International Journal of Aeronautical and Space Sciences (2023) 24:1136–1146
https://doi.org/10.1007/s42405-023-00587-3

IJASS International Journal of Aeronautical and Space Sciences

ORIGINAL PAPER


# An Implicit Multicloud Convergence Accelerator for Compressible Flows

Jaesang Rhee[1] · Junseok Oh[1] · Taeyoon Kung[1] · Kyuhong Kim[1,2]

Received: 28 November 2022 / Revised: 27 February 2023 / Accepted: 7 March 2023 / Published online: 1 April 2023


**Abstract**
In this study, we developed an implicit multicloud method using geometric conservative meshless discretization. The Geometric Conservation Least Squares Method (GC-LSM) was used for meshless discretization which allowed lower–upper symmetric Gauss–Seidel (LU-SGS) time integration to be implemented in a multicloud context. The LU-SGS algorithm for meshless discretization was implemented in the same framework as the unstructured finite volume method owing to the properties of the GC-LSM. The developed method was tested for three-dimensional inviscid flows on unstructured grids. The explicit 4th-order Runge–Kutta method and the implicit LU-SGS algorithm were compared to demonstrate the effect of the proposed implicit multicloud method. Furthermore, the restriction functions were modified for three-dimensional space. The results highlight the capability of the implicit multicloud method combined with the original multicloud coarsening procedure, which is straightforward and mesh-transparent, by only applying GC-LSM for discretization.

**Keywords** Multigrid · Meshless · Implicit · Unstructured

## 1 Introduction

Multigrid methods are the most widely used techniques for accelerating the convergence rates of iterative methods. The first multigrid method was formulated for solving Poisson equation on structured grids [1]. Subsequently, multigrid methods have been improved then the techniques were applied to three-dimensional hyperbolic equations on unstructured grids [2].

Unstructured type grid is the most widely used grid type thus unstructured multigrid methods have been mainly studied. For unstructured grids, it is difficult to form coarse level grids. Therefore, a variety of approaches for forming coarse level grids have been suggested, such as retriangulation [3] or agglomeration [4–6].

In practical computational fluid dynamics(CFD) problems, grid agglomeration is the preferred method for managing unstructured grids owing to the automation process. Even for grid agglomeration, meshing steps, such as edge fusing, are accompanied by the formation of a coarse grid. To bypass the re-meshing process, the multicloud method was proposed by Katz [7]. The multicloud method uses meshless discretization on coarse level domains; thus, no re-meshing step is required to form coarse level domains. Without the re-meshing step, the coarsening procedure can be simple and storage-efficient. In the absence of a re-meshing step at the coarse level, the multicloud method has been demonstrated to be as quick as traditional multigrid methods [7,8].

Although it is obvious that the implicit time integration scheme is significantly quicker than the explicit time integration scheme, previous multicloud study was been presented based on an explicit time integration scheme [7]. Because the application of an implicit time integration scheme to a meshless method had rarely been studied, the presentation of an implicit multicloud was beyond the scope of the initial multicloud paper [7].

✉ Kyuhong Kim
aerocfd1@snu.ac.kr

Jaesang Rhee
jsrhee@snu.ac.kr

Junseok Oh
youra998@snu.ac.kr

Taeyoon Kung
ty115@snu.ac.kr

[1] Department of Aerospace Engineering, Seoul National University, Seoul 08826, Republic of Korea

[2] The Institute of Advanced Aerospace Technology, Seoul National University, Seoul 08826, Republic of Korea

In this study, we introduced the LU-SGS [9] algorithm (abbreviated as "LU-SGS" in what follows) for unstructured grids [10] as a means of achieving multicloud time integration. Generally, LU-SGS for unstructured grids cannot be directly implemented for meshless discretization owing to its nonconservative properties. Despite formulating LU-SGS for the meshless method, we used the least squares method with the geometric conservation law (the GC-LSM) [11] in this study. The geometrically conservative property of the GC-LSM allows time integration to be performed implicitly.

Because the explicit multicloud method has already been compared with the conventional multigrid method [7], a comparison between the implicit and explicit methods is presented in this study. Detailed comparisons with other implicit unstructured multigrid methods are beyond the scope of this study, because numerous sets of implicit schemes and coarsening strategies exist [12,13].

The remainder of this paper is structured as follows. First, the meshless discretization method is described as the basis for numerically analyzing the three-dimensional Euler equations based on the GC-LSM. Second, the coarsening procedure and multicloud operator are introduced. Next, the LU-SGS method with GC-LSM is described. Finally, the implicit multicloud results are compared with the single grid implicit and explicit results.

## 2 Meshless Discretization

### 2.1 Least Squares Method

In this study, a Taylor series was used to estimate the derivatives of a trial function $\phi$. A trial function $\phi$ at point $\boldsymbol{r}_i$ can be approximated by its value at a neighboring point at $\boldsymbol{r}_0$ using the Taylor series expansion:

$$\hat{\phi}\left(\boldsymbol{r}_i\right) \approx \phi\left(\boldsymbol{r}_0\right) + \Delta\boldsymbol{r}_{0i} \cdot \nabla\phi\left(\boldsymbol{r}_0\right), \tag{1}$$

where $\Delta\boldsymbol{r}_{0i} = \boldsymbol{r}_i - \boldsymbol{r}_0$ and Eq. (1) may be recast as

$$\Delta\phi_{0i} \approx \boldsymbol{p}^{\mathrm{T}}(\Delta\boldsymbol{r}_{0i}) \cdot \nabla\phi(\boldsymbol{r}_0), \tag{2}$$

where $\Delta\phi_{0i} = \hat{\phi}(\boldsymbol{r}_i) - \phi(\boldsymbol{r}_0)$ and $\boldsymbol{p}$ is a three-dimensional monomial basis function:

$$\boldsymbol{p}(\boldsymbol{r}) = [x\ y\ z]^{\mathrm{T}}. \tag{3}$$

To estimate the gradient of $\phi(\boldsymbol{r}_0)$, the least squares problem is established using its nearest points (See Fig. 1):

$$J = \sum_i^N \omega_{0i} \left[\boldsymbol{p}^{\mathrm{T}}(\Delta\boldsymbol{r}_{0i}) \cdot \nabla\hat{\phi}(\boldsymbol{r}_0) - \Delta\phi_{0i}\right]^2, \tag{4}$$

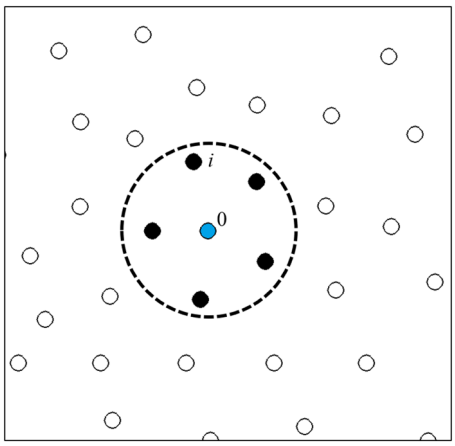



**Fig. 1** Illustration of local point cloud

where $\omega_{0i} = 1/|\Delta\boldsymbol{r}_{0i}|$ and $N$ is the number of neighboring points of the point at $\boldsymbol{r}_0$, as shown in Fig. 1.

$\nabla\hat{\phi}(\boldsymbol{r}_0)$, the estimation of the gradient of $\phi(\boldsymbol{r}_0)$, may be obtained by finding the coefficients that minimize the function $J$, expressed as

$$\frac{\partial J}{\partial\nabla\hat{\phi}(\boldsymbol{r}_0)} = 0. \tag{5}$$

Equation (5) can be written as

$$\sum_i^N \omega_{0i}\,\boldsymbol{p}\left(\Delta\boldsymbol{r}_{0i}\right) \cdot \boldsymbol{p}^{\mathrm{T}}\left(\Delta\boldsymbol{r}_{0i}\right)\nabla\hat{\phi}\left(\boldsymbol{r}_0\right) = \sum_i^N \omega_{0i}\,\boldsymbol{p}\left(\Delta\boldsymbol{r}_{0i}\right)\Delta\phi_{0i}. \tag{6}$$

Equation(6) may be simplified as

$$\boldsymbol{S}\nabla\hat{\phi}(\boldsymbol{r}_0) = \boldsymbol{T}\Delta\phi_{0i}, \tag{7}$$

where

$$\boldsymbol{S} = \begin{bmatrix} \sum_i^N \omega_{0i}\Delta x_{0i}^2 & \sum_i^N \omega_{0i}\Delta x_{0i}\Delta y_{0i} & \sum_i^N \omega_{0i}\Delta x_{0i}\Delta z_{0i} \\ \sum_i^N \omega_{0i}\Delta y_{0i}\Delta x_{0i} & \sum_i^N \omega_{0i}\Delta y_{0i}^2 & \sum_i^N \omega_{0i}\Delta y_{0i}\Delta z_{0i} \\ \sum_i^N \omega_{0i}\Delta z_{0i}\Delta x_{0i} & \sum_i^N \omega_{0i}\Delta z_{0i}\Delta y_{0i} & \sum_i^N \omega_{0i}\Delta z_{0i}^2 \end{bmatrix}, \tag{8}$$

$$\boldsymbol{T} = \left[\omega_{01}\Delta\boldsymbol{r}_{01}\ \omega_{02}\Delta\boldsymbol{r}_{02}\ \ldots\ \omega_{0N}\Delta\boldsymbol{r}_{0N}\right]. \tag{9}$$

By Eqs. (8) and (9), the estimation of the gradient can be expressed as:

$$\nabla\hat{\phi}(\boldsymbol{r}_0) = \boldsymbol{S}^{-1}\boldsymbol{T}\Delta\phi_{0i}, \tag{10}$$

where

$$\boldsymbol{S}^{-1}\boldsymbol{T} = \begin{bmatrix} a_{01} & a_{02} & \cdots & a_{0N} \\ b_{01} & b_{02} & \cdots & b_{0N} \\ c_{01} & c_{02} & \cdots & c_{0N} \end{bmatrix}. \tag{11}$$

Thus, the estimation of the partial derivatives of function $\phi$ at point $\boldsymbol{r}_0$ may be expressed as

$$\frac{\partial \phi(\boldsymbol{r}_0)}{\partial x} \approx \sum_{i}^{N} a_{0i}\Delta\phi_{0i}, \tag{12}$$

$$\frac{\partial \phi(\boldsymbol{r}_0)}{\partial y} \approx \sum_{i}^{N} b_{0i}\Delta\phi_{0i}, \tag{13}$$

$$\frac{\partial \phi(\boldsymbol{r}_0)}{\partial z} \approx \sum_{i}^{N} c_{0i}\Delta\phi_{0i}. \tag{14}$$

## 2.2 Geometric Conservative Least Squares Method

The geometric conservation law and first-order consistency with respect to meshless coefficients may be expressed as

$$\begin{aligned}
&\sum_{i=1}^{N} a_{0i} = 0,\ \sum_{i=1}^{N} b_{0i} = 0,\ \sum_{i=1}^{N} c_{0i} = 0, \\
&\sum_{i=1}^{N} a_{0i}\Delta x_{0i} = 1,\ \sum_{i=1}^{N} b_{0i}\Delta x_{0i} = 0,\ \sum_{i=1}^{N} c_{0i}\Delta x_{0i} = 0, \\
&\sum_{i=1}^{N} a_{0i}\Delta y_{0i} = 0,\ \sum_{i=1}^{N} b_{0i}\Delta y_{0i} = 1, \\
&\sum_{i=1}^{N} c_{0i}\Delta y_{0i} = 0, \\
&\sum_{i=1}^{N} a_{0i}\Delta z_{0i} = 0,\ \sum_{i=1}^{N} b_{0i}\Delta z_{0i} = 0, \\
&\sum_{i=1}^{N} c_{0i}\Delta z_{0i} = 1.
\end{aligned} \tag{15}$$

To satisfy the geometric conservation law and first-order consistency, the Lagrange multiplier takes the form

$$\Lambda = J + \sum_{p=1}^{3} \mu_p M_p + \sum_{p=1}^{3}\sum_{q=1}^{3} \nu_{p,q} N_{p,q}, \tag{16}$$

where $J$ is the object function denoted in Eq. (4), whereas $M$ and $N$ take the form

$$\begin{aligned}
&M_1 = \sum_{i=1}^{N} a_{0i} = 0,\ M_2 = \sum_{i=1}^{N} b_{0i} = 0, \\
&M_3 = \sum_{i=1}^{N} c_{0i} = 0, \\
&N_{1,1} = \sum_{i=1}^{N} a_{0i}\Delta x_{0i} = 1, \\
&N_{1,2} = \sum_{i=1}^{N} b_{0i}\Delta x_{0i} = 0, \\
&N_{1,3} \sum_{i=1}^{N} c_{0i}\Delta x_{0i} = 0, \\
&N_{2,1} = \sum_{i=1}^{N} a_{0i}\Delta y_{0i} = 0, \\
&N_{2,2} = \sum_{i=1}^{N} b_{0i}\Delta y_{0i} = 1, \\
&N_{2,3} \sum_{i=1}^{N} c_{0i}\Delta y_{0i} = 0, \\
&N_{3,1} = \sum_{i=1}^{N} a_{0i}\Delta z_{0i} = 0, \\
&N_{3,2} = \sum_{i=1}^{N} b_{0i}\Delta z_{0i} = 0, \\
&N_{3,3} \sum_{i=1}^{N} c_{0i}\Delta z_{0i} = 1.
\end{aligned} \tag{17}$$

The constrained least squares problem with a Lagrange multiplier can be solved in a similar fashion to the simple least squares problem by finding $\nabla\Lambda = 0$ with respect to $\nabla\hat{\phi}_0$, $\mu_p$ and $\nu_{p,q}$. $\nabla\Lambda = 0$ can be written in matrix form, as follows:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \tag{18}$$

where

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{D} & \boldsymbol{E} \\ \boldsymbol{E}^T & \boldsymbol{0} \end{bmatrix},$$

$$\boldsymbol{D} = \begin{bmatrix} \boldsymbol{S} & 0 & \ldots & 0 \\ 0 & \boldsymbol{S} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \boldsymbol{S} \end{bmatrix},$$

$$\boldsymbol{E} = \begin{bmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \\ \vdots \\ \boldsymbol{e}_N \end{bmatrix}, \tag{19}$$

$$\boldsymbol{e}_i = \begin{bmatrix} \boldsymbol{I} & \Delta x_{0i}\boldsymbol{I} & \Delta y_{0i}\boldsymbol{I} & \Delta z_{0i}\boldsymbol{I} \end{bmatrix}, \tag{20}$$

$$\boldsymbol{x} = \begin{bmatrix} a_{01} \\ b_{01} \\ c_{01} \\ \vdots \\ a_{0N} \\ b_{0N} \\ c_{0N} \\ \mu_1 \\ \mu_2 \\ \mu_3 \\ \nu_{1,1} \\ \nu_{1,2} \\ \vdots \\ \nu_{3,3} \end{bmatrix}, \boldsymbol{b} = \begin{bmatrix} \omega_{01} \Delta \boldsymbol{r}_{01}^{\mathrm{T}} \\ \vdots \\ \omega_{0N} \Delta \boldsymbol{r}_{0N}^{\mathrm{T}} \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \tag{22}$$

where $\boldsymbol{I}$ is a $3 \times 3$ identity matrix and $\boldsymbol{S}$ is the matrix in Eq. (8). As a consequence, $\boldsymbol{D}$ is a $3N \times 3N$ matrix, because $S$ is a $3 \times 3$ matrix. The meshless coefficients may be obtained by multiplying the inverse matrix of $\boldsymbol{A}$ by $\boldsymbol{b}$.

The meshless coefficients derived from Eq. (19) satisfy geometric conservation laws because the geometric conservative conditions in Eq. (15) are constrained when solving the least squares problem. It is known that geometric conservative meshless coefficients are robust, even in randomly distributed point clouds [11].

### 2.3 Governing Equation

In this study, three-dimensional Euler equations were considered to verify the effect of an implicit multicloud convergence accelerator. We proceeded as follows:

$$\frac{\partial \boldsymbol{q}}{\partial t} + \frac{\partial \boldsymbol{f}}{\partial x} + \frac{\partial \boldsymbol{g}}{\partial y} + \frac{\partial \boldsymbol{h}}{\partial z} = 0, \tag{23}$$

where $\boldsymbol{q}$ represents the set of conservative variables $\left[\rho \ \rho u \ \rho v \ \rho w \ \rho E\right]^{\mathrm{T}}$, and the convective fluxes $\boldsymbol{f}$, $\boldsymbol{g}$, $\boldsymbol{h}$ are expressed as

$$\boldsymbol{f} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ \rho uH \end{bmatrix}, \boldsymbol{g} = \begin{bmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vw \\ \rho vH \end{bmatrix}, \boldsymbol{h} = \begin{bmatrix} \rho w \\ \rho wu \\ \rho wv \\ \rho w^2 + p \\ \rho wH \end{bmatrix}. \tag{24}$$

In Eq. 24, $p$ is the pressure and $\rho, u, v, w, E, H$ are the mass density, Cartesian velocity components, total energy, and total enthalpy (that is defined as $E + \rho/p$), respectively. From the equation of state of an ideal gas,

$$E = \frac{p}{(\gamma - 1)\rho} + \frac{1}{2}\left(u^2 + v^2 + w^2\right). \tag{25}$$

The Euler equations may be discretized as follows:

$$\frac{\partial \boldsymbol{q}}{\partial t} + \left[\sum_i^N a_{0i}\left(\boldsymbol{f}_i - \boldsymbol{f}_0\right) + \sum_i^N b_{0i}\left(\boldsymbol{g}_i - \boldsymbol{g}_0\right) + \sum_i^N c_{0i}\left(\boldsymbol{h}_i - \boldsymbol{h}_0\right)\right] = 0. \tag{26}$$

In Eq. (26), the coefficients $a_{0i}, b_{0i}$, and $c_{0i}$ may be obtained in various ways. In this study, meshless discretization was applied to the residual computation for all levels. Meshless discretization using the GC-LSM was exploited to obtain a fine-level solution and accelerated convergence. At the fine level, the nodes of the unstructured grids formed by the advancing-front algorithm [14] were used, and its edge-connected nodes were used for its local point cloud discretization, as shown in Fig. 2. AUSMUP$^+$[15] was used for the flux scheme because of its stable convergence behavior for all speeds, and TVD-Minmod was used for unstructured grid types [16,17].

Meshless discretization is known to be less accurate because of its non-conservative behavior. Thus, the GC-LSM was used because the method performs well in three-dimensional space compared to finite volume methods, owing to its conservative properties, while other meshless discretizations exhibit substantial errors in the mass flux [11].

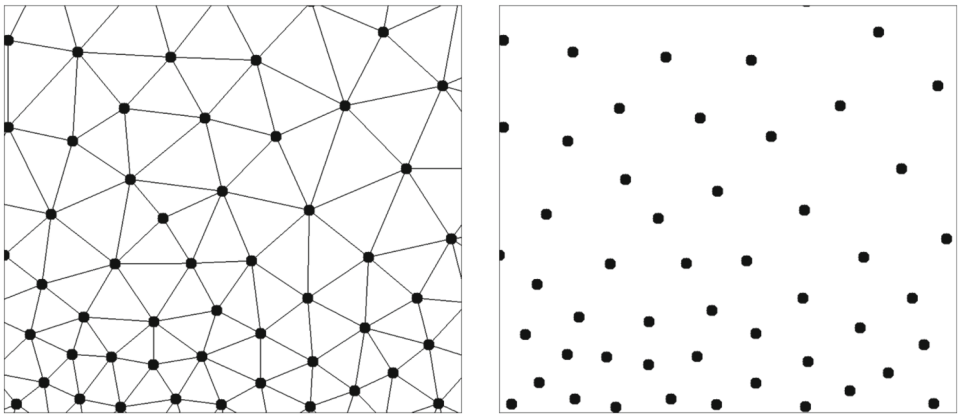
## 3 Coarsening Strategy

In this study, the methodology of the multicloud coarsening procedure described by Katz and Jameson [7] (which originated from Chan and Smith [3]) was used. The coarsening proceeded as shown in Algorithm 1.

**Algorithm 1** Multicloud coarsening

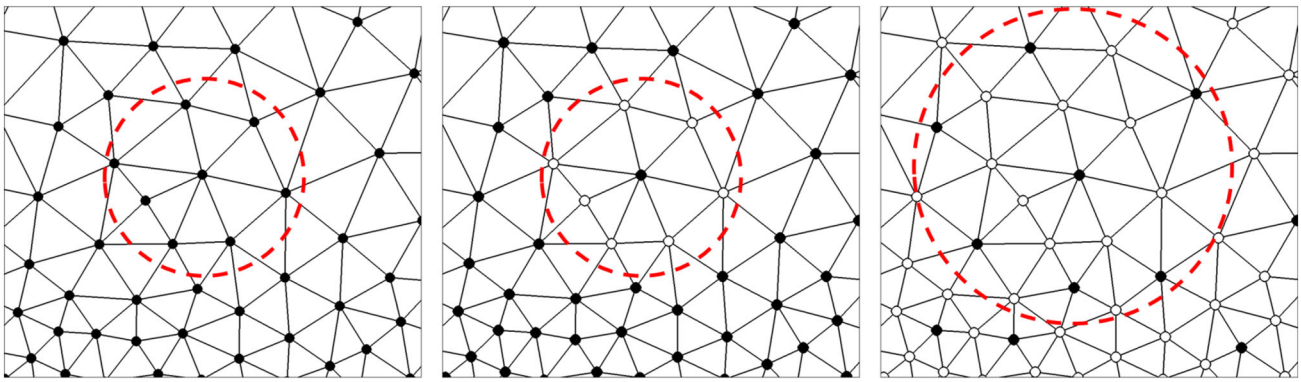
1: For each node $v_i$, Set $\Gamma(v_i)$ = .True., where $\Gamma$ denotes the logical function that defines validity on the next coarse level. The nodes with $\Gamma(v_i)$ = .True. are shown as black dots in Fig.3a.
2: Define $V(v_i) = \{v_j \| v_j$ is a vertex that forms the edge with the vertex $v_i\}$ which is denoted as a red dashed circle in Fig.3a.
3: **while** $i \leq N_c$ **do**, where $N_c$ is the number of computational points
4: **if** $\Gamma(v_i)$ is .True. **then**
5: $\Gamma(V(v_i))$ =.False.
6: **end if**
7: **end while**
8: The loop results in point validity determination as shown in Fig.3c.
9: Form coarse-level local clouds $V^{k+1}(v_i)$, where $k+1$ denotes the coarse level as shown in Fig.3c.
10: Perform the coarsening through coarse-level local point clouds $V^{k+1}(v_i)$ until the coarsest level is reached

This strategy is applicable to a three-dimensional unstructured grid for node-based computational methods.



**(a)** The set of nodes of grids **(b)** Point cloud obtained by nodes

**Fig. 2** Illustration of global point cloud from unstructured grid nodes



**(a)** Definition of local point cloud **(b)** Validity determination **(c)** Traverse and coarse local point cloud

**Fig. 3** Multicloud coarsening procedure

## 4 Restriction and Prolongation

For the implicit multicloud method, the Full Approximation Storage (FAS) algorithm of Brandt [18] is also used, as described in Algorithm 2.

**Algorithm 2** FAS algorithm

1: Fine level state variables at a node are directly transferred to the coarse level point. The fine level state vector $\boldsymbol{q}_0^k$ at node 0 is transferred to the coincident coarse level point $\boldsymbol{q}_0^{k+1}$.
2: Compute forcing function $\boldsymbol{P}$ using fine-level residuals of nearest points as in Eq. (27)
3: Perform iteration on the coarse level with restricted residuals $\boldsymbol{R}^{k+1} + \boldsymbol{P}$
4: From the solutions computed on the coarse level, correct the fine-level solutions by the nearest coarse-level valid point denoted as black nodes in Fig.4b.

The residual forcing function $\boldsymbol{P}$ is computed as follows:

$$\boldsymbol{P} = T_R \boldsymbol{R}_k(\boldsymbol{q}_k) - \boldsymbol{R}_{k+1}\left(\boldsymbol{q}_{k+1}^{(0)}\right), \tag{27}$$

where

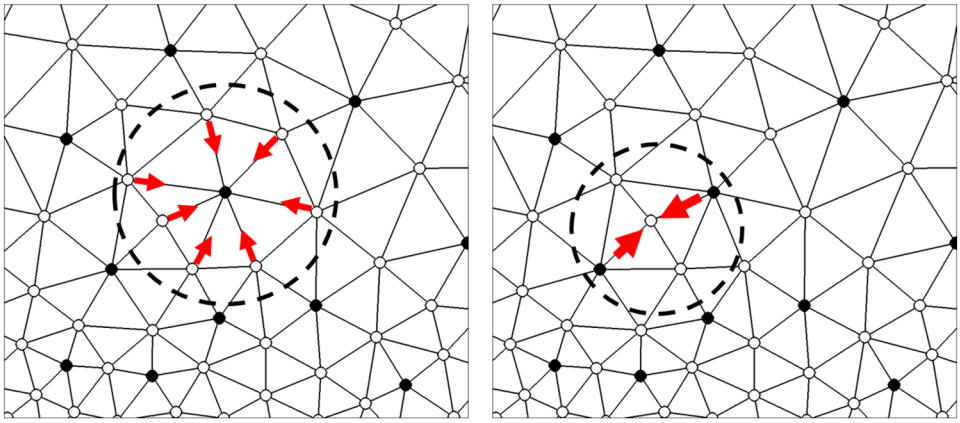
$$T_R^T = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_N \end{bmatrix}, \boldsymbol{R} = \begin{bmatrix} R_{0,1} & R_{0,1} & \dots & R_{0,m} \\ R_{1,1} & R_{1,2} & \dots & R_{1,m} \\ \vdots & \vdots & \ddots & \vdots \\ R_{N,1} & R_{N,2} & \dots & R_{N,m} \end{bmatrix}. \tag{28}$$

In Eq. (28), $N$ denotes the number of nearest points at level $k$ in Fig. 4a, whereas $m$ denotes the number of equations. In this study, $m$ equalled five because three-dimensional Euler equations were considered. The parameter $\beta_i$ is defined as follows:

$$\beta_0 = \left(\mathrm{d}s_k / \mathrm{d}s_{k+1}\right)^3, \tag{29}$$

$$\beta_i = \left(\frac{1-\beta_0}{\sum_{j \in V^k(0)} c_j}\right) c_i, \tag{30}$$

where $V^k(0)$ denotes the set of restriction stencils in Fig. 4a, and $\mathrm{d}s_k$ and $ds_{k+1}$ denote the average distances between the nearest points of the fine and coarse level local clouds, respectively. Because $\beta_0$ is a function similar to the ratio of the volume of the fine and coarse level cells, it is a cube of the



**(a)** Restriction stencils **(b)** Prolongation stencils

**Fig. 4** Stencils for multicloud operators

ratio in three-dimensional space (whereas it is a square in the original multicloud method). The parameter $c_i$ is the inverse distance between points, which is expressed as

$$c_i = \begin{cases} |\, r_i - r_0 \,|^{-1} & \text{if } i \neq 0 \\ 1 & \text{if i = 0.} \end{cases} \tag{31}$$

Fine level solutions are corrected by the change caused by an iteration at the coarse level, as shown in Fig. 4b. Prolongation of the solution is performed by the inverse weighted sum, as follows:

$$\boldsymbol{q}_0^+ = \boldsymbol{q}_0 + \frac{\sum_i c_i \left(\boldsymbol{q}_i^+ - \boldsymbol{q}_i\right)}{\sum_i c_i}. \tag{32}$$

In Eq. (32), $i$ is the set of nearest coarse-level valid points, depicted as black-colored nodes in the dashed circle in Fig. 4b. Both restriction and prolongation were formulated using a pure meshless process for a multicloud.

## 5 Meshless LU-SGS

LU-SGS was used for the implementation of implicit time integration because its suitability for this purpose is well established in industrial CFD, for both structured and unstructured methods.

Equation (26) may be implicitly recast as

$$\frac{\Delta \boldsymbol{q}_0^n}{\Delta \tau} + \sum_i^N \boldsymbol{H}_{0i}^{n+1} = 0, \tag{33}$$

where

$$\boldsymbol{H}_{0i} = a_{0i}\left(\boldsymbol{f}_i - \boldsymbol{f}_0\right) + b_{0i}\left(\boldsymbol{g}_i - \boldsymbol{g}_0\right) + c_{0i}\left(\boldsymbol{h}_i - \boldsymbol{h}_0\right). \tag{34}$$

Then, Eq. (34) may be linearized as

$$\boldsymbol{H}_{0i}^{n+1} = \boldsymbol{H}_{0i}^n + \boldsymbol{A}_{0i}^+ \Delta \boldsymbol{q}_0 + \boldsymbol{A}_{0i}^- \Delta \boldsymbol{q}_i, \tag{35}$$

where

$$\boldsymbol{A}_{0i}^{\pm} = \frac{1}{2}\left(\boldsymbol{A}_{0i} \pm \lambda_{0i} I\right), \tag{36}$$

$$\boldsymbol{A} = \partial \boldsymbol{H} / \partial \boldsymbol{q}, \tag{37}$$

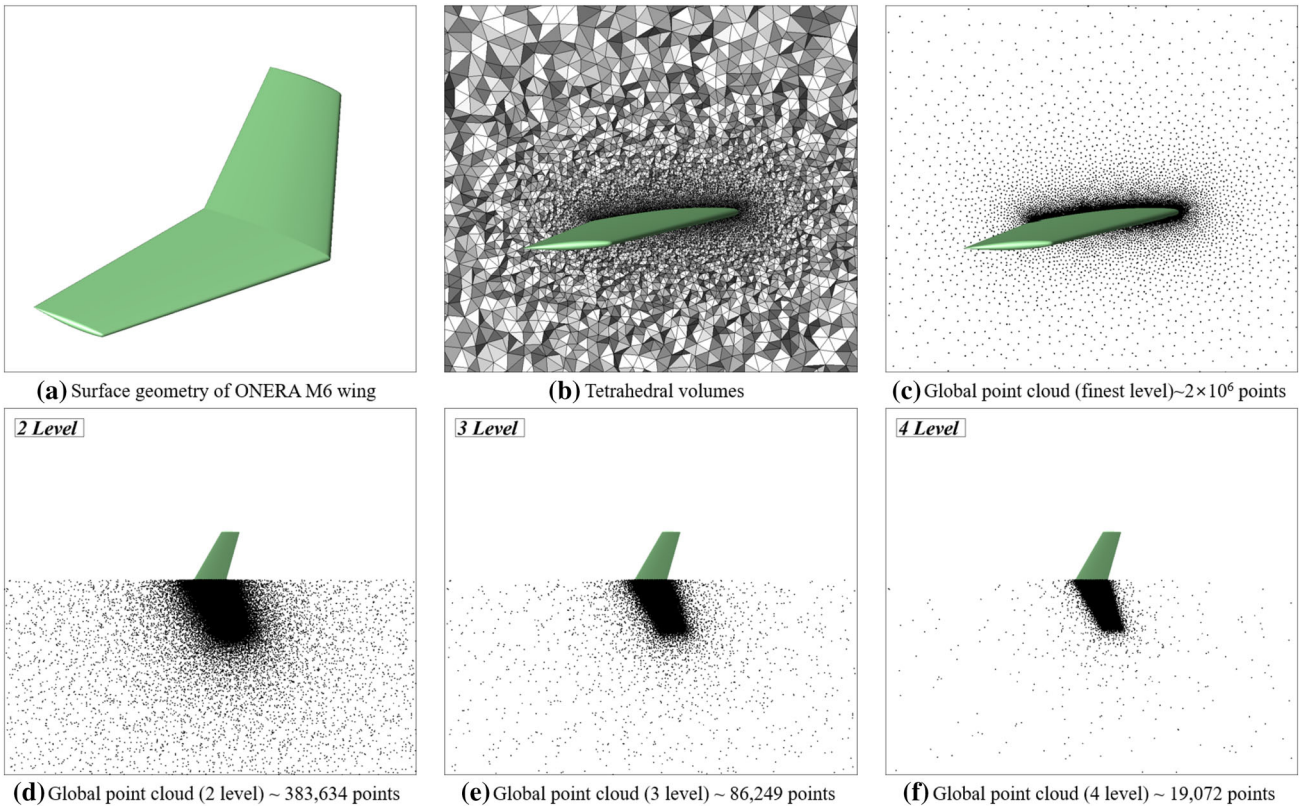
where $\lambda_{0i}$ is the eigenvalue of the Jacobian matrix $\boldsymbol{A}_{0i}$. By Eq. (35), Eq. (33) may be written as

$$\left(\frac{1}{\Delta \tau_0} + \frac{1}{2}\sum_i^N \lambda_{0i}\right) \boldsymbol{I}\, \Delta \boldsymbol{q}_0 + \sum_i^N \boldsymbol{A}_{0i}^- \Delta \boldsymbol{q}_i - \sum_i^N \boldsymbol{A}_{0i} \Delta \boldsymbol{q}_0 = -\sum_i^N \boldsymbol{H}_{0i}^n. \tag{38}$$

For finite volume methods, $\sum_i^N \boldsymbol{A}_{0i}$ in Eq. (38) must equal 0 because of the geometric conservation law. Thus, $\sum_i^N \boldsymbol{A}_{0i} = 0$ for GC-LSM because GC-LSM satisfies the geometric conservation law, as shown in Eq. (15), whereas non-conservative meshless schemes cannot be combined. Consequently, Eq. (38) may be recast as

$$\left(\frac{1}{\Delta \tau_0} + \frac{1}{2}\sum_i^N \lambda_{0i}\right) \boldsymbol{I}\, \Delta \boldsymbol{q}_0 + \sum_{i \in L(0)}^N \boldsymbol{A}_{0i}^- \Delta \boldsymbol{q}_i + \sum_{i \in U(0)}^N \boldsymbol{A}_{0i}^- \Delta \boldsymbol{q}_i = \boldsymbol{R}_0^n, \tag{39}$$

where $L(0)$ and $U(0)$ are the set of nearest points whose indices are less than and greater than those for point 0, respec-




**(a)** Surface geometry of ONERA M6 wing **(b)** Tetrahedral volumes **(c)** Global point cloud (finest level)~$2\times10^6$ points

**(d)** Global point cloud (2 level) ~ 383,634 points **(e)** Global point cloud (3 level) ~ 86,249 points **(f)** Global point cloud (4 level) ~ 19,072 points

**Fig. 5** Coarsening results and the original geometry of ONERA M6

**Table 1** Units of work and CFL for each case

| Scheme | RK4 SG | RK4 MC | LU SG | LU MC |
|---|---|---|---|---|
| Unit work | 1.00 | 1.59 | 0.39 | 0.78 |
| CFL | 1 | 1 | 20 | 20 |

tively. Consequently, Eq.(39) may have the same form with unstructured LU-SGS so that the equation may be decomposed in the same way as for the unstructured method:

$$\Delta \boldsymbol{q}_0^* = \boldsymbol{D}_0^- \left[ \boldsymbol{R}_0^n - \sum_{i \in L(0)}^{n} \boldsymbol{A}_{0i}^- \Delta \boldsymbol{q}_i^* \right], \tag{40}$$

$$\Delta \boldsymbol{q}_0 = \Delta \boldsymbol{q}_0^* - \boldsymbol{D}_0^- \sum_{i \in U(0)}^{N} \boldsymbol{A}_{0i}^- \Delta \boldsymbol{q}_i, \tag{41}$$

where

$$\boldsymbol{D}_0 = \left( \frac{1}{\Delta \tau_0} + \frac{1}{2} \sum_{i}^{N} \lambda_{0i} \right) \boldsymbol{I}. \tag{42}$$

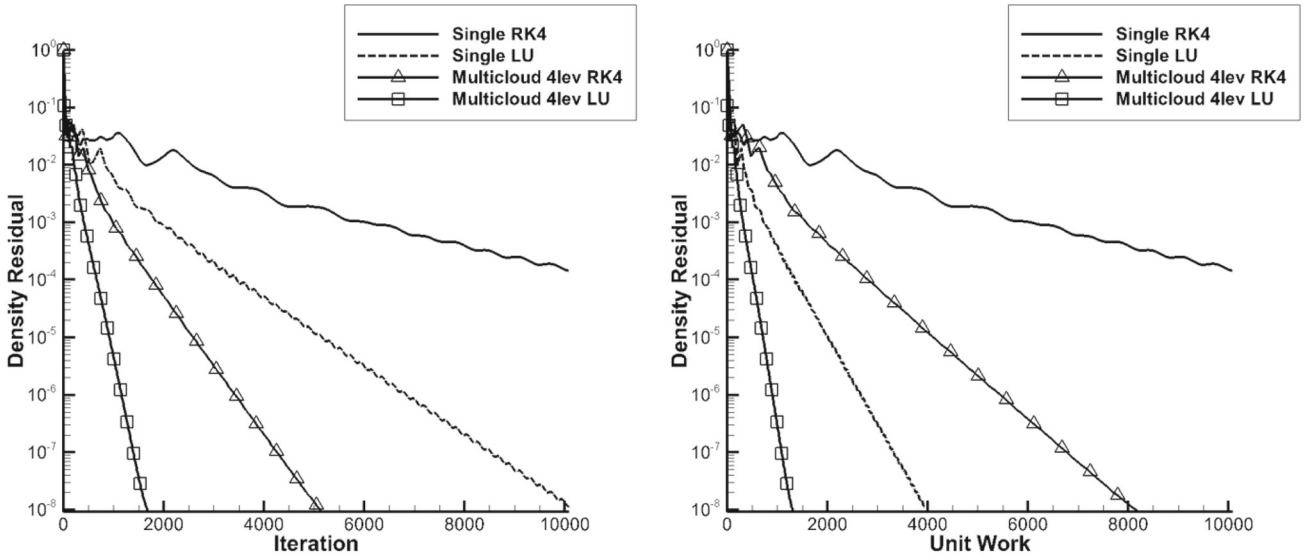
Equations (41) and (40) are forward and backward sweeps, respectively. Thus, LU-SGS may be implemented for meshless discretization, which accelerates convergence without algorithm complexity using only GC-LSM.

## 6 Results

### 6.1 ONERA M6

The first test case in this study consists of transonic flow around the ONERA M6 wing, which is the most widely used model for the three-dimensional validation case. The fine level tetrahedral grid is shown in Fig. 5b. In this study, meshless discretization was used for all levels, such that the nodes of the grids were considered as meshless points, as shown in Fig. 5c. The coarsening results are shown in Fig. 5d, e. The Mach number of the flow was 0.8395 with an angle of attack of 3.06°. Four cases were tested: the explicit 4th-order Runge–Kutta (RK4) method and implicit LU-SGS, for both single grid and multicloud methods.

With respect to the Courant–Friedrich–Lewy (CFL) condition, the CFD numbers for the explicit and implicit schemes were set to 1 and 20, respectively. Here, we define the unit work as CPU time per iteration on the finest grid compared




(a) Density residual history per iteration (b) Density residual history per unit work

**Fig. 6** Convergence history of ONERA M6 test case

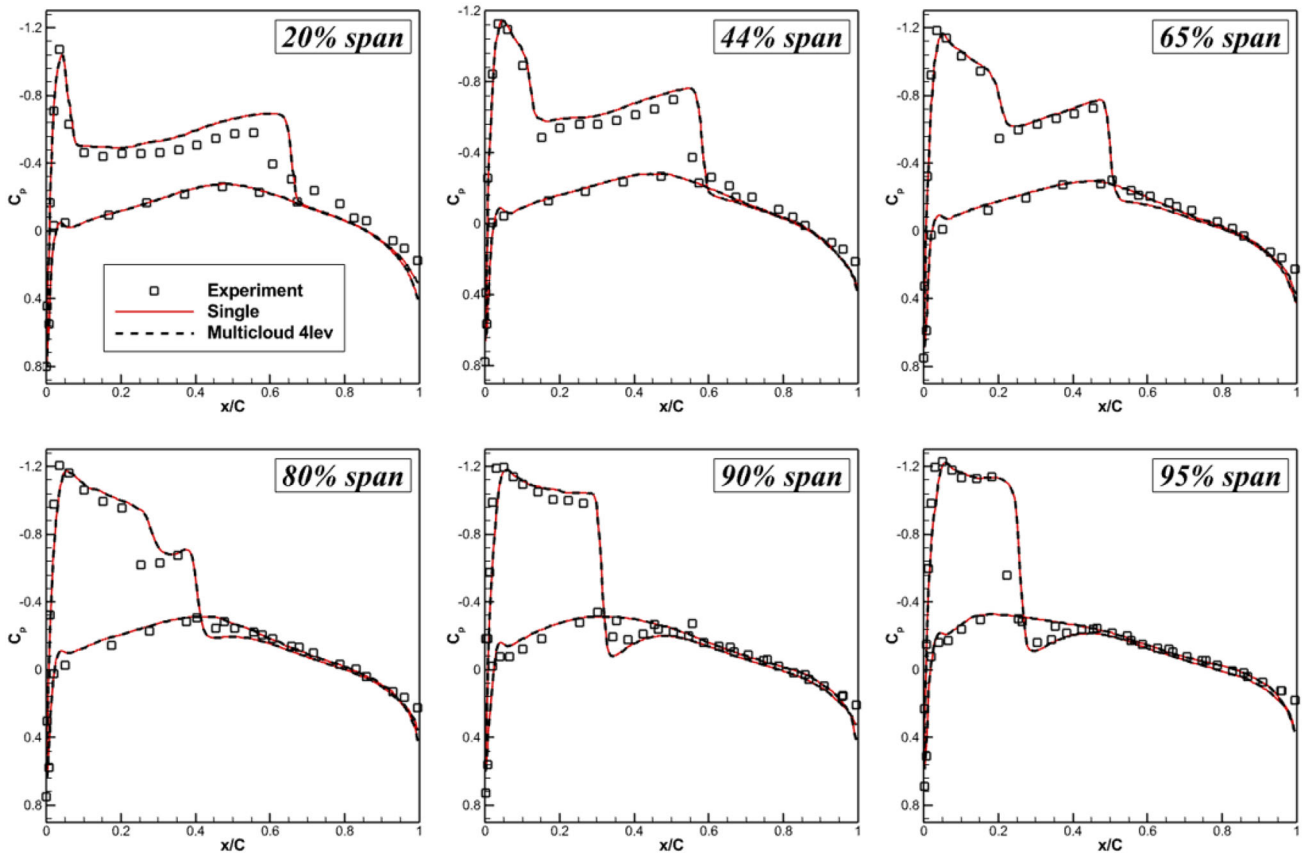
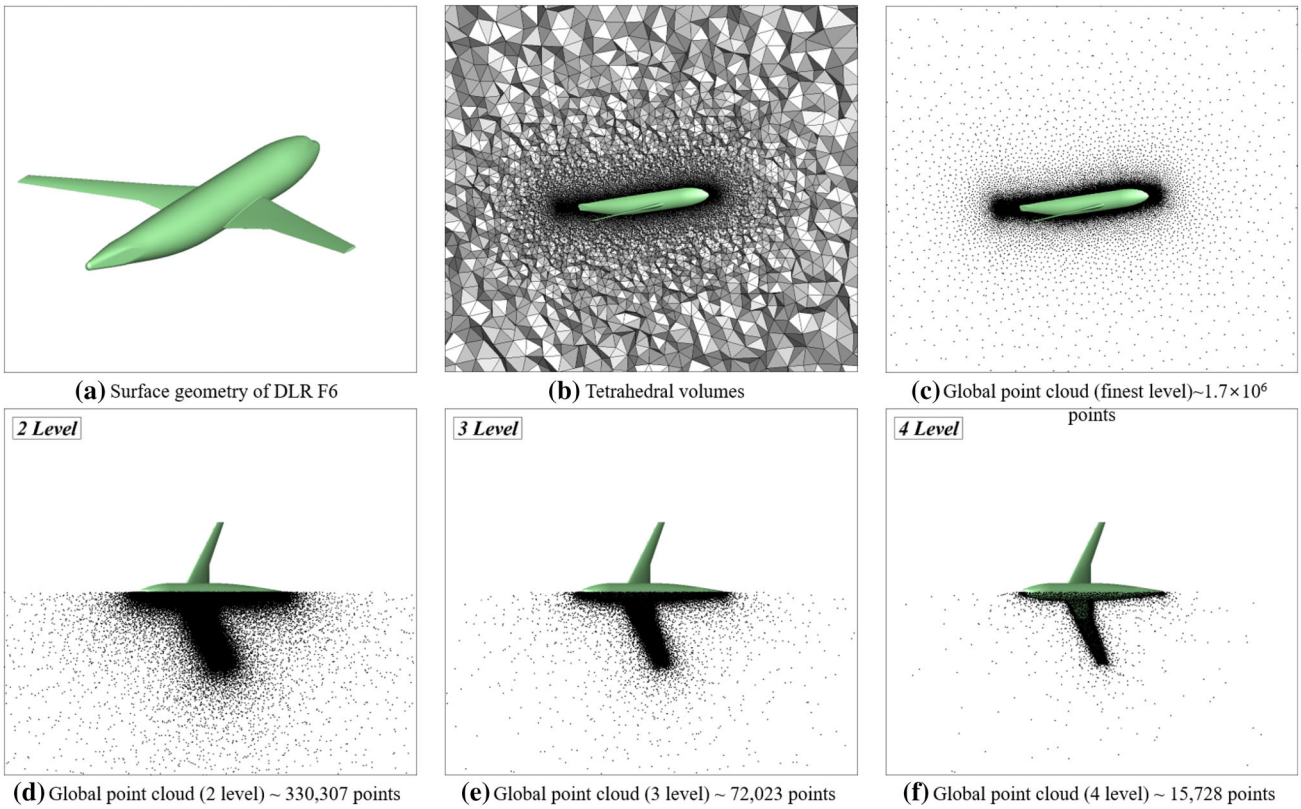


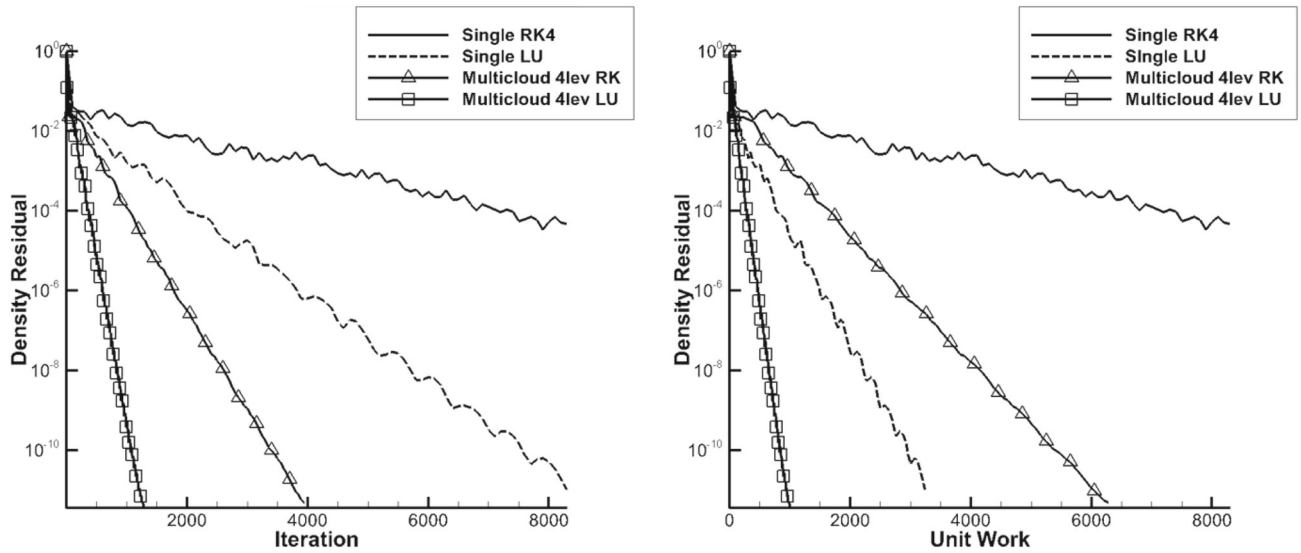

**Fig. 7** Surface pressure of ONERA M6 test case




**(a)** Surface geometry of DLR F6 **(b)** Tetrahedral volumes **(c)** Global point cloud (finest level)~$1.7\times10^6$ points

**(d)** Global point cloud (2 level) ~ 330,307 points **(e)** Global point cloud (3 level) ~ 72,023 points **(f)** Global point cloud (4 level) ~ 15,728 points

**Fig. 8** Coarsening results and the original geometry of DLR-F6




**(a)** Density residual history per iteration **(b)** Density residual history per unit work

**Fig. 9** Convergence history of DLR-F6 test case

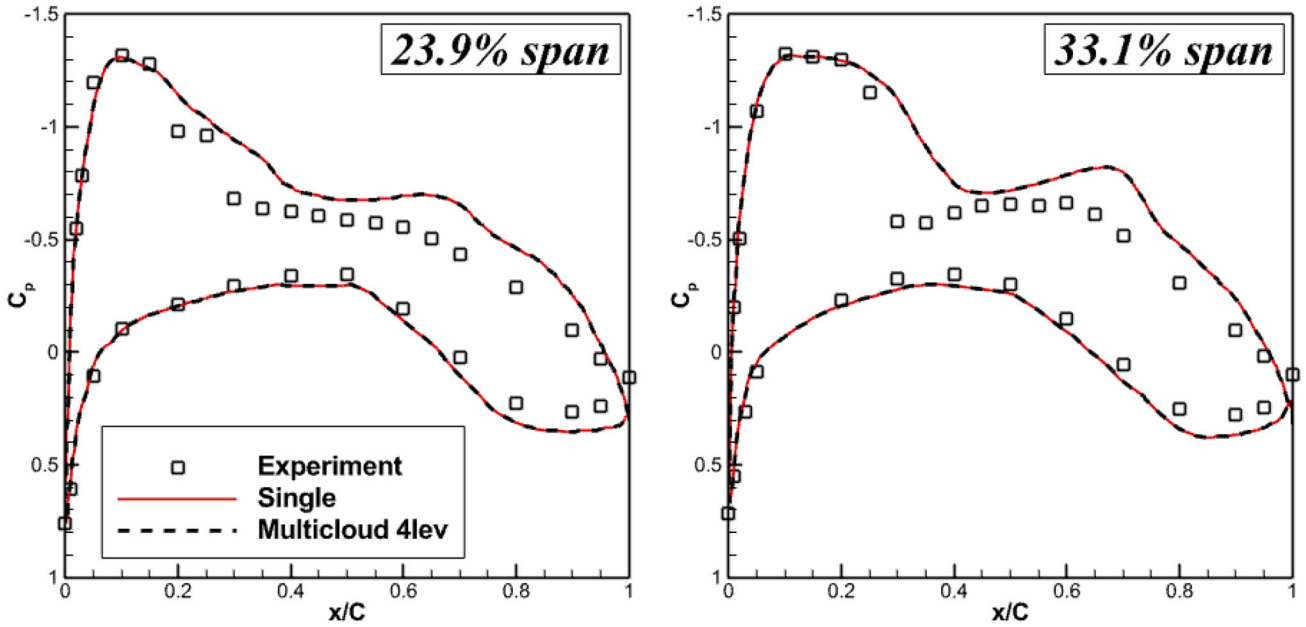



**Fig. 10** Surface pressure of DLR-F6 test case

to the single RK4 method, such that the work associated with RK4 is unity, as shown in Table 1. The convergence history is presented in Fig. 6. Figure 6a displays the density residual per iteration on the finest grid, and Fig. 6b displays the density residual per unit work. In Fig. 6a, the four-level procedure with LU-SGS exhibits dramatic convergence speedup with respect to the number of iterations. Four levels with RK4 also exhibited substantial speedup, exceeding that of single LU-SGS. However, the work unit associated with RK4 exceed those associated with LU, such that the convergence history with respect to the number of work units shows a different result compared to the history with respect to the number of iterations, as shown in Fig. 6b. In terms of unit work, four levels with LU-SGS exhibited the highest convergence rates, as formulated in this study. For the ONERA M6 wing, inviscid analyses exhibited slight differences in surface pressure because turbulence is not a crucial factor in that context [19]. The surface pressure results are compared with the experimental results [20] in Fig. 7. The surface pressure results agree well with the experimental results. However, slight differences in the location of the shock on both the 0.2 and the 0.8 span are shown. These differences are typical of inviscid methods [21]. Finally, the four-level procedure and single LU display satisfactory agreement with each other as well as with the experimental results.

### 6.2 DLR-F6

To highlight the performance of the implicit multicloud method, the flow around the baseline wing-body DLR-F6 [22], whose geometry is complicated, was analyzed. The Mach number of the flow was 0.75 with an angle of attack of 0.49°. In Fig. 8, the finest and coarse level clouds are shown. It appears that the coarsening procedure works well, even for complex geometries.

In Fig. 9, residual history results are plotted in the same manner as for the ONERA M6 test case. For the complicated geometry, the four-level LU-SGS procedure exhibits the highest convergence speed compared to other methods. In Fig. 10, the results of surface pressure are compared for the four-level, single, and experimental [23] cases. As seen in Fig. 10, the four-level and single results show strong agreement. However, a disagreement with the experimental results is shown downstream, where the shock is located, while strong agreement is shown around the stagnation line. As mentioned previously, these disagreements are typical for the inviscid method [21]. Furthermore, the DLR-F6 model is more sensitive to turbulence than the ONERA M6 model because the DLR-F6 geometry has a large separation bubble on the wing [22]. Although it is necessary to solve the Navier–Stokes equations with turbulence modelling to obtain accurate surface pressure results, considering turbulent flow is beyond the scope of this study, and a multicloud for viscous flow will be discussed in a future study. Such that it seems that disagreement is acceptable for inviscid method. The L2 norm of the surface pressure difference between the single and four-level procedures is also less than 1e-11, such that the four-level and single methods display the same results.

## 7 Conclusion

An implicit multicloud convergence accelerator using the GC-LSM has been considered in this study. Owing to the geometric conservative feature of GC-LSM, the original LU-SGS algorithm for unstructured methodology was easily combined with meshless discretization, maximizing the convergence speedup of the multicloud method. Furthermore, the convergence acceleration effect was demonstrated for complex geometries such as the DLR-F6 model, displaying no differences with respect to the single grid method. As a result, we conclude that the implicit multicloud method presents a dramatic speedup effect compared to the original explicit multicloud method using the 4th-order Runge–Kutta scheme, without incurring programming and mathematical complexity.

**Acknowledgements** This work was supported by the National Research Foundation of Korea (NRF) Grant funded by the Ministry of Science and ICT (NRF-2017R1A5A1015311).

### Declarations

**Conflict of interest** All authors have no conflicts of interest.

## References

1. Fedorenko RP (1964) The speed of convergence of one iterative process. USSR Comput Math Math Phys 4(3):227–235
2. Jameson A (2006) Multigrid algorithms for compressible flow calculations. In: Multigrid Methods II: Proceedings of the 2nd European Conference on multigrid methods held at Cologne, October 1–4, 1985, pp 166–201. Springer
3. Chan TF, Smith BF (1994) Domain decomposition and multigrid algorithms for elliptic problems on unstructured meshes. Contemp Math 180:175–175
4. Lallemand M-H, Steve H, Dervieux A (1992) Unstructured multigridding by volume agglomeration: current status. Comput Fluids 21(3):397–433
5. Nishikawa H, Diskin B, Thomas J (2010) Development and application of agglomerated multigrid methods for complex geometries. In: 40th Fluid Dynamics Conference and Exhibit, p 4731
6. Nishikawa H, Diskin B, Thomas J, Hammond D (2013) Recent advances in agglomerated multigrid. In: 51st AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, p 863
7. Katz A, Jameson A (2009) Multicloud: multigrid convergence with a meshless operator. J Comput Phys 228(14):5237–5250
8. Zamolo R, Nobile E, Šarler B (2019) Novel multilevel techniques for convergence acceleration in the solution of systems of equations arising from rbf-fd meshless discretizations. J Comput Phys 392:311–334
9. Yoon S, Jameson A (1988) Lower-upper symmetric-Gauss-Seidel method for the Euler and Navier-Stokes equations. AIAA J 26(9):1025–1026
10. Sharov D, Nakahashi K, Sharov D, Nakahashi K (1997) Reordering of 3-d hybrid unstructured grids for vectorized lu-sgs navier-stokes computations. In: 13th Computational Fluid Dynamics Conference, p 2102
11. Huh JY, Rhee JS, Kim KH, Jung SY (2018) New least squares method with geometric conservation law (gc-lsm) for compressible flow computation in meshless method. Comput Fluids 172:122–146
12. Parsani M, Van den Abeele K, Lacor C, Turkel E (2010) Implicit lu-sgs algorithm for high-order methods on unstructured grid with p-multigrid strategy for solving the steady Navier-Stokes equations. J Comput Phys 229(3):828–850
13. Dargaville S, Buchan AG, Smedley-Stevenson RP, Smith PN, Pain CC (2021) A comparison of element agglomeration algorithms for unstructured geometric multigrid. J Comput Appl Math 390:113379
14. Löhner R, Parikh P (1988) Generation of three-dimensional unstructured grids by the advancing-front method. Int J Numer Methods Fluids 8(10):1135–1149
15. Liou M-S (2006) A sequel to ausm, part ii: Ausm+-up for all speeds. J Comput Phys 214(1):137–170
16. Sweby PK (1984) High resolution schemes using flux limiters for hyperbolic conservation laws. SIAM J Numer Anal 21(5):995–1011
17. Darwish M, Moukalled F (2003) Tvd schemes for unstructured grids. Int J Heat Mass Transf 46(4):599–611
18. Brandt A (1977) Multi-level adaptive solutions to boundary-value problems. Math Comput 31(138):333–390
19. Liu C, Hu C (2017) An immersed boundary solver for inviscid compressible flows. Int J Numer Methods Fluids 85(11):619–640
20. Schmitt V, Charpin F (1979) Pressure distributions on the ONERA-M6-WING at transonic Mach numbers, experimental data base for computer program assessment, AGARD-AR-138, pp B1-1–B1-44
21. Batina JT (1991) Accuracy of an unstructured-grid upwind-Euler algorithm for the onera m6 wing. J Aircr 28(6):397–402
22. Laflin KR, Klausmeyer SM, Zickuhr T, Vassberg JC, Wahls RA, Morrison JH, Brodersen OP, Rakowitz ME, Tinoco EN, Godard J-L (2005) Data summary from second aiaa computational fluid dynamics drag prediction workshop. J Aircr 42(5):1165–1178
23. Rumsey CL, Rivers SM, Morrison JH (2005) Study of cfd variation on transport configurations from the second drag-prediction workshop. Comput Fluids 34(7):785–816