Journal of Computational Physics 341 (2017) 377–385

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Short note

A note on a conservative finite volume approach to address numerical stiffness in polar meshes

Rajapandiyan Asaithambi, Krishnan Mahesh *

Aerospace Engineering and Mechanics, University of Minnesota, United States

ARTICLE INFO

Article history:
Received 5 January 2017
Received in revised form 24 March 2017
Accepted 7 April 2017
Available online 12 April 2017

Keywords:
Conservative
Finite-volume
Stiffness
Polar meshes
DNS
LES

ABSTRACT

A polar coordinate system introduces a singularity at the pole, $r = 0$, where terms with a factor $1/r$ can be ill-defined. While there are several approaches to eliminate this pole singularity in finite difference methods, finite volume methods largely bypass this issue by not storing or computing data at the pole. However, all methods face a very restrictive time step when using an explicit time advancement scheme in the azimuthal direction, where cell sizes are of the order $O(\Delta r(r\Delta\theta))$. We use a conservative finite volume approach of merging cells on a structured O-mesh to remove this time step limit imposed by the CFL condition. The cell-merging procedure is implemented as a corrector step and incurs no changes to the underlying data structure for a structured grid. This short note describes the procedure and presents the validation and application of the algorithm to various problems. The algorithm is shown to be inexpensive and scalable. In addition, the cell-merging procedure is easily coupled with a line implicit scheme in the radial direction.

1. Introduction

A large number of canonical problems are naturally defined in polar or cylindrical coordinates such as flows in pipes, round jets, vortices, axisymmetric wakes and shear layers. The applications range from transport of fluids, mixing, combustion, aeroacoustics, and external aerodynamics of axisymmetric objects and their boundary layers. Jets form a large subset of these canonical problems owing to their ubiquity in industrial applications. The need to study jets in the real world, ranging from cooling micro-jets to jet engines necessitates us to develop algorithms that can handle round turbulent jets. Specifically, our interests lie in studying auto-igniting fuel jets in vitiated co-flow conditions [6] where controlling autoignition can lead to the design of efficient internal combustion engines.

While Cartesian meshes can be used to study round jets [2,5], they can be inefficient when it comes to the number of grid cells required. In general, structured meshes have an inherent advantage over unstructured meshes in computational cost due to the regular data structure. Some of the largest direct numerical simulations have been performed with Cartesian meshes [16,15]. For round jets however, most of the grid cells are needed in the vicinity of the jet's shear layer. A cylindrical mesh is appropriate in this case as it allows clustering of cells in the shear layer. We estimate that to simulate the auto-igniting fuel jet [6], a Cartesian mesh would need about 10 times more cells than a cylindrical mesh at a jet Reynolds number (Re) of 10,000. At $Re = 24,000$, this ratio increases to a 100, making a cylindrical mesh highly economical. The same effect can also be achieved by a spherical mesh, as shown by Boersma et al. [4], who used it for a jet simulation

* Corresponding author.
E-mail address: kmahesh@umn.edu (K. Mahesh).

http://dx.doi.org/10.1016/j.jcp.2017.04.025
0021-9991/

with the pole placed along the streamwise direction. Since both cylindrical and spherical meshes are three dimensional extensions to a polar mesh, the issues discussed here are equally applicable to both meshes.

A cylindrical mesh, when appropriate for the problem, still has two numerical difficulties to overcome: (a) a grid singularity at the pole $r = 0$ and (b) the severe time-step limitation from the small azimuthal edges of size $O(\Delta r(r\Delta\theta))$ as we get closer to the pole. These problems are addressed in various ways depending on the method used to solve the governing equations.

Finite difference methods and pseudo-spectral methods need to explicitly address the grid singularity at $r = 0$ when the Navier–Stokes equations are solved in polar coordinates. Methods used to alleviate this problem range from directly treating singular terms to shifting the grid nodes away from the center altogether. The following set of methods directly address the singularity at $r = 0$. Griffin et al. [10] apply L'Hospital's rule to all the terms with a $1/r$ component. One-sided differencing was applied at the center and second-order accuracy was necessary to prevent spurious pressure oscillations. Freund et al. [9] solve for the center point in Cartesian coordinates to avoid the singularity. This procedure transforms variables back and forth from polar coordinates to Cartesian coordinates at the centerline. Constantinescu and Lele [7] derive a new set of equations at the pole using series expansions for the variables and find that the method produces better results than using Cartesian equations at the pole.

Another set of methods avoid the singularity by not placing a grid point at $r = 0$. Mohseni and Colonius [13] note that most methods use pole conditions at the centerline which acts as a boundary condition and reduces the accuracy of the solution. Hence, Mohseni and Colonius transform the grid from $(0, R)$ to $(-R, R)$ and avoid placing points at $r = 0$ which are instead placed at $r = -\Delta r/2$ and $r = \Delta r/2$. To remove the time-step limitation in the azimuthal direction the solution was filtered with a sharp spectral filter with a cutoff wavelength which is a function of the radial location. This grid transformation also has the benefit of applying higher order schemes to evaluate terms close to the centerline [7,11].

Formulated in what might be considered the most natural way, finite volume methods on the other hand avoid most of the complexities of the pole singularity problem. At the pole, grid metrics such as face normals can be undefined, and multivalued variables have to be correctly addressed. These are however, easily resolved by setting fluxes from the degenerate faces at the centerline to zero. The issue of restrictive time-steps, however, still needs to be addressed. Eggels et al. [8] treat all azimuthal derivatives implicitly to avoid the explicit time-step limit. All radial and axial terms, however, were explicitly integrated in time. Akselvoll and Moin [1] developed a method in which the cylindrical domain was split into two regions. Near the centerline, the azimuthal terms were treated implicitly and for cells close to the radial boundary, the radial terms were implicit. This allowed clustering in the radial direction along the wall without its associated time-step limitations. At the interface between the two regions, conditions were derived to maintain overall accuracy and avoided coupling the implicit terms in two axes.

In this note, we solve the finite volume equations on a cylindrical O-mesh but in Cartesian coordinates. We propose a finite volume strategy to address the time-step limitation in the azimuthal direction which can easily be coupled to an implicit method in the radial direction. We note that a jet or pipe flow does not require the excessive azimuthal resolution at the centerline and since this is the source of the time-step restriction, we solve this problem by merging these thin cells into larger cells in the azimuthal direction. This procedure is conservative, computationally inexpensive and can be easily implemented in an existing finite volume code. This note is organized as follows: Section 2 describes the numerical method and Section 3 gives the formulation of the cell-merging procedure. Section 4 shows the validation of the algorithm for a periodic laminar pipe, a Lamb–Oseen vortex, and a turbulent jet. Section 5 concludes the note.

2. Numerical details

2.1. Governing equations

The governing equations employed are the compressible Navier–Stokes equations. In Cartesian coordinates, the equations for non-dimensionalized density (ρ), momentum (ρu_i) and total energy (E_t) are written as follows:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_j}{\partial x_j} = 0, \tag{1}$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_j} + \frac{1}{Re}\frac{\partial \tau_{ij}}{\partial x_j}, \tag{2}$$

$$\frac{\partial \rho E_t}{\partial t} + \frac{\partial}{\partial x_j}(\rho E_t + p)\, u_j = \frac{1}{Re}\frac{\partial \tau_{ij} u_i}{\partial x_j} + \frac{1}{\gamma_r M_r^2 Re Pr}\frac{\partial}{\partial x_j}\left(\mu c_p \frac{\partial T}{\partial x_j}\right), \tag{3}$$

where the viscous stress (τ_{ij}) is:

$$\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right). \tag{4}$$

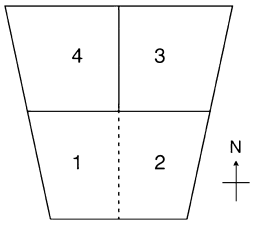

Fig. 1. Cell merging schematic.

The total energy E_t can be written as a sum of sensible energy e_t and kinetic energy. The sensible energy can be approximated to $c_v T$ for a cold jet with no temperature gradients.

$$E_t = e_t + \frac{u_i u_i}{2} = c_v T + \frac{u_i u_i}{2}. \tag{5}$$

The non-dimensional equation of state is $\rho T = \gamma_r M_r^2 pW$ where the pressure is non-dimensionalized using the compressible scaling $p = p^d/\rho_r u_r^2$. γ_r is the reference heat capacity ratio and W is the molecular weight. The medium is assumed to be a thermally and calorically perfect ideal gas with constant heat capacity.

2.2. Numerical method

The governing equations are discretized using a finite volume algorithm which is second order accurate in space and time. The data is co-located and stored at the cell centers. Note that we solve in Cartesian coordinates and compute the Cartesian velocities u, v and w. The discretized form of the equations – density, momentum, and energy, are as follows:

$$\frac{\partial \rho}{\partial t} = -\frac{1}{V_f} \sum_{faces} \rho_f v_n A_f, \tag{6}$$

$$\frac{\partial \rho u_i}{\partial t} = -\frac{1}{V_f} \sum_{faces} \left[\rho_f u_{i,f} v_n + p_f n_i - \frac{1}{Re} \tau_{ik,f} n_k \right] A_f, \tag{7}$$

$$\frac{\partial \rho E_t}{\partial t} = -\frac{1}{V_f} \sum_{faces} \left[(\rho E_t + p) v_n - \frac{1}{Re} \tau_{ik,f} u_{i,f} n_k - Q_{k,f} n_k \right] A_f. \tag{8}$$

While the above equations are general and are applicable to any finite volume grid, we focus on a structured grid with a cylindrical mesh topology. The cells are hexahedral with flat surfaces, except for the cells adjacent to the pole, which are prisms. The flux terms in the equations are split into fluxes aligned with the radial direction and non-radial directions (azimuthal and longitudinal). Radial inviscid fluxes are integrated in time with an implicit Crank–Nicholson scheme and solved with a direct block-tridiagonal line solver. Only density, momentum and energy equations are coupled together and the Jacobian is derived for the inviscid terms. The viscous terms are treated explicitly. This method was chosen to eliminate stiffness due to the acoustic time-scale which was the limiting factor in the current simulations. Linearization of energy was performed assuming constant heat capacity ratio, γ. The non-radial fluxes use the fully explicit second order Adams–Bashforth scheme.

3. Cell merging

Stiffness in the azimuthal direction arises from the very thin cells of size $O(\Delta r(r\Delta\theta))$ as we approach the pole at $r = 0$. The smallest cell is a triangle of area $(\Delta r)^2 \sin(\Delta\theta)/4$. By merging cells together in the azimuthal direction, we construct larger cells that do not impose a time step restriction as severe as the thinnest cells. When enough cells are merged to make the azimuthal spacing similar to the radial spacing, i.e. $n(r\Delta\theta) \approx \Delta r$, we effectively relax the time step restriction to depend on the radial spacing alone. In the process of merging cells, we have to ensure that fluxes from the merging process are still conservative. We describe the process using the schematic in Fig. 1. The schematic shows 4 cells with cells labeled '1' and '2' being merged.

All terms on the right-hand side of the governing equations (6)–(8) are expressed as fluxes and can be represented by the following equation 9. In two dimensions (r, θ), the fluxes from the four faces are identified by directions North, South, East and West in the subscript. North is aligned with increasing r.

$$\frac{\partial \phi}{\partial t} = -\frac{1}{V} \sum_{faces} \phi_f v_n A_f = \frac{1}{V} [F_N + F_S + F_E + F_W]. \tag{9}$$

The finite volume equations for the unmerged cells C_1 and C_2 and merged cell C_{12} as shown in the schematic can be written as follows:

$$\frac{\partial \phi_1}{\partial t} = \frac{1}{V_1}[F_{1N} + F_{1S} + F_{1E} + F_{1W}] \qquad = RHS_1,$$
$$\frac{\partial \phi_2}{\partial t} = \frac{1}{V_2}[F_{2N} + F_{2S} + F_{2E} + F_{2W}] \qquad = RHS_2,$$
$$\frac{\partial \phi_{12}}{\partial t} = \frac{1}{V_{12}}[(F_{1N} + F_{2N}) + (F_{1S} + F_{2S}) + F_{2E} + F_{1W}].$$

Since $F_{1E} = -F_{2W}$ by construction, i.e. they are fluxes of the same face from opposite directions, we observe that the right-hand side (RHS) of the merged cell can be exactly written in terms of the RHS of the constituent cells as a simple volume weighted average.

$$\frac{\partial \phi_{12}}{\partial t} = \frac{(RHS_1)V_1 + (RHS_2)V_2}{V_1 + V_2} = \frac{\sum_n (RHS_n)V_n}{\sum_n V_n}. \tag{10}$$

Having expressed the discretized equation for a merged cell, we extend the process to include the explicit time-integration in non-radial terms and implicit radial terms. A purely explicit method with cell-merging would be written as:

$$\delta\phi^t = h\frac{\sum_{fn} F_f^t}{\sum_n V_n} + R^t. \tag{11}$$

The same procedure applied to a fully implicit Crank–Nicholson time-integration gives us:

$$\delta\phi^t + \frac{\Delta t}{2}\left[\frac{\sum_{fn} F_f^{t+1}}{\sum_n V_n} + \frac{\sum_{fn} F_f^t}{\sum_n V_n}\right] = R^t. \tag{12}$$

We can now write the semi-implicit form: Crank–Nicholson in the radial direction and Adams–Bashforth in azimuthal and longitudinal directions for the unmerged cells and merged cell:

$$\delta{\phi_1}^t + \frac{\Delta t}{2}\left[\frac{F_{1N}^{t+1} + F_{1S}^{t+1}}{V_1} + \frac{F_{1N}^t + F_{1S}^t}{V_1}\right] + \frac{\Delta t}{2}\left[3\frac{F_{1E}^t + F_{1W}^t}{V_1} + \frac{F_{1E}^{t-1} + F_{1W}^{t-1}}{V_1}\right] = R_1^t,$$
$$\delta{\phi_2}^t + \frac{\Delta t}{2}\left[\frac{F_{2N}^{t+1} + F_{2S}^{t+1}}{V_2} + \frac{F_{2N}^t + F_{2S}^t}{V_2}\right] + \frac{\Delta t}{2}\left[3\frac{F_{2E}^t + F_{2W}^t}{V_2} - \frac{F_{2E}^{t-1} + F_{2W}^{t-1}}{V_2}\right] = R_2^t,$$
$$\delta{\phi_{12}}^t + \frac{\Delta t}{2}\left[\frac{(F_{1N} + F_{1S} + F_{2N} + F_{2S})^{t+1}}{V_1 + V_2} + \frac{(F_{1N} + F_{1S} + F_{2N} + F_{2S})^t}{V_1 + V_2}\right],$$
$$+\frac{\Delta t}{2}\left[3\frac{(F_{1E} + F_{1W} + F_{2E} + F_{2W})^t}{V_1 + V_2} - \frac{(F_{1E} + F_{1W} + F_{2E} + F_{2W})^{t-1}}{V_1 + V_2}\right] = \frac{R_1^t V_1 + R_2^t V_2}{V_1 + V_2}.$$

In a regular polar grid, assuming $V_1 = V_2 = V$ in the azimuthal direction, we can write for n merged cells:

$$\delta\phi^t + \frac{\Delta t}{2}\left[\frac{\sum_{NS} F_f^{t+1}}{nV} + \frac{\sum_{NS} F_f^t}{nV}\right] + \frac{\Delta t}{2}\left[3\frac{\sum_{EW} F_f^t}{nV} - \frac{\sum_{EW} F_f^{t-1}}{nV}\right] = \frac{\sum_n R_n^t}{n}.$$

Linearizing the fluxes for a coupled implicit line solve in the N-S direction:

$$\delta\phi^t + \frac{\Delta t}{2}\left[\frac{\sum_{NS} J_f^t \delta\phi_f^t A_f}{nV} + 2\frac{\sum_{NS} F_f^t}{nV}\right] + \frac{\Delta t}{2}\left[3\frac{\sum_{EW} F_f^t}{nV} - \frac{\sum_{EW} F_f^{t-1}}{nV}\right] = C(R^t).$$

In a more compact and general form, this can be written as:

$$\delta\phi^t + \frac{\Delta t}{2}\sum_{NS} C(J_f^t)\delta\phi_f^t = C(RHS_{CN}^t) + \frac{3}{2}C(RHS_{AB2}^t) - \frac{1}{2}C(RHS_{AB2}^{t-1}) + C(R^t), \tag{13}$$

where $C(x)$ is the coarsening operator defined as:

$$C(x) \equiv \frac{\sum_n (x)V_n}{\sum_n V_n}. \tag{14}$$

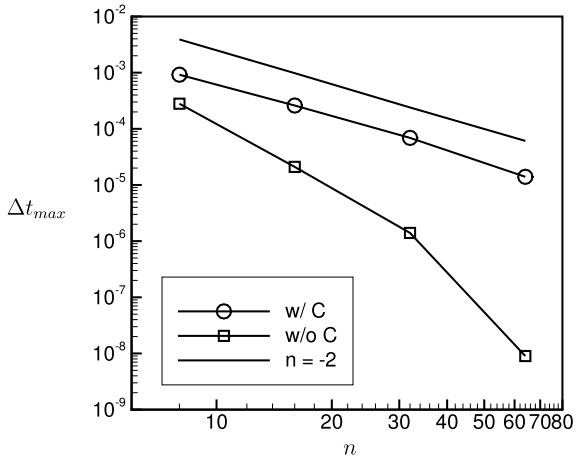

Fig. 2. Maximum time step size vs grid size. Simulations with cell merging are shown in circles and squares without. The solid line is drawn at a slope of -2.

3.0.1. Numerical implementation

The finite volume grid is structured and three-dimensional arrays are used to store all the variables for a simulation. A cell-centered data storage is used with face-centered quantities computed on the fly. In the current implementation, a regular cylindrical grid is assumed. The grid is non-uniform in the radial direction to cluster cells near walls or shear layers. The grid is uniform in the azimuthal and longitudinal directions. Cell merging is performed in the azimuthal direction based on the edge length $r\Delta\theta$. If this edge is smaller than the local radial edge of length Δr, n cells in the azimuthal direction are merged until the condition $nr\Delta\theta > \Delta r$ is met. This condition used also ensures that the aspect ratio of the merged cell is least skewed.

The process of cell merging starts with fluxes that are computed based on the cell-centered values stored on the fine grid. The Jacobians for the implicit formulation, right hand side terms and source terms are computed for all the cells. Once this is done, the coarsening operator is applied on these terms as per equation (13). This step takes the values stored in the arrays and replaces them with the coarsened value for all cells that are being merged. These coarsened values are then used to compute the change in variable over time step ${\delta\phi_{12}}^t$. Since the coarsening operator is linear, $\delta\phi^t$ can be computed for each fine grid cell and the resulting ${\delta\phi_1}^t$ and ${\delta\phi_2}^t$ can be coarsened to achieve the same result. This cell merging implementation operates on structured data and does not need unstructured representation. We also avoid resorting to complex block-structured meshes. The underlying structured data representation brings the benefit of a regular memory access, extensions to higher order flux reconstruction and as discussed in the next section: simpler parallelization with balanced loads.

3.0.2. Time step limit

We estimate the maximum time step that can be taken with and without the coarsening operation with the periodic pipe problem (section 4.1). When the limiting grid size changes from $r\Delta\theta$ to Δr the maximum time-step that can be taken should increase proportionately. Note that without cell merging, the smallest cell spacing is $\Delta r\Delta\theta/2$ in the azimuthal direction whereas with cell merging, it is Δr. The viscous time limit for an explicit method in one-dimension is $\nu\Delta t/\Delta x^2 \leq 1/2$. We observe this behavior in Fig. 2 where merging (line with circles) increases the maximum time step size by orders of magnitude compared to the case without merging (line with squares). In addition, the time step decrease is proportionate to the square of grid spacing, consistent with the viscous time limit. Without merging, the time step drops faster as the smallest grid spacing depends on both Δr and $\Delta\theta$.

Cell merging thus removes the severe time-step restriction without having to use a fully implicit scheme. This further boosts the parallel performance of the code.

3.0.3. Cost of cell-merging

The cost of cell merging is estimated by running the simulation with and without cell merging for the periodic pipe problem discussed in section 4.1. Table 1 shows the cost of the simulation in seconds per hundred time steps. This test was performed on small meshes on a single desktop processor. The time taken by merging is less than 7% of the total simulation time. Cell merging therefore adds little to the cost of the simulation while allowing us to take much larger time steps.

3.0.4. Discrete conservation of cell-merging

The cell merging step effectively add fluxes together for the supercell from its constituent cells and therefore ensures discrete conservation of cell-centered quantities. This property is demonstrated in Fig. 3 where the difference in total mass flux after the coarsening operation is shown as a function of simulation time for the Lamb–Oseen vortex problem, which

Table 1
Cost of simulations in seconds per hundred iterations, with and without the coarsening operation.

Grid	Run-time (s) w/ C()	Run-time (s) w/o C()
8×8	0.79	0.77
16×16	1.42	1.37
32×32	3.92	3.72
64×64	13.4	12.6

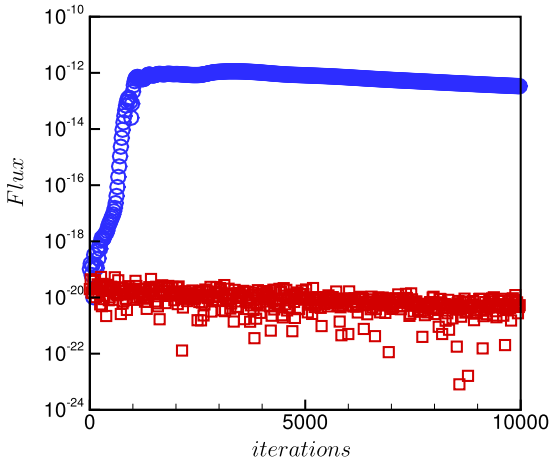

Fig. 3. Conservation error: Total mass flux in shown in blue and the error due to merging as a function of time for the Lamb–Oseen vortex simulation is shown in red.

is discussed in section (4.2). The comparison is made with the total mass flux in the domain which shows that the cell merging operation conserves to machine zero accuracy.

3.1. Parallel algorithm for merging

Large simulations on parallel computers can result in coarsened cells that span multiple processors. A parallel algorithm for coarsening is described in this section for 2^n cells in the azimuthal direction. If the azimuthal domain is split into 2^p processors and the cells are merged into 2^k coarse cells, a single coarse cell will span $2^p/2^k$ processors. Thus the coarsening operator has to be split across processors and this is achieved by successively applying the operator within a processor and then across processors:

$$C_{n/k}(x) = \frac{\sum_{2^n/2^k}(x)}{2^n/2^k} = \frac{\sum_{2^p/2^k}(\sum_{2^n/2^p}(x))}{2^p/2^k \times 2^n/2^p} = C_{p/k}(C_{n/p}(x))$$

Programmatically, this is accomplished in a Message Passing program by computing the local coarsening and storing it in an array with 2^p elements where each element would correspond to a processor. The inter-processor coarsening operator is applied next after gathering all elements using an *MPI_Allreduce* call. The final value can now be stored in each grid cell.

Scaling tests were performed on Argonne National Laboratory's super-computing facility and the results are shown in Fig. 4. The two plots correspond to strong scaling and weak scaling respectively. Strong scaling measures how much faster the program is executed upon increasing the number of processors for a given simulation. The strong scaling test shown in Fig. 4(a) measures the parallelizability in the azimuthal direction and shows good results. Weak scaling measures the time taken to execute the program with a constant number of grid elements per processor. The program is fully parallel in the longitudinal direction as well and has been run efficiently on hundreds of thousands of processors. Accounting for parallelization in all three axes, the algorithm can potentially scale up to a million or more processors. Weak scaling in the longitudinal axis in addition to the azimuthal axis scales well up to a hundred thousand processors as shown in Fig. 4(b). The largest grid tested had 1.34 *Billion* cells.

4. Validation and results

Validation is performed for three problems: (a) a periodic laminar pipe, (b) a Lamb–Oseen vortex and, (c) a turbulent cold jet. The periodic pipe has a steady solution with an axisymmetric axial velocity field. The velocity field in the $r-\theta$ plane is zero. The Lamb–Oseen vortex has a non-zero azimuthal velocity field and a decaying solution. Both these problems have analytical solutions. The turbulent cold jet is included to demonstrate applicability to highly unsteady flows while retaining robustness and accuracy.

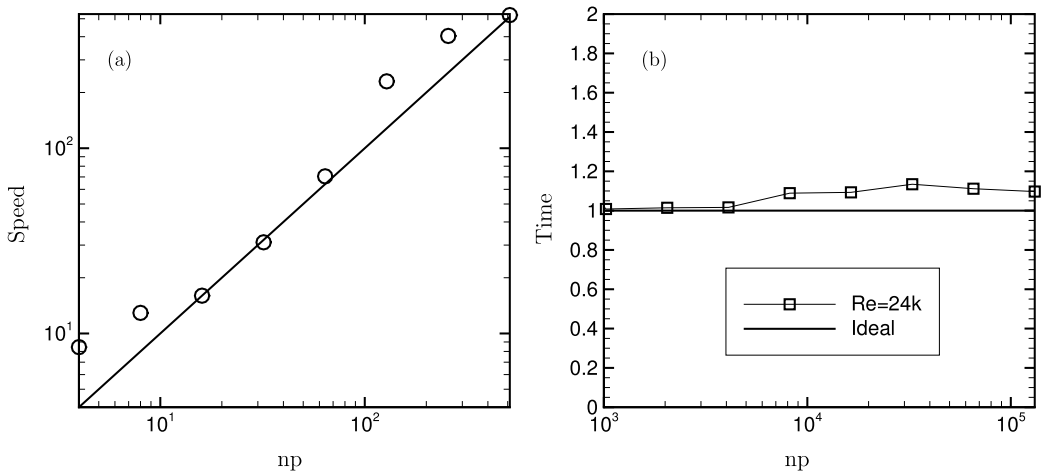

Fig. 4. Scaling results for the parallel cell merging algorithm are shown: (a) plot of normalized speed versus number of processors showing Strong scaling using a grid with 590,000 cells, (b) plot of normalized time versus number of processors showing Weak scaling with 10,200 cells per processor.

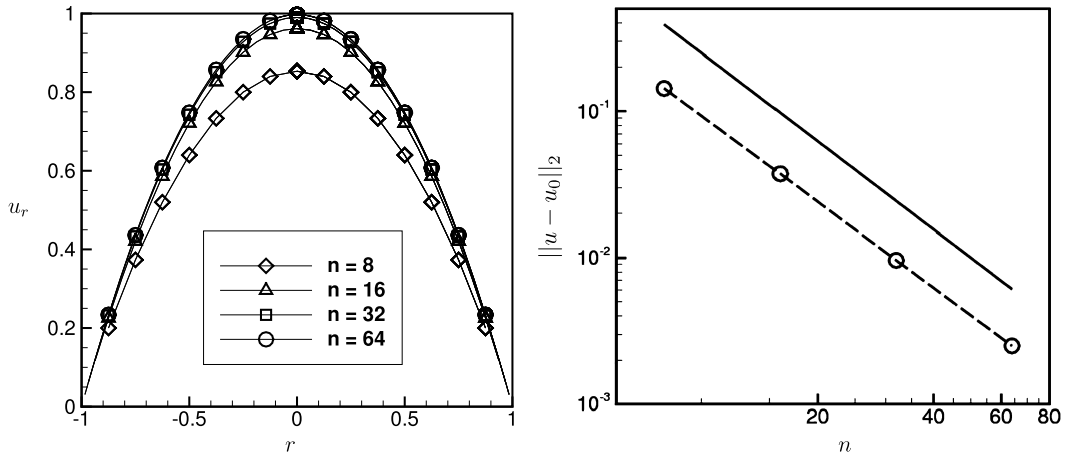

Fig. 5. (a) Streamwise velocity profile vs radial distance for different grid resolutions. (b) Log plot of L_2 error norm versus grid cells in the radial direction. The solid line indicates a slope of -2.

4.1. Periodic laminar pipe

A fully developed laminar pipe flow (Hagen–Poisueille Flow) is simulated for validation of the scheme and to confirm second order accuracy. The analytical solution for pipe flow is:

$$u(r) = \frac{Re}{4} F_b (1 - r^2), \tag{15}$$

where $u(r)$ is the streamwise velocity, Re is the Reynolds number, F_b is the body force and r the radial distance. The diameter of the pipe is denoted by D and the bulk velocity U_b. The body force is set to $F_b = 4/Re$ and $Re = 1$ for all cases tested. The grid size is varied from 8×8 to 64×64 in the radial and azimuthal directions respectively. All cases are simulated up to three flow-through time-units, $3D/U_b$, which was found to provide a converged solution. Fig. 5 shows the velocity profile from different grid resolutions and demonstrates second order spatial accuracy.

4.2. Lamb–Oseen vortex

The Lamb–Oseen vortex is another axisymmetric fluid flow problem with an analytical solution to the incompressible Navier–Stokes equations. While the radial velocity field is axisymmetric, the Cartesian velocities are not and since we solve for Cartesian velocities, this problem is a good validation test on a cylindrical mesh. The analytical solution for the Lamb–

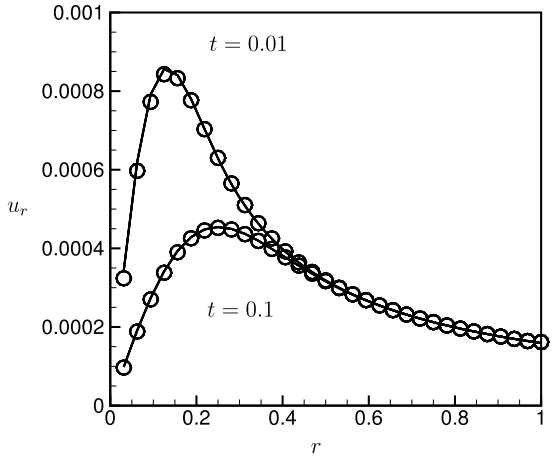

Fig. 6. Velocity vs radial distance at times $t = 0.01$ and $t = 0.1$. The circles are the numerical simulation compared with the solid lines from the analytical solution.

Oseen vortex can be written for the vorticity field, as $\omega = \frac{\Gamma_0}{4\pi\nu t} e^{-r^2/4\nu t}$. The azimuthal velocity field corresponding to the vorticity solution expressed as a function of radial distance and time can be written as follows:

$$v_\theta(r,t) = \frac{\Gamma_0}{2\pi r}\left(1 - e^{-r^2/4\nu t}\right), \tag{16}$$

where v_θ is the azimuthal velocity field, Γ_0 is the initial circulation, r is the radial distance, t is time and ν is kinematic viscosity.

The comparison between the analytical velocity profile and simulation result as a function of radial distance from the vortex center is shown in Fig. 6. The parameters used for the simulation are $\Gamma_0 = 0.001$, $\nu = 0.01$ and the initial radius at time $t = 0$ was set to $r_0 = 0.1$. The (r, θ) grid was $(64, 64)$ for this simulation. The figure shows the solution obtained at two instants, $t = 0.01$ and $t = 0.1$ from the initialization. Since the solution is a decaying velocity field in time, the comparison is made at different time instants. The numerical solution captures the decay in time and the spatial changes in the field. The numerical solution shown by circles are compared with the analytical solution plotted as solid lines. We observe good agreement with the theoretical solution.

4.3. Turbulent cold jet

The merging algorithm was developed to perform very large scale Direct Numerical Simulations (DNS) of reacting round jets. A cold jet simulation is carried out on a fairly coarse grid to demonstrate the robustness of the numerical method. The jet is simulated at a Reynolds number of $2,400$ on a domain of $20D \times 2\pi \times 45D$ where D is the jet diameter. The grid had $(80, 64, 450)$ cells in (r, θ, z) directions with a total of $2.3M$ elements. The inflow was specified using a hyperbolic tangent function, which closely approximates a top-hat profile but has smooth edges and was initially perturbed with random velocities of magnitude 0.1%. The cells in the radial directions were clustered near the center and in the axial direction the cell density was gradually decreased downstream. The Mach number of the jet was set to 0.2 to minimize compressibility effects.

The results obtained are shown in Fig. 7. U denotes jet velocity in the streamwise direction, U_o is the velocity at the inlet and U_c is the centerline velocity. The centerline velocity U_c follows a $1/x$ scaling as seen in Fig. 7(a) where the normalized inverse centerline velocity U_o/U_c increases linearly with the slope of dashed red line. Fig. 7(b) shows the self-similarity of the jet where the axial velocity cross-section profiles are taken from $18D$ to $30D$ downstream of the inlet. The velocity ratio U/U_c is plotted against the self-similar variable $\eta = r/(z - z_0)$ for all the profiles taken and can be seen to collapse onto each other. We obtain a velocity decay constant of $B_u = 5.6$ and a spreading rate of $S = 0.99$ which compares well with experimental observation of Hussein et al. [12] and Panchapakesan & Lumley [14]. The potential core was also observed to close $11D$ downstream of the inlet which matches well with the simulation results of Babu & Mahesh [3] and Boersma et al. [4]. This simulation shows that the cell merging algorithm can handle a turbulent flow-field and produce results that compare well with existing data while offering an increased time-step and lower grid cell count compared to a Cartesian mesh.

5. Conclusion

A conservative finite volume formulation to prevent the restrictive time-step limitation from a polar grid has been presented. Extremely thin grid cells close to the centerline are merged together in the azimuthal direction to form thicker

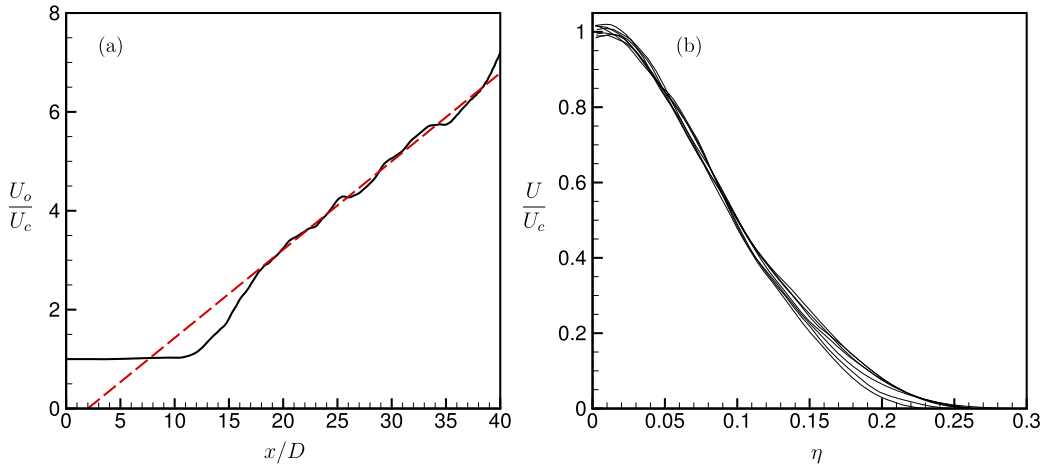

Fig. 7. (a) Inverse centerline velocity vs axial distance. (b) Self-similarity of axial velocity cross-section with profiles taken between 18D and 30D downstream.

cells with an aspect ratio close to one. The cell width is thus effectively increased which removes the excessive time-step limitation due to the CFL limit. Cell merging in the azimuthal direction allows implicit time integration in the radial direction without coupling the azimuthal terms. This allows for better parallel scalability and usage on large clusters for DNS studies of jets.

Acknowledgements

Computer time for the simulations was provided by the Minnesota Supercomputing Institute (MSI) and Argonne Leadership Computing Facility (ALCF).

References

[1] K. Akselvoll, P. Moin, An efficient method for temporal integration of the Navier–Stokes equations in confined axisymmetric geometries, J. Comput. Phys. 125 (1996) 454–463.
[2] R. Asaithambi, S. Muppidi, K. Mahesh, A Numerical Method for DNS of Turbulent Reacting Flows Using Complex Chemistry, AIAA Paper, 2012.
[3] P.C. Babu, K. Mahesh, Upstream entrainment in numerical simulations of spatially evolving round jets, Phys. Fluids 16 (2004) 3699–3705.
[4] B. Boersma, G. Brethouwer, F. Nieuwstadt, A numerical investigation on the effect of the inflow conditions on the self-similar region of a round jet, Phys. Fluids 10 (1998) 899–909.
[5] C. Bogey, C. Bailly, Turbulence and energy budget in a self-preserving round jet: direct evaluation using large eddy simulation, J. Fluid Mech. 627 (2009) 129–160.
[6] R. Cabra, T. Myhrvold, J.Y. Chen, R.W. Dibble, A.N. Karpetis, R.S. Barlow, Simultaneous laser Raman–Rayleigh-LIF measurements and numerical modeling results of a lifted turbulent H_2/N_2 jet flame in a vitiated coflow, Proc. Combust. Inst. 29 (2002) 1881–1888.
[7] G.S. Constantinescu, S. Lele, A highly accurate technique for the treatment of flow equations at the polar axis in cylindrical coordinates using series expansions, J. Comput. Phys. 183 (2002) 165–186.
[8] J. Eggels, F. Unger, M. Weiss, J. Westerweel, R. Adrian, R. Friedrich, F. Nieuwstadt, Fully developed turbulent pipe flow: a comparison between direct numerical simulation and experiment, J. Fluid Mech. 268 (1994) 175–210.
[9] J. Freund, S. Lele, P. Moin, Direct Simulation of a Supersonic Round Turbulent Shear Layer, AIAA paper, 1997, 97–0760.
[10] M.D. Griffin, E. Jones, J.D. Anderson, A computational fluid dynamic technique valid at the centerline for non-axisymmetric problems in cylindrical coordinates, J. Comput. Phys. 30 (1979) 352–360.
[11] R. Hixon, S.H. Shih, R. Mankbadi, Numerical treatment of cylindrical coordinate centerline singularities, Int. J. Comput. Fluid Dyn. 15 (2001) 251–263.
[12] H.J. Hussein, S.P. Capp, W.K. George, Velocity measurements in a high-Reynolds-number, momentum-conserving, axisymmetric, turbulent jet, J. Fluid Mech. 258 (1994) 31–75.
[13] K. Mohseni, T. Colonius, Numerical treatment of polar coordinate singularities, J. Comput. Phys. 157 (2000) 787–795.
[14] N.R. Panchapakesan, J.L. Lumley, Turbulence measurements in axisymmetric jets of air and helium. Part 1. Air jet, J. Fluid Mech. 246 (1993) 197–223.
[15] M. Yokokawa, K. Itakura, A. Uno, T. Ishihara, Y. Kaneda, 16.4-Tflops direct numerical simulation of turbulence by a Fourier spectral method on the Earth Simulator, in: Supercomputing, ACM/IEEE 2002 Conference, IEEE, 2002, p. 50.
[16] C.S. Yoo, R. Sankaran, J.H. Chen, Three-dimensional direct numerical simulation of a turbulent lifted hydrogen jet flame in heated coflow: flame stabilization and structure, J. Fluid Mech. 640 (2009) 453–481.