ICAS 2021 SHANGHAI
International Council of the Aeronautical Sciences
32nd Congress of the International Council of the Aeronautical Sciences
September 6-10, 2021
Pudong Shangri-La, Shanghai, China

AN ACCELERATION METHODOLOGY BASED ON DISTURBANCE REGION UPDATE METHOD FOR STEADY COMPRESSIBLE FLOWS

Hu Shuyao[1], Jiang Chongwen[1], Gao Zhenxun[1], Lee Chun-Hian[1]

[1]National Laboratory for Computational Fluid Dynamics, School of Aeronautic Science and Engineering, Beihang University, 100191 Beijing, China

Abstract

The disturbance region update method (DRUM) provides a framework for accelerating the simulation of compressible flows, solving the governing equations in dynamic computational domains (DCDs). The DCDs merely include disturbed cells with non-convergent solutions, and treats the viscous flows in a dynamic zonal way, to eliminate the worthless computational effort in the conventional global-update solution process as much as possible. In this paper, we propose some improvements on the algorithms of updating DCDs. Numerical test demonstrates that, benefiting from the reduction in the computational effort per iteration and the decrease in the total number of iterations, DRUM could accomplish remarkable convergence speedup for solving steady compressible flows; DRUM can be conjunction with spatial parallelism, capable of achieving a total speedup much higher than the linear speedup of the conventional global-update method.

Keywords: computational fluid dynamics; acceleration techniques; dynamic computational methods; zonal methods;

1. Introduction

NASA's Computational Fluid Dynamics (CFD) Vision 2030 Study [1] indicated that CFD has fundamentally changed the aerospace design process, and the performance in terms of numerical efficiency and solution accuracy of CFD is of critical importance. The development of acceleration techniques for CFD is still an active field since there is an ever-lasting demand for efficient numerical tools for the flow simulation. At present, the time-marching method is the most popular one for solving steady compressible flows. It has unified mathematical characteristics in time for both subsonic and supersonic flows, and thus attains good applicability in the full speed range of compressible flows. The time-marching method solves unsteady governing equations in a preassigned computational domain (PCD), starting from an arbitrarily assumed initial flowfield, and advances in time until the iterative update of flowfield can be neglected.

Acceleration techniques for the time-marching method can be classified into three categories, which are concentrated on the schemes, on the implementation and on the grid, as illustrated in Figure 1. The first category attempts to enhance the convergence per iteration by improving spatial and temporal schemes, such that the total number of iterations required for convergence could be lessened. Successful examples are the multigrid methodology [2], the large time step schemes [3], the implicit time-marching schemes [4], and so forth. The second category relies on parallelization. In CFD, parallelization based on domain decomposition is appealing and efficient [5]. Parallelization assigns the computational work to multiple threads (processes), and thus reduces the sequential execution time. The third category is drawn based on limiting the number of grid cells to decrease the computational effort per iteration, e.g., the adaptive mesh refinement (AMR) technique [6]. Besides, we proposed a method named the disturbance region update method (DRUM) that also belongs to the third category.

In the time-marching produce, the operations to discretize the spatial and temporal derivatives are termed the residual estimation and the time integration, respectively. In a conventional way, the two operations are conducted in a static PCD, and update all grid cells belonging to the PCD in every iteration for easy implementation, i.e., by a global-update method (GUM). Therefore, these two operations become the most two time-consuming ones, taking almost 99% of the total execution time. Disadvantages in GUM are twofold. Firstly, the selection of the PCD would play a crucial role in the numerical result but be highly dependent on experience, in which an inadequate PCD may lead to a failure of computations while a redundant one would waste computational resources. Secondly, a static PCD cannot take the dynamic characteristics of the flowfield evolution into account, thus resulting in the worthless computational efforts.

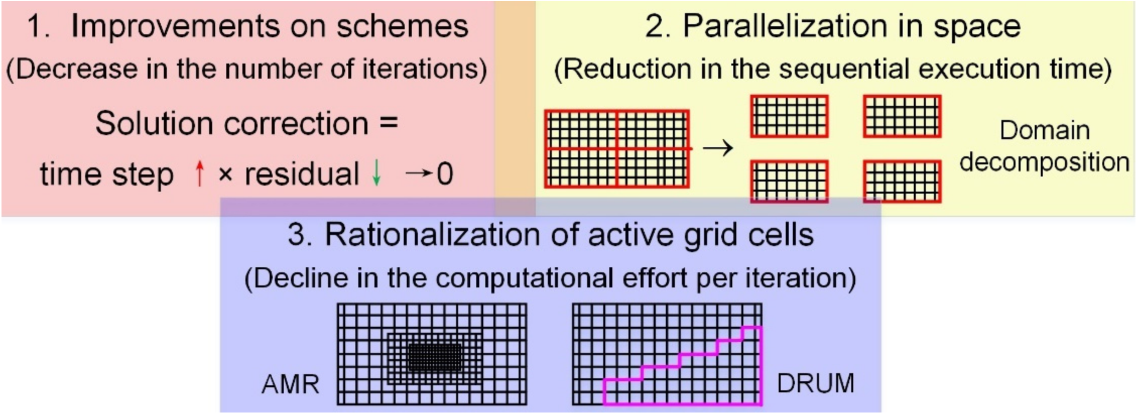
1. Improvements on schemes
(Decrease in the number of iterations)
Solution correction =
time step ↑ × residual ↓ →0
2. Parallelization in space
(Reduction in the sequential execution time)
→
Domain decomposition
3. Rationalization of active grid cells
(Decline in the computational effort per iteration)
AMR
DRUM

Figure 1 Principles of acceleration techniques for the time-marching method

To circumvent the problem, we propose an acceleration methodology named the disturbance region update method (DRUM), The idea of DRUM is inspired in accordance with the observation on solution processes of the time-marching method, which hold the following features: (1) disturbances are produced from where the governing equations cannot be satisfied, e.g., for the solution initialized from the freestream conditions, discontinuities in flow properties develop from the wall boundaries. (2) disturbances would propagate gradually to the surrounding flow as the pseudo time evolves, and flow properties in the region that have not been disturbed yet would maintain the initial states. (3) in compressible flows, the upstream region would not converge slower than its downstream since the flow speed is orders of magnitude equal to or greater than the speed of sound. (4) for viscous flows, viscous effects are solely dominant in a finite zone near the surface, no matter whether the flow is attached or separated. Therefore, a rational computational domain used by the time-marching method should be built from where disturbances are generated, extend with the propagation of disturbances and then contract and move downstream along with the convergence of solution. And the viscous-dominated region characterized by large velocity and temperature gradients should merely be part of the computational domain.

Obviously, compared with GUM, DRUM solving the governing equations in DCDs equips the capacity of reducing the worthless computational effort as much as possible, thus accelerating the convergence per second. DRUM defined two kinds of DCDs, named the advective and the viscous. The advective DCD contains all non-convergent disturbed cells in the PCD, while the viscous one covers part of the advective DCD where viscous effects are dominant. The extension and contraction of DCDs are implemented by two operations, namely, the insertion and the removal. The insertion operation is responsible for adding disturbed cells into the two DCDs separately. The removal operation is in charge of deleting converged cells from the two DCDs and misjudged viscous-dominated cells from the viscous DCD.

Noted that although DRUM shares the same acceleration principle with the AMR technique, jet they implement the principle in different ways. The AMR technique is to control the size and location of grid cells, making them refine, coarsen and deform. In contrast, grid cells remain unchanged in DRUM. DRUM distinguishes the active and inactive cells, thereby adapting the computational domain dynamically.

2. Time-Marching Method in Finite-Volume Framework

In DRUM, the inviscid, laminar and turbulent flows are governed by the Euler, the Navier-Stokes (N-S) and the density-weighted Reynolds-averages Navier-Stokes (RANS) equations, respectively. The integral form of these three governing equations can be unified as

$$\frac{\mathrm{d}}{\mathrm{d}t}\iiint_{\Omega}\boldsymbol{W}\mathrm{d}\Omega+\oint_{\partial\Omega}(\boldsymbol{F}_{\mathrm{c}}-\boldsymbol{F}_{\mathrm{v}})\mathrm{d}\boldsymbol{S}=\iiint_{\Omega}\boldsymbol{Q}\mathrm{d}\Omega \tag{1}$$

where the notations of Reynolds and Favre averaging are omitted. t represents the time variable; a structured control volume is denoted with Ω, the boundary of Ω with $\partial\Omega$, and a surface element on $\partial\Omega$ with dS. Neglecting body forces and heat sources, the conservative variables $\boldsymbol{W}$, convective fluxes $\boldsymbol{F}_{\mathrm{c}}$, viscous fluxes $\boldsymbol{F}_{\mathrm{v}}$ and the source term $\boldsymbol{Q}$ in Cartesian coordinates have the following components

$$\boldsymbol{W}=\begin{bmatrix}\rho\\ \rho u_i\\ \rho E\\ \boldsymbol{W}_{\mathrm{T}}\end{bmatrix},\quad \boldsymbol{F}_{\mathrm{c}}=\begin{bmatrix}\rho u_j\\ \rho u_i u_j+p\delta_{ij}\\ (\rho E+p)u_j\\ \boldsymbol{F}_{\mathrm{c,T}}\end{bmatrix},\quad \boldsymbol{F}_{\mathrm{v}}=\begin{bmatrix}0\\ \tau_{ij}\\ \tau_{ij}u_j+\chi\left(\partial T/\partial x_j\right)\\ \boldsymbol{F}_{\mathrm{v,T}}\end{bmatrix},\quad \boldsymbol{Q}=\begin{bmatrix}0\\0\\0\\ \boldsymbol{Q}_{\mathrm{T}}\end{bmatrix} \tag{2}$$

Here, ρ, p, T, E stand for the density, pressure, temperature and the specific total energy, respectively, which satisfy the equation of state for calorically perfect gas. x_i denotes a Cartesian coordinate, u_i a velocity component, τ_{ij} a component of the stress tensor (i, j =1, 2, 3). The thermal conductivity coefficient χ is a function of the dynamic viscosity μ. The subscript T denotes the turbulence model. In this work, Menter's k-ω shear stress transport (SST) turbulent model is adopted, in which the working variables $\boldsymbol{W}_{\mathrm{T}}$, the convective fluxes $\boldsymbol{F}_{\mathrm{c,T}}$, viscous fluxes $\boldsymbol{F}_{\mathrm{v,T}}$ and the source term $\boldsymbol{Q}_{\mathrm{T}}$ read

$$\mu_{\mathrm{T}}=\frac{\rho k}{\max\left(\omega \quad f_2\left\|\nabla\times\boldsymbol{u}\right\|_2/a_1\right)},\quad \boldsymbol{W}_{\mathrm{T}}=\begin{bmatrix}\rho k\\ \rho\omega\end{bmatrix},\quad \boldsymbol{F}_{\mathrm{c,T}}=\begin{bmatrix}\rho k u_j\\ \rho\omega u_j\end{bmatrix},\quad \boldsymbol{F}_{\mathrm{v,T}}=\begin{bmatrix}\left(\mu_{\mathrm{L}}+\sigma_k\mu_{\mathrm{T}}^{\mathrm{SST}}\right)\left(\partial k/\partial x_j\right)\\ \left(\mu_{\mathrm{L}}+\sigma_\omega\mu_{\mathrm{T}}^{\mathrm{SST}}\right)\left(\partial\omega/\partial x_j\right)\end{bmatrix},$$

$$\boldsymbol{Q}_{\mathrm{T}}^{\mathrm{SST}}=\begin{bmatrix}\tau_{ij}^{\mathrm{F}}S_{ij}-\beta^*\rho k\omega\\ c_\omega\rho\tau_{ij}^{\mathrm{F}}S_{ij}/\mu_{\mathrm{T}}^{\mathrm{SST}}-\beta\rho\omega^2+2(1-f_1)\rho\sigma_{\omega2}\left(\partial k/\partial x_j\right)\left(\partial\omega/\partial x_j\right)/\omega\end{bmatrix} \tag{3}$$

where k, ω denote the turbulent kinetic energy and the specific dissipation rate, respectively. The closure coefficients β and β^* are modified by the Wilcox compressibility correction [9]. Details of other closure coefficients a_1, c_ω, f_1, f_2, σ_k, σ_ω, $\sigma_{\omega2}$ can be found in [10].

If all terms involving viscosity or related to the turbulent model are dropped, then Eq. (1) degrades as the Euler equations, suitable for the inviscid flow. If all terms related to the turbulent model are omitted and μ is equal to the molecular viscosity μ_{L}, then Eq. (1) becomes the N-S equations governing the laminar flow, in which μ_{L} obeys Sutherland's law. If all terms are kept and μ is the sum of μ_{L} and the eddy viscosity μ_{T}, the Eq. (1) is the RANS equations closed with the Boussinesq eddy-viscosity models.

DRUM is implemented in a cell-centered finite volume solver on structured grids. Assuming that the conservative variables as well as the source term are constant inside a control volume, and the surface integral can be approximated by a sum of the fluxes crossing the faces of the control volume, Eq. (1) can be discretized as

$$|\Omega|\frac{\mathrm{d}\boldsymbol{W}}{\mathrm{d}t}=-\boldsymbol{R}=\underbrace{-\sum_{m=1}^{N_{\mathrm{f}}}\boldsymbol{F}_{\mathrm{c},m}\cdot\boldsymbol{n}_m\Delta S_m}_{\text{inviscid term}}+\underbrace{\sum_{m=1}^{N_{\mathrm{f}}}\boldsymbol{F}_{\mathrm{v},m}\cdot\boldsymbol{n}_m\Delta S_m}_{\text{viscous term}}+|\Omega|\boldsymbol{Q} \tag{4}$$

where the right-hand side terms are collectively termed the residual, denoted by $\boldsymbol{R}$; $|\Omega|$ and ΔS denote the volume and the face area of Ω, respectively, and N_{f} stands for the face number of Ω, equal to 4 in two-dimensional (2-D) cases and 6 in three-dimensional (3-D) cases. In general, the discretization of the left-hand side and the right-hand side terms are treated separately, namely, the method of lines. Their corresponding operations are the residual estimation and the time integration. The residual estimation is to evaluate $\boldsymbol{R}$ with spatial discretization schemes, while the time integration is to approximate d$\boldsymbol{W}$/dt by time-marching schemes, estimating the correction of the solution $\Delta\boldsymbol{W}$. The purpose of a time-marching method for steady compressible flow is to make every grid cell in the PCD satisfy $||\Delta\boldsymbol{W}||<\varepsilon_{\mathrm{c}}$ by means of an iterative process.

The time integration is accomplished based on the implicit lower-upper symmetric Gauss-Seidel (LU-SGS) scheme [4], in which Eq. (4) can be discretized further as

$$\left[\frac{|\Omega|}{\Delta t}+\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{W}}\right]\Delta \boldsymbol{W}^{(n)}=-\boldsymbol{R}^{(n)} \qquad (5)$$
$$\boldsymbol{W}^{(n+1)}=\boldsymbol{W}^{(n)}+\Delta \boldsymbol{W}^{(n)}$$

in which Δt is the time step and the superscript (n) denotes the present step. Let each cell of a structured grid be uniquely identified by the indices I, J, K. The LU-SGS scheme splits the solution of Eq. (5) into two steps, a forward and a backward sweep, i.e.,

$$\boldsymbol{D}\Delta \boldsymbol{W}^{*}=-\boldsymbol{R}^{(n)}-\boldsymbol{L}\Delta \boldsymbol{W}^{*}$$
$$\Delta \boldsymbol{W}^{(n)}=\boldsymbol{D}^{-1}\left[\boldsymbol{D}\Delta \boldsymbol{W}^{*}-\boldsymbol{U}\Delta \boldsymbol{W}^{(n)}\right] \qquad (6)$$

with

$$\boldsymbol{D}=\left[\frac{|\Omega|}{\Delta t}+\sum_{m=I,J,K}\left(\omega\Lambda_{\mathrm{c}}^{m}+2\Lambda_{\mathrm{v}}^{m}\right)\right]\boldsymbol{I}-|\Omega|\frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{W}}$$
$$\boldsymbol{L}=\frac{1}{2}\sum_{m=I,J,K}\left[\Delta \boldsymbol{F}_{\mathrm{c}}^{(n)}\Delta S+\left(\omega\Lambda_{\mathrm{c}}+2\Lambda_{\mathrm{v}}\right)\Delta \boldsymbol{W}\right]_{m-1} \qquad (7)$$
$$\boldsymbol{U}=\frac{1}{2}\sum_{m=I,J,K}\left[\Delta \boldsymbol{F}_{\mathrm{c}}^{(n)}\Delta S+\left(\omega\Lambda_{\mathrm{c}}+2\Lambda_{\mathrm{v}}\right)\Delta \boldsymbol{W}\right]_{m+1}$$

where ω denotes an overrelaxation factor; Λ_c and Λ_v are the spectral radii of the convective- and the viscous-flux Jacobians in a certain grid direction, respectively; $\Delta \boldsymbol{F}_{\mathrm{c}}^{(n)}=\boldsymbol{F}_{\mathrm{c}}^{(n+1)}-\boldsymbol{F}_{\mathrm{c}}^{(n)}$. For clarity, the cell indices I, J, K are omitted where not required. Here, the forward sweep is carried out from the minimum to the maximum of the sum of grid indexes, while the backward sweep is in a converse order.

3. Disturbance Region Update Method

3.1 Basic Principles

Taking a supersonic flow at Mach 2.5 over a 2-D wedge as an example, Figure 2 illustrates the principle of DCD evolution in DRUM, in which N stands for the number of iterations, Δ***W*** represents the correction of the solution, η_c, η_v denote the cell-number ration of the advective and the viscous DCDs to the PCD, respectively. When the flowfield is initialized from the freestream conditions, wall conditions are enforced on the cells adjacent to the wall and thus both DCDs are created based on the wall, as shown in Figure 2 a). At the early stage, the insertion operation gradually adds cells into the two DCDs. At the 330th step, both DCDs reach their peak, where the advective one covers the region near and downstream of the oblique shock while the viscous one shaped like a long strip lies next to the wall, as displayed in Figure 2 b). Then, the removal operation would exclude converged cells in compressible flow from both DCDs, starting from the upstream, like the cases in Figure 2 c) and d). At the end of the solution, only the near-wall cells stay in the DCDs and all cells in the PCD achieve the steady-state solution.

3.2 Procedure of DRUM

The execution procedure of a finite-volume solver with DRUM for steady compressible flow is carried out as follows:

(1) Loading input files, including the user-defined grid, boundary conditions and computational settings.

(2) Initializing the flowfield, which can be performed from freestream conditions or based on a specific flowfield.

(3) **Creating the advective and the viscous DCDs** in terms of the initialization way of flowfield. If the flowfield is initialized from freestream conditions, then both DCDs are built based on the wall conditions. If the flowfield is initialized based on a specific flowfield, then the advective DCD is the set of cells with flow properties different from freestream conditions while the viscous DCD is the viscous-dominated part of its advective counterpart.

(4) Imposing boundary conditions. Two layers of dummy cells are adopted in this work.

(5) Estimating the inviscid terms of the residual (performed in the advective DCD).

(6) Estimating the viscous terms of the residual (performed in the viscous DCD).

(7) Time integration (performed in the advective DCD).

(8) **Extending and contracting the advective DCD in grid blocks**.

(9) **Extending and contracting the viscous DCD in grid block**.

(10) **Extending the advective and the viscous DCDs between grid blocks**.

(11) Judging whether the convergence criteria are satisfied. If so, continuing the step (12); if not, repeating the steps (4)-(10).

(12) Outputting results.

Among these steps, steps in bold characters are exclusively owned by DRUM. The residual estimation is divided into two parts, i.e., the steps (5) and (6), which are performed in the advective and the viscous DCDs, respectively. The other time-consuming operation, the time integration, is also performed in the advective DCD. Since the advective DCD may merely be part of the PCD and the viscous DCD may only be part of the advective one, DRUM could achieve savings in terms of the computational cost, as compared with the conventional GUM.

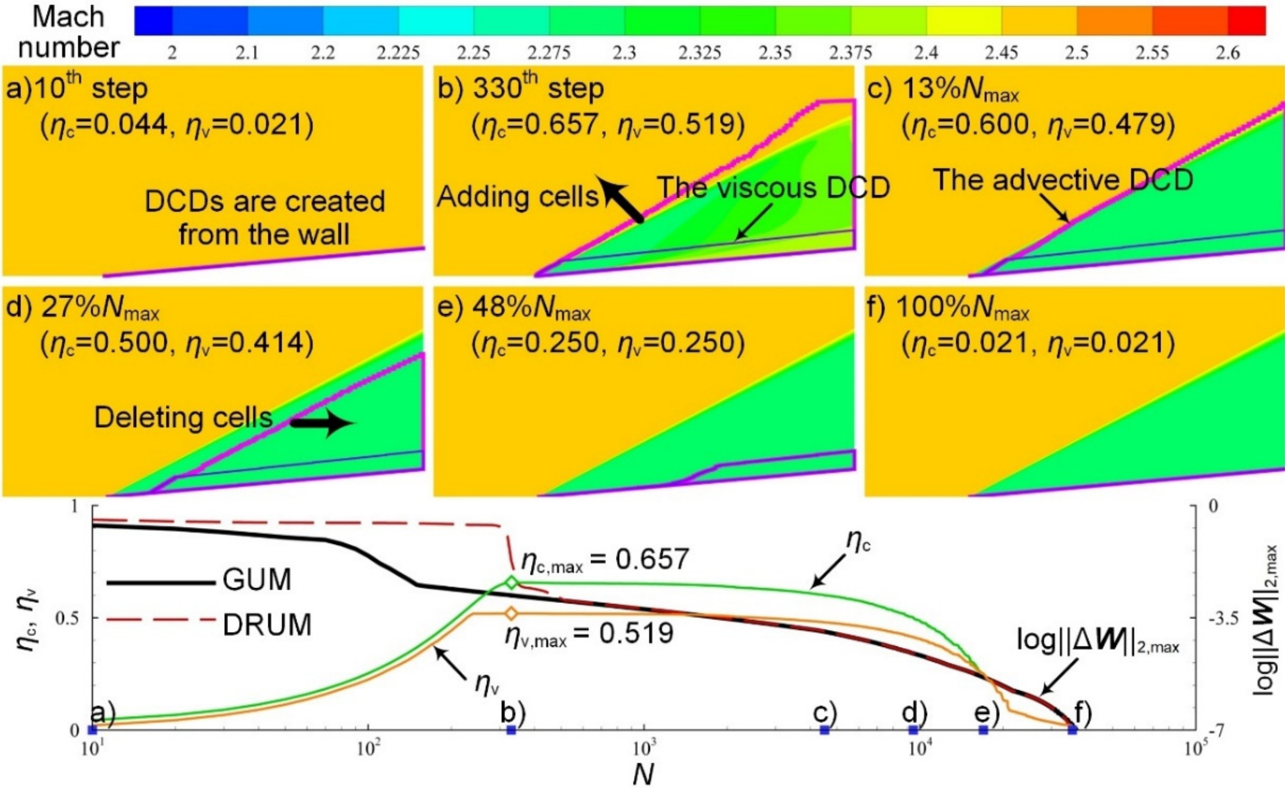

Figure 2 Examples of the update sequences of DRUM

3.3 Creation of DCDs

3.3.1 Flowfield Initialized from Freestream Conditions.

In this case, disturbances are generated because of the existence of wall, and would propagate from the wall. Therefore, both DCDs are built based on wall boundaries. Let $l_{ini,a}$ and $l_{ini,v}$ be the layer number of cells added into the initial advective and the viscous DCDs, respectively. In our previous work, both $l_{ini,a}$ and $l_{ini,v}$ are set to 1 in all cases. However, it is found that small values of $l_{ini,a}$ may make the convergence history of DRUM at the initial stage different from that of GUM. The value has no influence on a solution with thousands of iterations, but may slightly delay the convergence for a solution with hundreds of iterations. In order to avoid this detrimental effect, we increase $l_{ini,a}$ from 1 to 10 in this work, so that the maximum norm of $||\Delta \boldsymbol{W}||_{max}$ at the first step calculated by DRUM can be the same with that of GUM. Contrary to $l_{ini,a}$, the convergence is insensitive to the value of $l_{ini,v}$. Thus, $l_{ini,v}$ remains 1 for less computational work, i.e., the cells immediately next to the wall are regarded as the initial elements of the viscous DCD.

3.3.2 Flowfield Initialized Based on A Specific Flowfield.

An advective DCD should be the part of the PCD containing all disturbed cells with non-convergent solutions. In our previous work, we determined the initial elements of an advective DCD by two sub-steps: 1) sifting disturbed cells from all the cells of the PCD; 2) sifting converged cells out the set of disturbed cells. However, it is found that the step 2) is unnecessary, since converged cells could be removed from the DCDs immediately at the first step. This step has no beneficial effect on the solution process, instead of bringing extra computational work. Hence, we omit this step in this work. Disturbed cells are characterized by having flow properties different from freestream conditions. Hence, disturbed cells can be identified by

$$\left\| \boldsymbol{W} - \boldsymbol{W}_{\infty} \right\| > \varepsilon_{\mathrm{c}} \tag{8}$$

with the subscript ∞ being the freestream. All cells satisfying Eq. (8) are added into the initial advective DCD. For an initial viscous DCD, it should be the viscous-dominated part of its advective counterpart. The way to identify a viscous-dominated cell in the creation and the update of the viscous DCD is the same, which will be introduced in Section 3.5.

3.4 Extension of the Advective DCD

After the update of the solution, extending the advective DCD in terms of the updated information is a top priority so as to allow the propagation of disturbances. Considering a particular cell (I, J, K) adjacent to the boundary of an advective DCD, the insertion operation executes the following two sub-steps in sequence: (1) judging whether disturbances affect the cell; (2) Determining potential disturbed cells among the surrounding cells.

For the sub-step (1), a disturbed cell can be identified in terms of its correction that cannot be neglected, which can be expressed as

$$\left\| \Delta \boldsymbol{W}^{(n)} \right\| \Big/ \left\| \Delta \boldsymbol{W}^{(1)} \right\|_{\max} > \varepsilon_{\mathrm{a}} \tag{9}$$

where $||\Delta \boldsymbol{W}^{(n)}||$ corresponds to the value of the cell (I, J, K) at the present step and $||\Delta \boldsymbol{W}^{(1)}||_{\max}$ denotes the maximum norm of $\Delta \boldsymbol{W}$ at the first step, and ε_{a} is a preassigned advective insertion threshold, recommended to 10^{-5}. $||\Delta \boldsymbol{W}^{(1)}||_{\max}$ is a new factor introduced in this work to normalize $||\Delta \boldsymbol{W}^{(n)}||$ so as to avoid the adjustment of ε_{a} for different cases. If the cell (I, J, K) satisfies Eq. (9), the insertion operation continues working for it; if not, the operation will skip the cell and continue to judge other cells adjacent to the boundary of the advective DCD.

For the sub-step (2), assume that there is at most one layer of surrounding cells that would be disturbed at the next time step. Except the one adjacent to the PCD's boundaries, a disturbed cell owns two adjacent cells in each grid direction, e.g., the cell (I±1, J, K) in the I direction. This sub-step is aimed at determining which adjacent cell must be added into the advective DCD. The maximum wave speed in any direction is the sum of a and the velocity component along the specified direction. Let $\boldsymbol{q}$ be a unit direction vector, and then the condition that disturbances can propagate along a specified direction can be given by

$$\boldsymbol{u} \cdot \boldsymbol{q} + a > 0 \tag{10}$$

For a subsonic flow, sound waves move faster than the flow so that disturbances could propagate in all directions away from the disturbance source. That is to say, all the next neighboring cells of the disturbed cell satisfy Eq. (10) by nature and thus spDRUM must insert all of them into the advective DCD. However, for a supersonic flow, Eq. (10) becomes the key to limiting the advective DCD to cover the upstream of the leading-edge shock. Here, the choice of $\boldsymbol{q}$ is of significance. In our previous work, $\boldsymbol{q}$ is defined as a unit vector pointing from the centroid of the cell (I, J, K) towards that of a surrounding cell, but Eq. (10) with it may fail in certain cases.

Figure 3 sketches a 2-D case to illustrate the reason why Eq.(10) may fail. For the disturbed cell (I, J), $\boldsymbol{q}$ is directed from the centroid of the cell (I, J) towards that of the cell (I, J+1) to measure whether disturbances transmit along the positive J-direction. If $u_i q_i < -a$, then the cell (I, J+1) would not be defined as a potential disturbed cell. If all cells at the J index satisfy $u_i q_i < -a$, then the advective DCD would never extend to the row at the J+1 index. In [8], we proposed to search more surrounding cells like the cell (*I*+1, *J*+1) to address this failure, since a disturbed cell should propagate disturbances to its downstream indeed. However, this way may be time-consuming to search a lot of surrounding cells when $u_i q_i$ is much lower than $-a$.

In this work, we provide a new definition of $\boldsymbol{q}$ to encounter this failure. $\boldsymbol{q}$ is defined as the unit vector directed from the centroid of the disturbed cell towards one of its vertices. If $\boldsymbol{q}$ pointing from the centroid of the cell (*I*, *J*) towards the vertex (*i*+1, *j*+1) satisfies Eq. (10), then the cells (*I*, *J*+1) and (*I*+1, *J*) would be regarded as potential disturbed cells and inserted into the advective DCD. Although the new definition of $\boldsymbol{q}$ allows more cells join the advective DCD, undisturbed cells will be removed immediately at the next time step. Hence, the new definition not only has no harm to the computational efficiency but also avoids the time-consuming search.

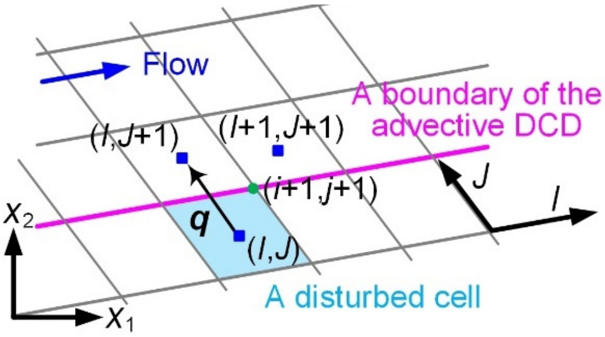

Figure 3 Schematic of the failure and the definition of $\boldsymbol{q}$

3.5 Contraction of the Advective DCD

After the extension of the advective DCD, the removal operation would try to remove cells from the rows without new-added cells. Considering a particular cell (*I*, *J*, *K*) adjacent to the boundary of an advective DCD, the removal operation measures whether to delete it from the advective DCD or not in terms of the following four conditions: (1) the cell and its surrounding cells converge to the steady state; (2) the cell is located at the most upstream; (3) the cell has little influence on the update of the solution; (4) neighboring cells have little influence on the update of the cell. If all of the four conditions are satisfied, then the cell (*I*, *J*, *K*) would be removed from the advective DCD. Note that the viscous DCD must be included in its advective counterpart. Hence, the cell (*I*, *J*, *K*) would be excluded from the viscous DCD as well if it is also a viscous-dominated cell.

Regarding the condition (1), it is similar to the convergence condition, i.e.,

$$\left\|\Delta \boldsymbol{W}^{(n)}\right\| \Big/ \left\|\Delta \boldsymbol{W}^{(1)}\right\|_{\max} < \varepsilon_{\mathrm{d}} \tag{11}$$

with ε_{d} being a preassigned removal threshold, recommended to $\varepsilon_{\mathrm{d}} \leq 10^{-7}$. A smaller ε_{d} achieves better accuracy but also results in loss of efficiency. Eq. (16) would be performed in the cell (*I*, *J*, *K*) and two layers of its surrounding cells.

Regarding the condition (2), the removal operation is constructed based on the characteristic of the time-marching solution process that the upstream region of the PCD may converge to the steady state faster than its downstream counterpart. Hence, a new-deleted cell must lie at the most upstream. Let $\boldsymbol{x}_0$ and $\boldsymbol{u}_0$ be the centroid's coordinates and velocity of the cell (*I*, *J*, *K*), respectively, $\boldsymbol{x}_1$ be the centroid's coordinates of its neighboring cell. If the cell is the most upstream one, then all of its neighboring cells that remain in the advective DCD must be located downstream, which should satisfy

$$\arcsin \frac{\boldsymbol{u} \cdot \left(\boldsymbol{x}_1 - \boldsymbol{x}_0\right)}{\left\|\boldsymbol{u}\right\|_2 \left\|\boldsymbol{x}_1 - \boldsymbol{x}_0\right\|_2} < \theta_{\mathrm{d}} \tag{12}$$

with θ_{d} being a preassigned angle tolerance of the upstream cell. θ_{d} is a tolerance aimed at avoiding a misjudgment between lying upstream and at the same flow station. Here, different values of θ_{d} are recommended for supersonic and subsonic flows. A supersonic flow is governed by the hyperbolic nature, and its advective DCD should contract from the upstream to the downstream strictly. Therefore, a small range of 5°≤θ_{d}≤15° is preferred for supersonic flows. In contrast, the outer boundary of a subsonic PCD must stay far away from the wall in all directions to eliminate the influence of a finite boundary domain. It is found that the region that lies downstream but far away from the wall has little influence to the convergence and accuracy of the solution. Therefore, a large range of $40° \leq \theta_d \leq 50°$ is recommended for supersonic flows.

Regarding the condition (3), it stems from addressing the phenomena that low-speed cells lying the upstream and the downstream may converge at the same pace, especially for the viscous cells. Given that the wall is a dominated source of disturbances, we define a condition in terms of the distance to the nearest wall d_w, to measure whether the cell to be deleted continues having an effect on the solution. Assume that in a subsonic or viscous flow, disturbances generated by the wall decay with the increase of d_w. Let $d_{w,c}$ be the maximum d_w of the cells satisfying Eq. (11). If a cell no longer has an influence on the update of the solution, then it must have a d_w satisfying

$$d_w > d_{w,c} \tag{13}$$

Here, d_w is a parameter prepared and stored in the flowfield initialization; $d_{w,c}$ is updated at every iteration along with the time integration, and would increasingly decrease with the convergence of the solution.

Regarding the condition (4), it can be skipped for a cell in the supersonic inviscid flow. This is because that the hyperbolic nature of the flow stipulates that the flow at any given point depends merely upon the properties at other points upstream of it. Thus, a supersonic inviscid cell can be immune to the update of others by nature if it lies at the most upstream.

Differentiating the residual defined in Eq. (4) with respect to $\boldsymbol{W}$ results in

$$\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{W}} = \sum_{m=1}^{N_f} \left(\frac{\partial \boldsymbol{F}_{c,m}}{\partial \boldsymbol{W}} - \frac{\partial \boldsymbol{F}_{v,m}}{\partial \boldsymbol{W}} \right) \cdot \boldsymbol{n}_m \Delta S_m - |\Omega| \frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{W}} \tag{14}$$

Here, $\boldsymbol{F}_c$ can be split into a positive and a negative part, i.e., $\boldsymbol{F}_c = \boldsymbol{F}_c^+ + \boldsymbol{F}_c^-$. The derivatives $\partial \boldsymbol{F}_c \pm / \partial \boldsymbol{W}$ can be approximated by $(\partial \boldsymbol{F}_c \pm / \partial \boldsymbol{W}) \cdot \boldsymbol{n} \Delta S = (\boldsymbol{A}_c \Delta S \pm \Lambda_c \boldsymbol{I})/2$, with $\boldsymbol{A}_c$ being the convective flux Jacobian and $\boldsymbol{I}$ being the identity matrix. The derivatives $\partial \boldsymbol{F}_v / \partial \boldsymbol{W}$ can be approximated by the viscous spectral radii Λ_v, satisfying $(\partial \boldsymbol{F}_v / \partial \boldsymbol{W}) \cdot \boldsymbol{n} \Delta S \approx \Lambda_v$. The effect of the next neighboring cells in the I-direction on $\boldsymbol{R}_{I,J,K}$ can be given as

$$\begin{aligned} \Delta \boldsymbol{R}^I = \frac{1}{2} \Big[& \Delta \boldsymbol{F}_{c,I+1} \cdot \boldsymbol{n}_{I+1/2} \Delta S_{I+1/2} - (\Lambda_{c,I+1} + 2\Lambda_{v,I+1}) \Delta \boldsymbol{W}_{I+1} \\ & + \Delta \boldsymbol{F}_{c,I-1} \cdot \boldsymbol{n}_{I-1/2} \Delta S_{I-1/2} + (\Lambda_{c,I-1} + 2\Lambda_{v,I-1}) \Delta \boldsymbol{W}_{I-1} \Big] \end{aligned} \tag{15}$$

For clarity, the cell indices I, J, K are omitted where not required. Similar expressions of Eq. (15) hold for the J- and K-direction. With the Euler approximation, i.e., $\Delta \boldsymbol{W} = \Delta \boldsymbol{R} \cdot C_{\mathrm{CFL}} \cdot \Delta t / |\Omega|$, the condition representing the cell (I, J, K) immune to the cells in the advective DCD can be described as

$$\left\| \Delta \boldsymbol{W}^{(n)} + \sum_{m=I,J,K} \Delta \boldsymbol{R}^m \cdot \frac{\Delta t}{|\Omega|} \cdot C_{\mathrm{CFL}} \right\| < \varepsilon_d \tag{16}$$

Here, C_{CFL} is the CFL number, which is new-added in this work to take the factor of different CFL numbers into account. Eqs. (15) and (16) also hold for the inviscid transonic case where terms associated with Λ_v can be ignored.

3.6 Extension of the Viscous DCD

The adjustment of the viscous DCD follows the update of its advective counterpart. Consider a particular cell (I, J, K) lying next to the boundary of a viscous DCD, the insertion operation to manipulate the viscous DCD follows only one criterion, i.e., measuring whether the cell is viscous-dominated. If so, all neighboring cells adjacent to the cell (I, J, K) would be inserted into the viscous DCD. This is because the elliptic behavior of the governing equations' viscous terms allows both the upstream and downstream propagation of disturbances via viscosity and thermal conduction.

Regarding the measure of viscous effects, a criterion in terms of $\boldsymbol{F}_c$, $\boldsymbol{F}_v$ is the most intuitive way. Unfortunately, the cost in extra computation or storage of these two vectors is high. $\boldsymbol{F}_c$, $\boldsymbol{F}_v$ include the information about the mass flow, momentum flow and the energy flow. To save the computational cost, a criterion only related to the mass flow of the inviscid and the viscous disturbances across a cell was employed in our previous work. The largest wave speeds of the inviscid and the viscous disturbances can be evaluated by the eigenvalues of the Jabobians of $\boldsymbol{F}_c$ and $\boldsymbol{F}_v$. Hence, the mass flow of the inviscid and the viscous disturbances across the cell can be expressed in terms of Λ_c and Λ_v. As a result, we defined a variable, $\Psi=\Lambda_v/\Lambda_c$, and a preassigned viscous insertion threshold, $\varepsilon_{a,v}$, as well as a criterion, $\Psi>\varepsilon_{a,v}$, to identify the viscous-dominated cells. However, in our previous work, $\varepsilon_{a,v}$ had to be adjusted for different cases, which degrades the universality of DRUM.

In order to adopt a constant $\varepsilon_{a,v}$ in all cases, in this work, we adopt a new criteria with self-adaptive thresholds to assess whether viscous effects can be ignored in a cell. It is known that the total enthalpy h_0 is constant in a steady inviscid flowfield, even across a shock wave. Hence, a viscous-dominated cell can be identified in terms of its h_0 that is different from the value of the freestream, i.e.,

$$\left|1-h_{0,\infty}/h_0\right| > \varepsilon_{\text{a,v}} \tag{17}$$

Here, the setting of $\varepsilon_{a,v}$ is similar to the definition of a boundary-layer thickness. It is recommended that $\varepsilon_{a,v}=5\times10^{-3}$, i.e., $h_0=1.005h_{0,\infty}$ is defined as the edge of the boundary layer. Although Eq. (17) is a perfect condition to distinguish the viscous-dominated cells in a real flowfield, h_0 is incorrect during the time-marching solution process of a steady flow, especially for the early stage. Therefore, Eq. (17) cannot be imbedded in DRUM in a direct way.

It is found that conditions combining flow properties with grid parameters would be more stable and robust for a numerical algorithm. Hence, the variable Ψ that includes flow properties, thermal properties and grid sizes is a proper option. However, as mentioned above, the threshold of Ψ is difficult to preassign for various cases. Besides, numerical results in [8] indicate that a condition in terms of Ψ may misjudge a small cell as a viscous-dominated one, even if it is very far away from the wall and should be inviscid.

To avoid the disadvantages of our previous work and take advantage of the physical nature expressed by Eq. (17), we would combine a new condition in terms of d_w with the original condition in terms of Ψ to limit the maximum d_w of the viscous DCD. Let Ψ_{vis} and $d_{w,v}$ be the thresholds of Ψ and d_w, respectively, in which Ψ_{vis} represents the maximum of Ψ on the cells satisfying Eq. (17) and $d_{w,v}$ corresponds to the minimum of d_w on these cells. Similar to the search of $d_{w,c}$, the search for Ψ_{vis} and $d_{w,v}$ are also embedded in the time integration to avoid extra gird-cell traversal. The new criteria can be given by

$$\Psi > \varphi_{\text{vis}}\Psi_{\text{vis}} \;\cap\; d_{\text{w}} < \varphi_{\text{vis}} d_{\text{w,v}} \tag{18}$$

with φ_{vis} being a scaling factor to avoid the instability stemmed from an inadequate viscous region

$$\varphi_{\text{vis}} = \max\left[1.0,\;\; \log\left(\left\|\Delta\boldsymbol{W}\right\|_{\max}\right)_{\min} \Big/ \log\left\|\Delta\boldsymbol{W}^{(n)}\right\|_{\max}\right] \tag{19}$$

where $||\Delta\boldsymbol{W}^{(n)}||_{max}$ denotes the maximum of $||\Delta\boldsymbol{W}||$ on all cells at the present step, and $(||\Delta\boldsymbol{W}||_{max})_{min}$ is the minimum of $||\Delta\boldsymbol{W}||_{max}$ during the convergence history. As can be seen, in the new-defined criteria, Ψ_{vis} and $d_{w,v}$ are self-adaptive based on Eq. (17) at every iteration; φ_{vis} would be greater than 1.0 if the convergence history shows a growth trend, i.e., $||\Delta\boldsymbol{W}^{(n)}||_{max}$ is greater than $(||\Delta\boldsymbol{W}||_{max})_{min}$, and thus the viscous DCD would be further enlarged; the only preassigned threshold $\varepsilon_{a,v}$ has clear physical implication so that one value could be suitable for various cases.

3.7 Contraction of the Viscous DCD

In [8], the contraction of viscous DCD could only be driven by that of the advective DCD rather than by itself. In an attempt to improve the flexibility of the viscous DCD, an extra step manipulating the contraction of the viscous DCD is integrated into spDRUM. In this step, the removal operation is responsible for sifting out the cells misjudged as viscous-dominated ones, and follows the opposite conditions defined in the extension of the viscous DCD, i.e.,

$$\begin{gathered}\left|1-h_{0,\infty}/h_0\right| < \varepsilon_{\text{d,v}} \\ \Psi < \varphi_{\text{vis}}\Psi_{\text{vis}} \;\cup\; d_{\text{w}} > \varphi_{\text{vis}} d_{\text{w,v}}\end{gathered} \tag{20}$$

3.8 A Summary of Improvements in DRUM

For better understand, we summarize the improved algorithms that are different from their original versions, as tabulated in Table 1.

Table 1 Improvements on algorithms in DRUM

Operations		Improvements	Purpose
Creation of the advective DCD	from freestream	Increasing $l_{\mathrm{ini,a}}$ from 1 to 10	Avoiding an adverse effect on the convergence
	from a flowfield	Omitting the evaluation of $\|\|\boldsymbol{R}\|\|$	Decreasing the computational effort
Extension of the advective DCD in blocks		Introducing $\|\|\Delta\boldsymbol{W}^{(1)}\|\|_{\max}$ in Eq. (9)	Applying one value of ε_{a} for all cases
		Defining a new direction vector in Eq. (10)	Avoiding the failure of the original
Contraction of the advective DCD		Introducing $\|\|\Delta\boldsymbol{W}^{(1)}\|\|_{\max}$ in Eq. (11)	Applying one value of ε_{d} for all cases
		A new inequality in terms of θ_{d} (12)	Improving the computational efficiency
		Replacing the original Mach-number limit with a new criterion in terms of d_{w} (Eq. (13))	Making the algorithm more rational
		Adding the factor of C_{CFL} in Eq. (16)	Avoiding the difference caused by C_{CFL}
Extension of the viscous DCD in blocks		Introducing a new definition of $\varepsilon_{\mathrm{a,v}}$ and new criteria in terms of h_0, d_{w} (Eqs. (18))	Applying one value of $\varepsilon_{\mathrm{a,v}}$ for all cases Defining a more rational viscous-dominated region
Contraction of the viscous DCD		A new step in spDRUM	Defining a more rational viscous-dominated region

4. Validation and Discussions

The test case considers a supersonic inviscid flow at Mach 6 over a 3-D wedge whose upper and lower wedge angles are both 6°. The preassigned grid with 6.2×10^4 cells consists of 18 blocks, including typical grid topologies, e.g., the Y-, C- and H-grid topology. Solutions solved by DRUM with 6 threads are displayed in Figure 4. The advective DCD is built initially on the wall boundary, and extends to the surrounding flowfield as disturbances propagate, as shown in the case at the 10th step. When the cell number of the advective DCD reaches its peak, the region near the leading edge of the wedge has been removed from the DCD, as shown in the case at the 85th step. Even though the preassigned grid is divided into 18 blocks, the insertion operation could add all disturbed cells into the advective DCD while the removal operation could delete converged cells, starting from the upstream. Solution in the regions with large flow gradients is more difficult to approach the steady state. Hence, the part of the advective DCD near the wall contracts faster than that in the vicinity of the shock, and one near the low-pressure side surface of the wedge contracts much faster than that near the upper and the lower surfaces.

Numerical results attained by spDRUM and the shock-fitting method [11] at a constant z plane are compared in right-bottom panel of Figure 4. As can be seen, the shock is well resolved and the results of two methods agree well. Therefore, the capacity of simulating compressible flow for DRUM is validated.

Figures 5 and 6 display cell-number evolutions of the advective DCDs and convergence histories, respectively. As compared in Figures 5, spatial parallelism has little effect on the advective DCD. The maximum of η_{c} equal to 0.594 means that there is at most 59.4% of the preassigned cells updated at one iteration, and thus DRUM achieves a nearly 17% reduction in the maximum memory requirements. Figure 6 demonstrates that spDRUM has no harm for the convergence of the time-marching method, and it could accomplish remarkable convergence speed.

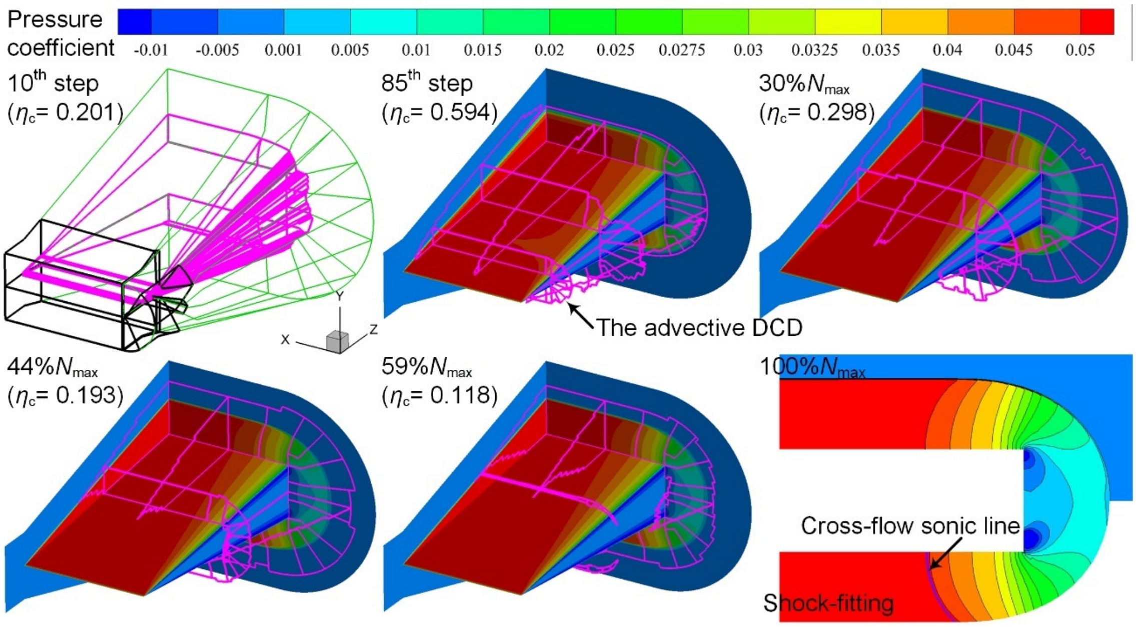

Figure 4 Solutions of the 3-D supersonic wedge (6 threads)

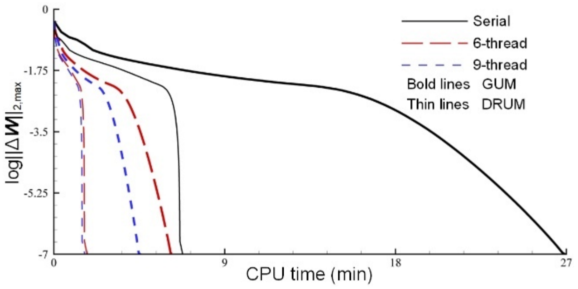

Figure 5 DCD evolutions for the 3-D supersonic wedge

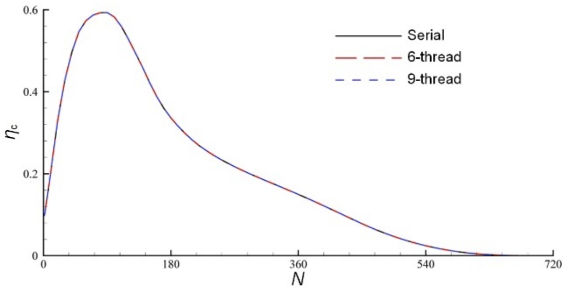

Figure 6 Convergence histories for the 3-D supersonic wedge

Let the parallelism speedup be the CPU-time ratio of serial and parallel cases cost by the same method, the DRUM speedup be the CPU-time ratio of GUM and DRUM performed with the same number of CPUs, $\Delta C_L/C_L$ and $\Delta C_D/C_D$ denote relative lift- and drag-coefficient discrepancies between GUM and DRUM performed with the same number of CPUs, respectively. Table 2 lists the accuracy and efficiency of DRUM for the 3-D supersonic wedge. The results demonstrate that spatial parallelism has no detrimental effect on the accuracy of DRUM. For all cases, $\Delta C_L/C_L$ and $\Delta C_D/C_D$ are both on the order of 10^{-6}. Therefore, numerical results provided by spDRUM are valid. Comparison between the CPU time cost by the serial GUM and DRUM indicates that the speedup of DRUM is almost equivalent to that of 6-thread parallelism. Owing to the load balancing, the speedup ratio of DRUM would decrease with the increase of the thread number. However, the total speedup of spDRUM for both 6- and 9-thread cases is above twice and even trice as high as the linear speedup of GUM with the same number of threads.

Table 2 performance comparison of GUM and DRUM for the 3-D supersonic wedge

Methods	Number of threads	Parallelism speedup	DRUM speedup	Total speedup	$\Delta C_L/C_L$	$\Delta C_D/C_D$
GUM	1	--	--	--	--	--
	6	4.346	--	--	--	--
	9	5.997	--	--	--	--
DRUM	1	--	4.166	--	1.63×10^{-6}	3.57×10^{-6}
	6	4.106	3.935	17.105	1.63×10^{-6}	3.57×10^{-6}
	9	4.394	3.053	18.305	1.63×10^{-6}	3.57×10^{-6}

5. Conclusion

The present work introduces some improvements on the algorithms of updating dynamic computational domains under the framework of the disturbance region update method, which make

DRUM more robust, efficient and universal. Numerical test demonstrates that spDRUM could achieve considerable speedup for compressible flow simulation, and also may accomplish savings in terms of the maximum memory requirements. The total speedup of DRUM conjunction with spatial parallelism would be much higher than the linear speedup provided by the global update method with the same thread number. It is proven that spatial parallelism has no harm to the accuracy of DRUM.

6. Acknowledgements

This work was supported by grants from the fellowship of China Postdoctoral Science Foundation (No. 2020M680286), and the National Natural Science Foundation of China (No. 11972061, No. U20B2006).

7. Contact Author Email Address

Corresponding author: Jiang Chongwen; mailto: cwjiang@buaa.edu.cn.

8. Copyright Statement

The authors confirm that they hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS proceedings or as individual off-prints from the proceedings.

References

[1] Slotnick J, Khodadoust A, Alonso J, et al. CFD vision 2030 study: a path to revolutionary computational aerosciences. NASA/CR-2014-218178, 2014.

[2] Jameson A, Yoon S, Multigrid Solution of the Euler Equations Using Implicit Schemes, *AIAA Journal,* Vol. 24, No. 11, pp 1737-1743, 1986.

[3] Qian Z, Lee C, On Large Time Step TVD Scheme for Hypersonic Conservation Laws and Its Efficiency Evaluation, Journal of Computational Physics, Vol. 231, pp 7415-7430, 2012.

[4] Jameson A, Yoon S, Lower-Upper Implicit Schemes with Multiple Grids for the Euler Equations, *AIAA Journal*. Vol. 25, No. 7, pp 929-935, 1987.

[5] Blazek J, *Computational Fluid Dynamics: Principles and Applications*, 2015, 3rd Edi., Elsevier.

[6] Zhang S, Gain A, Norato J, Adaptive Mesh Refinement for Topology Optimization with Discrete Geometric Components, *Computer Methods Applied Mechanics and Engineering*, Vol. 364, pp 112930, 2020.

[7] Hu S, Jiang C, Gao Z, et al. Disturbance Region Update Method for Steady Compressible Flows, *Computer Physics Communications*, Vol. 229, pp 68-86, 2018.

[8] Hu S., Jiang C, Gao Z, et al. Zonal Disturbance Region Update Method for Steady Compressible Viscous Flows, *Computer Physics Communications*, Vol. 244, pp 97-116, 2019.

[9] Brown J, Turbulence Model Validation for Hypersonic Flows, AIAA 2002-3308, 2002.

[10]Menter F, Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications, *AIAA Journal*. Vol. 32, No. 8, pp 1598-1605, 1994.

[11]Hu S, Jiang C, Gao Z, et al. Combined-Wedge Waverider for Airframe-Propulsion Integration, *AIAA Journal*, Vol. 56, No. 8, pp 3348-3352, 2018.