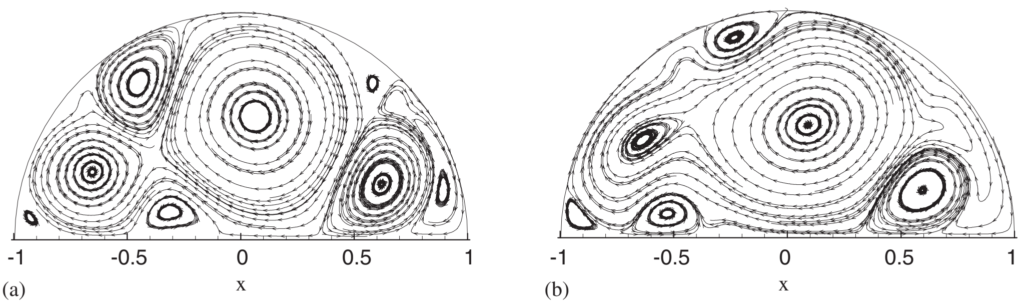

Figure 17. Development of the streamlines obtained numerically with the same flow parameter set as Figure 16 but with a higher grid resolution; $(M, I_0, J_0) = (5, 10, 40)$: (a) $t = 48$; and (b) $t = 56$.

same Reynolds number, $Re = 5000$, does not reveal such transposition, as shown in Figure 12 for $h = 0.3$. The secondary vortex moving fast around the circular boundary is usually smaller than (sometimes much smaller than) the primary vortex, and the movement is, as discussed above, driven mainly by the primary vortex. In this sense, we will call this 'satellite' vortex.

At higher Reynolds numbers and Rossby numbers, the flow pattern becomes more complex due to a lower damping effect. Figure 16 shows streamlines obtained numerically and pathlines given from the experiment for $Re = 10\,000$ and $\varepsilon = 1$. The experimental pathlines are considered to be approximately the same as the corresponding streamlines. Many small third vortices appear strong in the regions between the primary or secondary vortices and corners or walls, where the flow was almost stagnant at lower Reynolds numbers (Figure 12). Here again, the third vortex near the left-hand side corner and the primary vortex enforce the satellite vortex travel along the circular boundary (Figure 16(d) and (e)) toward the right-hand side of the domain. After the transposition, the vortex is expected to merge with the secondary vortex situated near the right-hand corner. Overall, the numerical and experimental results are in good agreement with each other except in Figure 16(e), where the numerical results reveal a faster movement of the satellite vortex than the experiment. Such a slight difference disappears when a higher resolution of the grid system is adopted in the numerical computation as shown in Figure 17. Now, the streamline pattern observed numerically at $t = 56$ (Figure 17(b)) is in a perfect agreement with the experimental result of Figure 16(e). On the other hand, the numerical results at the higher grid resolution are indistinguishable from the corresponding ones of Figure 16 at $t \leqslant 48$ and so they are not presented in this paper.

For $Re = 20\,000$ and $\varepsilon = 1$, fairly good prediction was also possible at roughly $t \leqslant 40$. However after that time, considerable discrepancy was found to exist between the numerical and experimental results even with higher grid resolutions. Considering that the Ekman-pumping velocity (6) has been derived from a quasi-steady linear approximation to the Ekman boundary-layer on the bottom wall of the container under the assumption of high Reynolds numbers and small Rossby numbers but with moderate vorticity, we cannot expect an accurate long-term prediction from the two-dimensional numerical simulations especially at such high Reynolds number and moderate Rossby number flows.

5. CONCLUSIONS

In this study, a zonal-embedded-grid technique has been proposed for numerical simulation of flows within a circular boundary with cylindrical coordinates. The benefit of this method is that even with the explicit method, it allows a time step far larger than the conventional regular-grid system.

In discretization of the azimuthal derivatives and evaluation of variables at undefined points, this method requires employing higher-order algorithms such as fourth-order to establish numerical accuracy, because the number of grids along the azimuthal direction is relatively small near the origin of the coordinates.

The developed code was applied to decaying swirl flows and time-periodic Lamb multi-pole flows within a full circle and spin-up flows within a semi-circle. Application to the Lamb multi-pole flows reveals that by choosing a suitable number of zones, M, we can achieve an overall performance of the proposed method better than the conventional one with regular grids.

For the spin-up flow case, the code was verified by comparing the numerical results with those from the visualization experiment. At relatively low Reynolds and Rossby numbers ($Re = 5000$, $\varepsilon = 0.5$), the spin-up flow organized by a three-cell structure was well predicted by the two-dimensional numerical simulation. At high Reynolds and Rossby numbers ($Re = 10\,000$, $\varepsilon = 1$), the flow field was occupied by many smaller vortices, and in this case too the numerical results well predict the overall vertical dynamics including the peculiar phenomenon observed in the experiment, i.e. transposition of the satellite vortex from one corner to the other.

We have successfully applied the zonal-embedded-grid method to the DNS of a fully developed turbulent pipe flow as shown in the appendix. It is seen that the results are in good agreement with those of Fukagata and Kasagi [9] even with far less grids.

APPENDIX A: DNS OF A TURBULENT PIPE FLOW

The code was developed with the zonal-embedded grids to solve the three-dimensional fully developed turbulent pipe flow. The governing equations are of a standard form similar to (1)–(2b) (also refer to e.g. Reference [9]). The periodic boundary conditions are applied at the entrance $z = 0$ and at the exit $z = h$, where h is the dimensionless axial length of the pipe subject to the numerical computation. The reference velocity here is the friction velocity u_τ. The time-averaged pressure gradient along the pipe axis in a dimensionless quantity is -2, and as such we introduce a new pressure $\tilde{p}$ defined as $\tilde{p} = p + 2z$ so that we can apply a periodic boundary condition for $\tilde{p}$ at the entrance and at the exit. The initial profile chosen for the axial velocity component is

$$w_{\text{ini}}(r, \theta, z) = w_{\text{amp}} \frac{Re}{2}(1 - r^n)$$

where $w_{\text{amp}} = 0.15$ and $n = 7$ was selected, after several test runs, to shorten the time needed to reach the fully developed velocity profile. The initial distributions for the radial and azimuthal velocity components were computed from asymmetric swirl flow defined in Section 2 with $r_c = 0.2$ and $\rho_c = 0.6$. To accelerate the convergence to the fully developed state, these

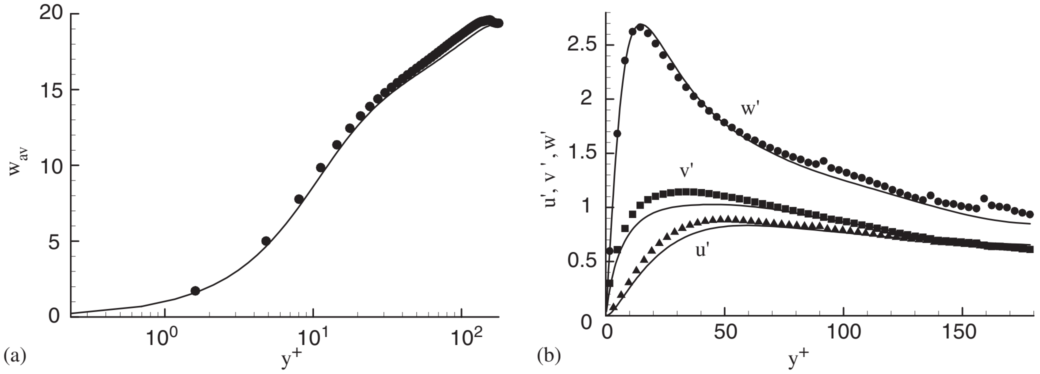

Figure A1. Typical results (symbols) of DNS for the fully developed turbulent pipe flow at $Re = 180$ in comparison with those (solid lines) from Reference [9]: (a) the axial mean velocity profile, w_{av}; and (b) RMS velocity fluctuations, u', v' and w'.

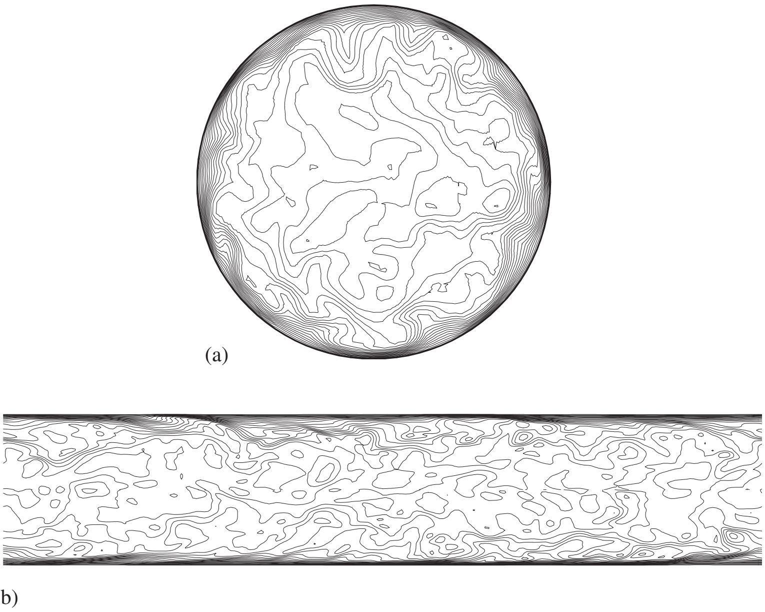

Figure A2. Instantaneous distributions of w at $t = 15$ obtained from the present results: (a) the (r, θ) plane at $z = 5$; and (b) the (r, z) plane at $\theta = \pi/2$.

velocity components are amplified by the factor 7. The three-dimensional turbulent pipe flow is characterized by a strongly random and fluctuating velocity field. Spatially, the flow pattern is far more complex than the laminar flow case. So, in this application we developed more accurate schemes for discretization of the azimuthal derivatives and interpolation along the azimuthal direction aside from the conventional ones described in Section 3. The conventional algorithm used for the discretization and interpolation is based on the polynomial functions. However, such polynomial functions may not be accurate enough near the origin of the cylindrical coordinate system when the number of grids is comparatively small especially for the complex flows like turbulence. In this regard we use the harmonic function

$$\phi = c + a_1 \cos \vartheta + b_1 \sin \vartheta + a_2 \cos 2\vartheta + b_2 \sin 2\vartheta$$

as the local representation of an arbitrary variable $\phi(\vartheta)$, where ϑ denotes the local azimuthal coordinate the origin $\vartheta = 0$ being at the point where we need to discretize the derivatives or to obtain the interpolated values. The five constants $c, a_1, b_1, \ldots,$ are determined by using the five neighbouring values of ϕ, e.g. ϕ_{-2}, ϕ_{-1}, ϕ_0, ϕ_1 and ϕ_2, where ϕ_k denotes the value of ϕ at the point $\vartheta = k\Delta\theta$. The value $\phi_{1/2}$ may be obtained by interpolation from the harmonic function constructed either from the set $(\phi_{-2}, \phi_{-1}, \phi_0, \phi_1, \phi_2)$ or from $(\phi_{-1}, \phi_0, \phi_1, \phi_2, \phi_3)$. However, it turned out that neither of the schemes leads to a successful run of the code. Such problem was solved by taking an average of the two results as the interpolated value. Next, the first and second derivatives are discretized by using the following algorithms:

$$\left[\frac{\partial \phi}{\partial \theta}\right]_{\vartheta = 0} = d_1(\phi_{1/2} - \phi_{-1/2}) + d_2(\phi_{3/2} - \phi_{-3/2})$$

$$\left[\frac{\partial^2 \phi}{\partial \theta^2}\right]_{\vartheta = 0} = e_0\phi_0 + e_1(\phi_1 + \phi_{-1}) + e_2(\phi_2 + \phi_{-2})$$

where d_1, d_2, e_0, e_1 and e_2 are constants.

The developed code was run with the parameter values $Re = 180$, $h = 10$, $\Delta t = 0.001$, $M = 4$, $I_0 = 7$, $J_0 = 12$, $K = 201$, where K is the number of grids along the axial direction. It took 160 h in a workstation to obtain the solutions up to $t = 15$. It was found that the flow establishes to a fully developed state after $t = 10$. So, the data within $10 \leqslant t \leqslant 15$ have been used in analysing the turbulent characteristics. Figure A1 shows typical numerical results in comparison with those given by Fukagata and Kasagi [9]. Here y^+ denotes the distance from the wall in the wall unit; $y^+ = Re(1 - r)$. The present results are obtained with coarser grids, $53 \times 65 \times 201$, than those of Fukagata and Kasagi, $96 \times 128 \times 256$, but the two results are in overall in a good agreement. We can detect a slight over-prediction in the RMS velocity fluctuations u' and v' near the wall and w' near the centre, and this difference is of course due to the coarser grids in the present calculations. Hopefully, we can curtail the slight protrusions shown in the distribution of w' at the points in between the neighbouring zones, when finer grids are employed. Figure A2 exhibits the contour plots of the instantaneous velocity w at $t = 15$. The figure reveals smooth patterns without any discontinuity at the interfaces between the neighbouring zones, and further they are very similar in a statistical sense to the patterns presented by Akselvoll and Moin [10].

ACKNOWLEDGEMENTS

This work has been supported by the Korea Research Foundation (KRF-2002-D00080). This work has also been supported by the National-Research-Laboratory program provided by Korea Science and Engineering Foundation. We appreciate Professor M. Duffy for reading the manuscript.

REFERENCES

1. Verkley WTM. A spectral model for two-dimensional incompressible fluid flow in a circular basin. I. Mathematical formulation. *Journal of Computational Physics* 1997; **136**:100–114.
2. Verkley WTM. A spectral model for two-dimensional incompressible fluid flow in a circular basin. II. Numerical examples. *Journal of Computational Physics* 1997; **136**:115–131.
3. Priymak VG, Miyazaki T. Accurate Navier–Stokes investigation of transitional and turbulent flows in a circular pipe. *Journal of Computational Physics* 1998; **142**:370–411.
4. Lopez JK, Shen J. An efficient spectral-projection method for the Navier–Stokes equations in cylindrical geometries. I. Axisymmetric cases. *Journal of Computational Physics* 1998; **139**:308–326.
5. Lopez JK, Marques F, Shen J. An efficient spectral-projection method for the Navier–Stokes equations in cylindrical geometries. II. Three-dimensional cases. *Journal of Computational Physics* 2002; **176**:384–401.
6. Manna M, Vacca A. An efficient method for the solution of the incompressible Navier–Stokes equations in cylindrical geometries. *Journal of Computational Physics* 1999; **151**:563–584.
7. Constantinescu GS, Lele SK. A highly accurate technique for the treatment of flow equations at the polar axis in cylindrical coordinates using series expansions. *Journal of Computational Physics* 2002; **183**:165–186.
8. Verzicco R, Orlandi P. A finite-difference scheme for three-dimensional incompressible flows in cylindrical coordinates. *Journal of Computational Physics* 1996; **123**:402–414.
9. Fukagata K, Kasagi N. Highly energy-conservative finite difference method for the cylindrical coordinate system. *Journal of Computational Physics* 2002; **181**:478–498.
10. Akselvoll K, Moin P. An efficient method for temporal integration of the Navier–Stokes equations in confined axisymmetric geometries. *Journal of Computational Physics* 1996; **125**:454–463.
11. Naik V, Naik N, Nicoules M. Implicit CFD applications on message passing multiprocessor. In *Parallel Computational Fluid Dynamics*, Simon HD (ed.). MIT Press: Cambridge, MA, 1992; 97–126.
12. Giraud L, Manzini G. Parallel implementations of 2D explicit Euler solvers. *Journal of Computational Physics* 1996; **123**:111–118.
13. Ecer A, Gopalaswamy N, Tarkan I. Parallel computation of flow problems using explicit schemes. In *Computational Fluid Dynamics for the 21st Century*, Hafez M, Morinishi K, Periaux J (eds). Springer: Berlin, 2001; 211–221.
14. Lomax H, Pulliam TH, Zingg DW. *Fundamentals of Computational Fluid Dynamics*. Springer: Berlin, 2001.
15. Lin P, Guo Q, Chen X. A fully explicit methods for incompressible flow computation. *Computer Methods in Applied Mechanics and Engineering* 2003; **192**:2555–2564.
16. Patera AT. A spectral element method for fluid dynamics: laminar flow in a channel expansion. *Journal of Computational Physics* 1984; **54**:468–488.
17. Fischer PF, Ronquist EM. Spectral element methods for large scale parallel Navier–Stokes calculations. *Computer Methods in Applied Mechanics and Engineering* 1994; **116**:69–76.
18. Timmermans LJP, Minev 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.
19. Chang R-Y, Hsu C-H. A variable-order spectral element method for incompressible viscous flow simulation. *International Journal for Numerical Methods in Engineering* 1996; **39**:2865–2887.
20. McIver DM, Blackburn HM, Nathan GJ. Spectral element-Fourier methods applied to simulation of turbulent pipe flow. *ANZIAM Journal* 2000; **42**(E):C954–C977.
21. Kravchenko AG, Moin P, Moser R. Zonal embedded grids for numerical simulations of wall-bounded turbulent flows. *Journal of Computational Physics* 1996; **127**:412–423.
22. Greenspan HP. *The Theory of Rotating Fluids*. Cambridge University Press: Cambridge, U.K., 1968.
23. Suh YK, Choi YH. Study on the spin-up of fluids in a rectangular container using Ekman pumping models. *Journal of Fluid Mechanics* 2002; **458**:103–132.
24. van Heijst GJF. Spin-up phenomena in non-axisymmetric containers. *Journal of Fluid Mechanics* 1989; **206**: 171–191.
25. Andersson HI, Billdal JT, van Heijst GJF. Spin-up in a semicircular cylinder. *International Journal for Numerical Methods in Fluids* 1992; **15**:503–524.
26. Suh YK. Multi-frame MQD-PIV. *KSME International Journal* 2003; **17**:1463–1473.