공학박사학위논문

무격자 기법을 이용한 격자 비종속 수렴 가속 기법 개발

A Development of a Mesh-transparent Convergence Accelerator Based on a Meshless Method

2023년 8월

서울대학교 대학원
기계항공공학부
이 재 상

무격자 기법을 이용한 격자 비종속 수렴 가속 기법 개발

A Development of a Mesh-transparent Convergence Accelerator Based on a Meshless Method

지도교수 김 규 홍

이 논문을 공학박사 학위논문으로 제출함

2023년 5월

서울대학교 대학원

기계항공공학부

이 재 상

이재상의 공학박사 학위논문을 인준함

2023년 6월

위 원 장	이 관 중	(인)
부위원장	김 규 홍	(인)
위 원	이 복 직	(인)
위 원	박 수 형	(인)
위 원	허 기 훈	(인)

Abstract

This dissertation describes the development of a Mesh-transparent convergence accelerator based on a meshless Method. The mesh-transparent convergnece accelerator was named as multicloud method. The multicloud method relies on meshless discretization on coarse levels not using grid elements. The solutions computed by meshless discretization on the coarse level lead to the correction of fine-level solutions, resulting in convergence acceleration. The mutlcloud method significantly advatageous in forming coarse level domain and robust discretization on the coarse levels. Hence, the method can be implemented to any type of fine level discretization such as cell finite volume, nodal finite volume, and meshless methods.

A new improved cell coarsening strategy was developed in this study. The new strategy provide superior coarsening rates than the structured multigrid coarsening. However, the acceleration effect according to its coarsening rates can be obtained only if robust meshless discretization is guaranteed. Due to this, meshless discretization by least squares method with the geometric conservation law is used for robust discretization. The application of GC-LSM with the new coarsening strategy showed a dramatic convergence acceleration effect. In contrast, meshless discretization using the least squares method(LSM) without the conservation property fails to converge.

Implicit time integration method was also implemented to the multicloud method using lower-upper symmetric Gauss-Seidel (LU-SGS), which has not been implemented to the multicloud before.

Finally, pressure-based meshless damping functions were developed to accelerate the convergence of problems involving strong shocks. The results showed not only the efficacy of the multicloud method for hypersonic problems but also superior speed-up effect compared to than the structured multigrid method with the new coarsening method.

The results obtained from the developed algorithms demonstrate that the improved multicloud method provides a significant enhancement in efficiency for various types of flows, including inviscid, viscous, and hypersonic flows, regardless of geometries.

Keywords: meshless, geometric multigrid, convergence acceleration, implicit time integration, unstructured grid

Student Number: 2016-30190

Contents

Abstract **i**

Contents **iii**

List of Tables **vi**

List of Figures **vii**

1 Introduction **1**

1.1 Multigrid methods 1

1.2 Multicloud algorithm 5

1.3 Meshless discretization 6

1.4 Motivation 8

2 Multicloud **10**

2.1 Coarsening Strategy 10

2.1.1 Node coarsening strategy 10

2.1.2 Cell coarsening strategy 12

2.1.3 Directional coarsening 18

2.1.4 Composing coarse level local point clouds 21

2.2 Multicloud Operator 22

2.3 multicloud operators for hypersonic flows 26

2.3.1 Damped restriction 26
2.3.2 Damped prolongation 28

3 Numerical methods **30**
3.1 Governing equations 30
3.1.1 Navier-Stokes equations 30
3.1.2 Spalart-Allmaras Turbulence Model 32
3.2 Spatial discretization 33
3.2.1 Cell finite volume 33
3.2.2 Least Squares Method 33
3.2.3 Geometric Conservative Least Squares Method 36
3.3 Flux schemes 39
3.3.1 AUSMPW+ 39
3.3.2 AUSM$^+$-up 41
3.4 Limiter 42
3.4.1 TVD schemes 42
3.4.2 Venkatakrishnan limiter 44
3.4.3 Gradient calculation 45
3.5 Time integration 46
3.5.1 Local time stepping 46
3.5.2 Explicit Runge-Kutta 47
3.5.3 LU-SGS for unstructured grids 47
3.5.4 LU-SGS for non-conservative meshless discretization 51

4 Numerical Results **54**
4.1 Comparisons for explicit and implicit time integration 54
4.1.1 ONERA M6 55
4.1.2 DLR-F6 wing body configuration 56
4.2 Application to non-primal grid system 63

4.2.1 NACA0012 63
4.2.2 ONERA M6 68
4.3 Cell coarsening strategy comparisons 76
4.3.1 DLR-F6 wbnp configuration 76
4.3.2 ONERA M6 82
4.3.3 DLR-F6 wing body configuration 91
4.4 Laminar flows 96
4.4.1 NACA0012 Laminar flows 96
4.4.2 Laminar flows around the cylinder (Re 40) 100
4.5 Turbulent flows 104
4.5.1 Transonic flows around ONERA M6 (Re 11.75e6) 104
4.6 hypersonic flows 107
4.6.1 Mach 8 flow around the cylinder 107
4.6.2 Mach 8 flows around the sphere 110
4.6.3 HTV-2 113

5 Concluding remarks **116**
5.1 Conclusions 116
5.2 Future works 117

Abstract (In Korean) **128**

List of Tables

4.1 Units of work and CFL for each case 56
4.2 The number of point comparisons for each level for NACA0012 . . 68
4.3 The number of point comparisons for each level 78
4.4 Description for test cases 78
4.5 The number of point comparisons for each level 84

List of Figures

1.1 non-nested multigrid 2
1.2 subdivision of triangle cell 3
1.3 Retriangulation 4
1.4 Edge collapsing 4
1.5 The original grid(a) and its agglomerated grid(b) 5
1.6 The difference between multicloud and grid agglomeration 6
1.7 The difference between Cell finite volume method(a) and meshless method(b) 7

2.1 The definition of a local point cloud 12
2.2 The multicloud coarsening procedure 12
2.3 The definition of a initial local point cloud for node-centered methods 13
2.4 Comparison between ideal and CNA coarsening 15
2.5 Cell coarsening procedures for each method 16
2.6 Coarsening results comparison 17
2.7 Prism grids for high Reynolds number flow 19
2.8 Illustration of id labelling 19
2.9 Resulting coarse level point by directional coarsening 20
2.10 Illustration of procedure of forming coarse level local point cloud . 21
2.11 Illustration of procedure of forming coarse level local point cloud . 23
2.12 Illustration of stencil for multicloud operators 24

2.13 Restriction stencil for structured grids 27
2.14 Illustration of meshless damped restriction 28
2.15 Illustration of shock sensing 29

3.1 Illustration of cell finite volume method 34
3.2 Node notation for unstructured TVD schemes 43
3.3 Node notation for Venkatakrishnan limiter 45
3.4 Illustration of nodal value interpolation 46
3.5 Illustration of flow grid for f6wbnp 52
3.6 Comparison between M-LU and U-LU with LSM discretization . . 53

4.1 Coarsening results for ONERA M6 57
4.2 Convergence histories for ONERA M6 58
4.3 Surface pressure along y=0.2η 59
4.4 Surface pressure along y=0.44η 59
4.5 Surface pressure along y=0.65η 60
4.6 Surface pressure along y=0.8η 60
4.7 Surface pressure along y=0.9η 61
4.8 Surface pressure along y=0.95η 61
4.9 Surface pressure contour for ONERA M6 63
4.10 Coarsening results for DLR-F6 64
4.11 Convergence histories for DLR-F6 65
4.12 Surface pressure along $y = 0.239\eta$ 66
4.13 Surface pressure along $y = 0.331\eta$ 66
4.14 Surface pressure contour for dlr-f6 wing body 67
4.15 Illustration of computational domains for NACA0012 67
4.16 Convergence history for NACA0012, M=0.5 AOA=3° (QuaP) . . . 68
4.17 Surface pressure for NACA0012, M=0.5 AOA=3° (QuaP) 69
4.18 Convergence history for NACA0012, M=0.85 AOA=1° (QuaP) . . . 69

4.19 Surface pressure for NACA0012, M=0.85 AOA=1° (QuaP) 70
4.20 Each level point distribution based on octree-prism for ONERA M6 71
4.21 Convergence history for ONERA M6 (OctP) 72
4.22 Surface pressure along y=0.2η 72
4.23 Surface pressure along y=0.44η 73
4.24 Surface pressure along y=0.65η 73
4.25 Surface pressure along y=0.8η 74
4.26 Surface pressure along y=0.9η 74
4.27 Surface pressure along y=0.95η 75
4.28 Configuration of f6wbnp and its flow grid 77
4.29 Convergence histories for f6wbnp in term of iteration 79
4.30 Surface results of fully converged solutions on level 2 point domains 79
4.31 Convergence history of f6wbnp in terms of the walltime 80
4.32 Convergence history of lift coefficient for f6wbnp in terms of the iteration 80
4.33 Surface pressure comparison for f6wbnp 81
4.34 Illustration of the unstructured grid for the ONERA M6 wing. (a),(b) Tetrahedral grid (TG), (c) hybrid grid (HG). 84
4.35 Comparison of the coarsening results between TG-CNA (left) and TG-CFA (right) 84
4.36 Convergence history in term of iteration for transonic ONERA M6 flows (Tet) 85
4.37 Convergence history in term of walltime for transonic ONERA M6 flows (Tet) 85
4.38 Lift convergence history in term of iteration for transonic ONERA M6 flows (Tet) 86
4.39 Surface pressure along y=0.2η 86
4.40 Surface pressure along y=0.44η 87

4.41 Surface pressure along y=0.65η 87
4.42 Surface pressure along y=0.8η 88
4.43 Surface pressure along y=0.9η 88
4.44 Surface pressure along y=0.95η 89
4.45 Convergence history in term of iteration for transonic ONERA M6 flows (Hybrid) 89
4.46 Convergence history in term of walltime for transonic ONERA M6 flows (Hybrid) 90
4.47 Lift convergence history in term of iteration for transonic ONERA M6 flows (Hybrid) 90
4.48 Configuration of f6wb and its flow grid 92
4.49 Frontal view of surface point distribution on level 2 for f6wb 92
4.50 Frontal view of surface point distribution on level 2 for f6wb 93
4.51 Frontal view of surface point distribution on level 4 for f6wb 93
4.52 Convergence history of f6wb in terms of the iteration 94
4.53 Convergence history of f6wb in terms of the walltime 94
4.54 Convergence history of lift coefficient for f6wb in terms of iteration 95
4.55 Surface pressure comparison for f6wb 95
4.56 The flow grid for NACA0012 97
4.57 Density residual history for NACA0012 terms of iteration 97
4.58 Density residual history for NACA0012 in terms of walltime 98
4.59 Contour comparison for NACA0012 Laminar flows (Upper : Single, Nether : multicloud 4 levels 99
4.60 surface coefficient comparison for NACA0012 Laminar flows 99
4.61 Flow grid for Re 40 cylinder flows 100
4.62 Density residual history of Re 40 cylinder flow in terms of iteration . 101
4.63 Density residual history of Re 40 cylinder flow in terms of walltime 102
4.64 Streamline pattern in the downstream (upper : single, nether : 4 levels) 102

4.65 surface coefficient comparison for Re 40 cylinder flow 103
4.66 Illustration of coarsening results for the highly stretched grid for ONERA M6 105
4.67 Density residual history of ONERA M6 (Re 11.75e6) 105
4.68 $\tilde{\nu}$ residual history of ONERA M6 (Re 11.75e6) 106
4.69 Unstructured triangle grids for cylinder 107
4.70 Coarse level point cloud generated by CNA coarsening 108
4.71 Density residual history for Mach 8 cylinder in terms of iteration . . 109
4.72 Density residual history for Mach 8 cylinder in terms of walltime . . 109
4.73 Contour comparisons between the single and four-level grid 110
4.74 Symmetric flow grid for the sphere 111
4.75 Density residual history for Mach 8 sphere in terms of iteration . . . 111
4.76 Density residual history for Mach 8 sphere in terms of walltime . . . 112
4.77 Surface result comparisons for the Mach 8 sphere (left : single, right : four-level multicloud) 112
4.78 Symmetric flow grid for HTV-2 114
4.79 Density residual history for HTV-2 in terms of iteration 114
4.80 Density residual history for HTV-2 in terms of walltime 115
4.81 Result comparisons for HTV-2 115

Nomenclature

0	: Index of computation center point (subscript)
β_i	: Inverse distance weight between point 0 and i
$\boldsymbol{A}_{0i}$	: Flux Jacobian matrix between 0 and i
$\boldsymbol{H}_{0i}$	: Numerical Flux between 0 and i
$\boldsymbol{S}_{0i}$	: Cell interface area vector between cell 0 and i
a_{0i}	: Meshless discretization coefficient between 0 and i in x-direction
b_{0i}	: Meshless discretization coefficient between 0 and i in y-direction
c_{0i}	: Meshless discretization coefficient between 0 and i in z-direction
d	: Dimension
i	: Index of nearest point of 0 (subscript)
k	: The level of coarseness of computational domain (superscript)
N	: The number of nearest neighbors of 0
Re	: Reynolds number
V_0	: Volume of the cell 0

Chapter 1

Introduction

1.1 Multigrid methods

A multigrid method is a traditional convergence acceleration method for an iterative method. The multigrid method was based on the idea of solving a given problem on multiple grids with different resolutions. By moving the error from one grid to another resolution, it can be effectively dampened.

The first multigrid technique was proposed by Fedorenko[1]. And the multigrid techniques were initially applied to elliptic equations[1, 2]. Since then, the multigrid method has evolved and improved such that the methods have further applied to hyperbolic equations such as Euler[3, 4] and Navier-Stokes equations[5]. Furthermore, the structured multigrid method was applied to reacting flows[6, 7]

Under structured grids, forming grids of different resolutions is straightforward. However, it is not straightforward for unstructured grids. Then, Algebraic multigrid (AMG) method have been used[8, 9] since application geometric multigrid for unstructured grids is too difficult[10]. AMG does not required coarse meshes. The

coarsening procedure for AMG is not making coarse grid but the selection of coarse indices.

However, various strategies have been proposed to apply geometric multigrid (GMG) to unstructured grids since GMG provides extremely fast solutions for fluid problems. The most primitive approach is forming grids of difference resolutions manually which is called non-nested multigrid as shown in Figure 1.1. However, in CFD, manual grid generation of different resolutions is not feasible due to the significant labor and time involved in the process.

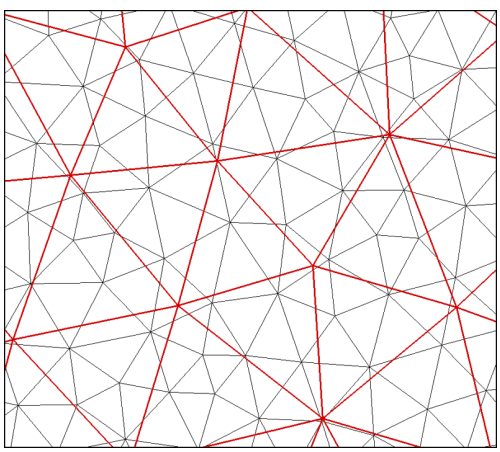


Figure 1.1: non-nested multigrid

To meet the demand for automatic coarse grid generation, an adaptation algorithm proposed by Perez[11] is used to generate unstructured grids of different resolutions, as shown in Figure 1.2. The strategy is applied to triangle and tetrahedral grids by Mavriplis[12, 13] for analysis of Euler equations. In numerical analysis, the accuracy of results is heavily reliant on fine-level resolution. Therefore, the quality of the grid should be determined based on the specific demands of the user. If fine level grids are generated from coarse grids, it can be challenging to control the resolution of the

fine level grids according to the users' demands. In other words, coarse grids are not suitable for the starting grid.

To overcome these challenges, several methodologies that enable automatic generation of coarse grids from the fine level have been proposed. Guillard[14] pro-

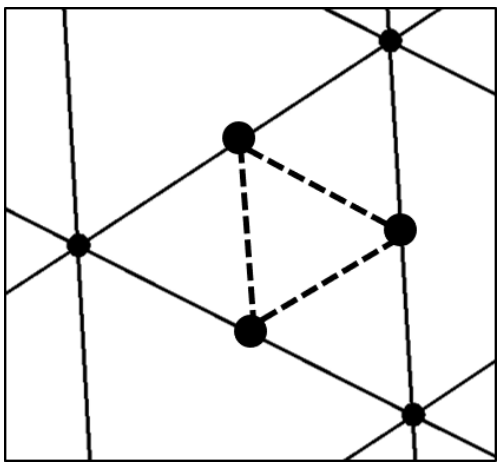


Figure 1.2: subdivision of triangle cell

posed coarsening strategy using Delaunay-Voronoi algorithm. The selected nodes from fine grids are used as coarse level nodes. Subsequently, coarse level volumes are regenerated from the selected nodes using the Delaunay-Voronoi method as shown in Figure 1.3. This re-meshing algorithm was expanded to three-dimensional space by Adams[15]. Chan and Smith[16] suggested the retriangulation approach. In the retriangulation approach, The selected vertices of fine level grid are defined as the coarse level then coarse level triangulation is generated by the subset of vertices of fine level grids. In edge collapse approach by Grumpton and Giles[17], a coarse volume is generated by collapsing edge as shown in Figure 1.4. Finally, grid agglomeration algorithm is the most widely used technique owing to its versatility and automation for three-dimensional grids. The grid agglomeration merge fine level volume to coarse

level volume as shown in Figure 1.5. Various automatic merging algorithms have been studied by many researchers[18, 19, 20, 21, 22, 23, 24, 25].

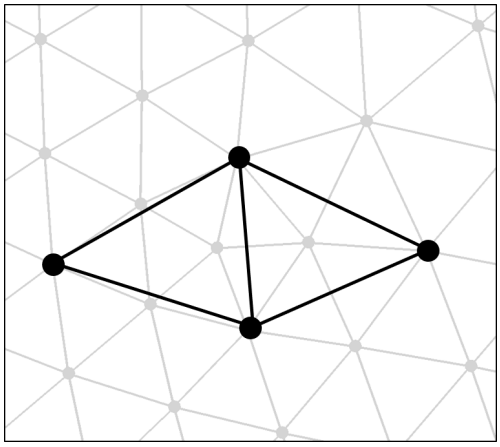


Figure 1.3: Retriangulation

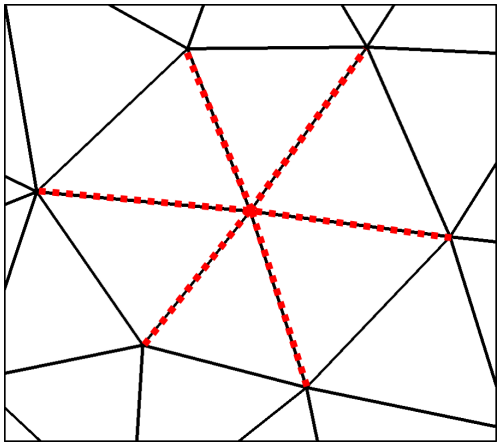
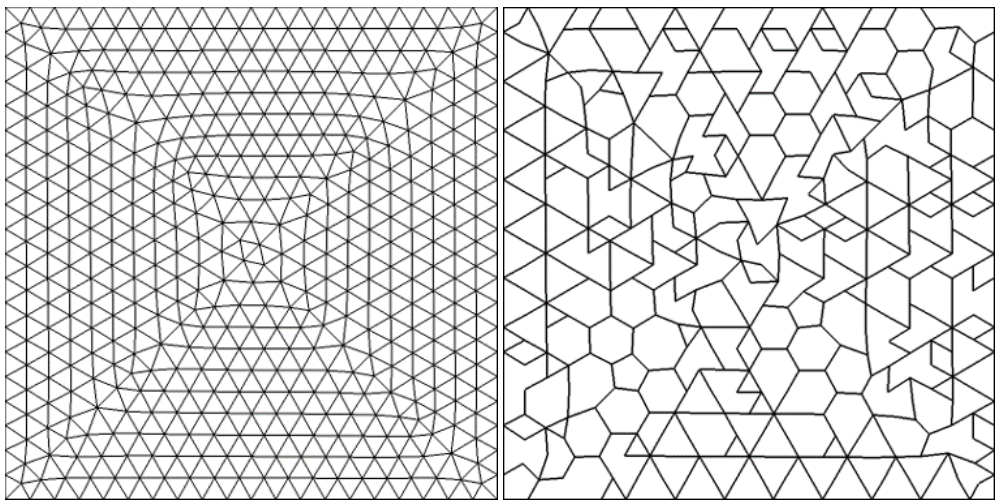


Figure 1.4: Edge collapsing



(a) (b)

Figure 1.5: The original grid(a) and its agglomerated grid(b)

1.2 Multicloud algorithm

Although many advanced coarsening strategies for unstructured geometric multigrid methods have been proposed, they may not be free from problems induced by mesh complexity[22] such as solution discrepancy or instability issues. Even though agglomeration can automatically yield a coarse-level grid, additional treatment is necessary to effectively handle the multi-faced features of the coarse volumes. Furthermore, grid agglomeration methods are significantly challenged to form high-quality coarse grids and The range of available numerical algorithms for fine level handling of arbitrary polyhedra has been limited. To overcome the mesh-induced problems, a new straightforward geometric multigrid method was proposed by Katz and Jameson[26, 27] which is called multicloud method. The multicloud method takes advantage of meshless discretization on coarse level domains. Therefore, the multicloud method only requires a selection process to determine the validity of points, making

the coarsening procedure simpler and more efficient. Unlike traditional mesh-based methods, the meshless approach allows for the coarsening process to skip additional mesh treatment steps such as remeshing or edge fusing. As a result, the coarse-level domains consist only of points, reducing the computational cost and complexity of the method. Thus, the multicloud method is available for any type of method such as cell finite volume(CFV), nodal finite volume(NFV), and meshless schemes. The detailed description of multicloud coarsening strategy will be discussed at the further section.

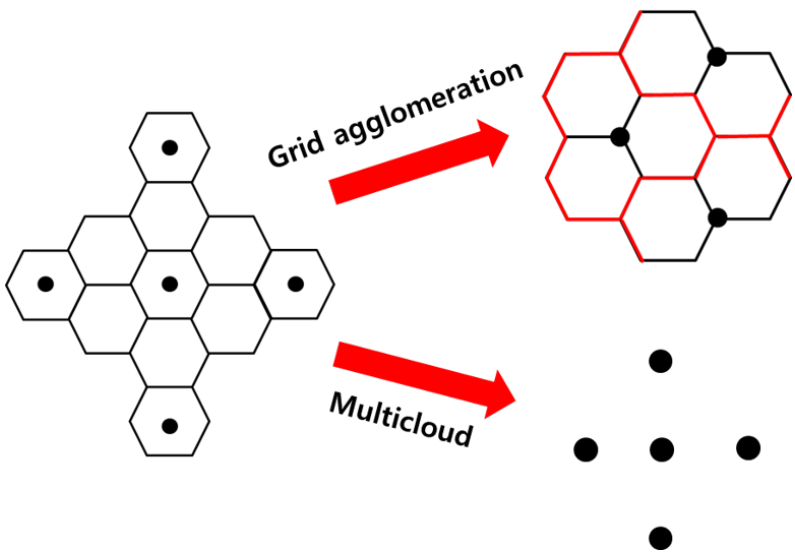
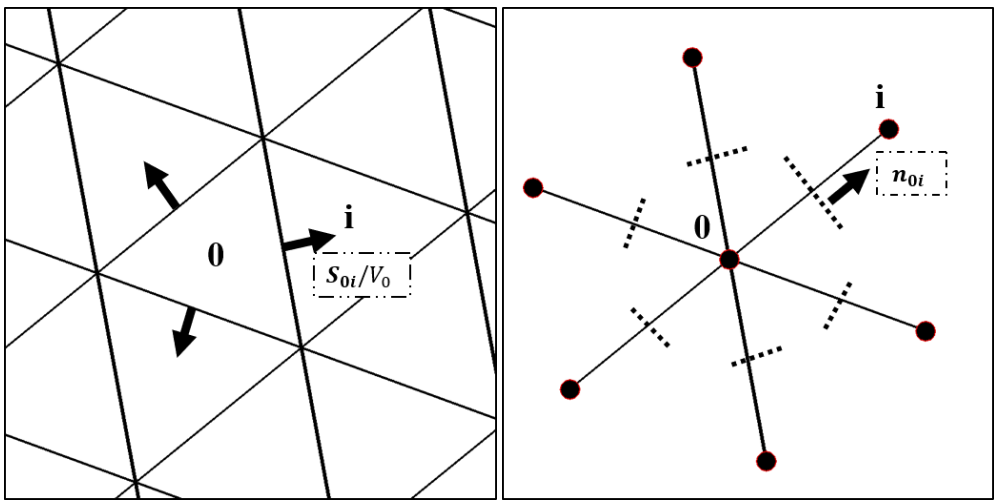


Figure 1.6: The difference between multicloud and grid agglomeration

1.3 Meshless discretization

Meshless methods(also known as gridless, meshfree, gridfree and, etc) are discretization strategy for partial derivatives. Unlike traditional CFD methods that rely on a grid to discretize the computational domain, meshless methods do not require a predefined



(a) (b)

Figure 1.7: The difference between Cell finite volume method(a) and meshless method(b)

grid. Instead, they use a set of discrete points that are distributed in the computational domain to approximate the fluid flow field. For cell finite volume(CFV) methods fluxes are computed on the cell interface but the midpoint on the edge between two points is used flux computation for meshless methods as shown in Figure 1.7.

In the early 1980s, Smooth particle hydrodynamics(SPH), a Lagrangian method, was introduced by Monaghan[28]. In SPH, the fluid is represented as discrete points interacting each other by the certain physical laws. SPH approximation can be obtained by kernel function defined by nearby particles. SPH usually used in astrophysics, oceanography, volcanology which have complex boundary dynamics.

Meshless methods based on radial basis functions(RBF) are also widely used meshless method. RBF is initially used for interpolation by scattered data[29]. Then, RBF was further applied in solving PDEs by Kansa[30].

Least squares method(LSM) is popular in solving traditional CFD problems. LSM approach was first suggested to formulate interpolation function by Shepard[31]. Then, LSM method usually used to estimate gradient for reconstruction[32]. Excellent works in solving PDE using meshless methods have been suggested. Ghosh and Deshpande[33] used Least Squares Kinetic Upwind Method(LSKUM) to numerically analyze compressible inviscid flows. Sridar and Balakrishnan[34] presented a new Least Squares based Upwind Finite Difference(LSBUFD) method in solving Euler equations. Furthermore, Geometric Conservation Least Squares Method(GC-LSM) was presented by Huh[35] which can compute compressible flows robustly and accurately.

1.4 Motivation

Although the multicloud method appears to have significant potential for three-dimensional applications, a few following studied have been presented on simple grids. Zamolo[36] applied the multicloud method to the Poisson equation with new correction and restriction algorithms. The method was developed based on RBF discretization for all levels. Radhakrishnan[37] also presented the multicloud application to the Poisson equation. Barik[38] applied the multicloud method to incompressible Navier-Stokes equations on two-dimensional Cartesian grid. Ha and Choi[39] developed meshless multigrid for finie element methods. However, practical usage of mutlicloud method for three-dimensional CFD problems has not been presented. Considering the potential of the multicloud method, it is necessary to develop the three-dimensional multicloud method for CFD.

Most of all, the primary challenge of the three-dimensional multicloud method

is unsatisfactory acceleration for cell-centered methods[27], which is most widely used CFD method. In order to enhance the acceleration effect, a new cell coarsening method will be described, which makes coarse grid coarser than the ideal coarsening rates since the grid size is bigger the errors are damped quickly. Next, the process of finding an appropriate meshless discretization that works robustly on extremely coarsened domains will be presented since conventional meshless methods are likely to incurs problems on the extremely coarsened domains.

Then, implicit multicloud method is also to be presented. The original multicloud method was tested based on explicit Runge-Kutta 4th order time integration method. Three-dimensional problems are usually involved with a large number of grids. Consequently, explicit time integration method might not be appropriate for practical problems since it spent a lot of time in obtaining converged solutions. Therefore, implicit algorithm for multigrid methods is interest of many researchers[23, 40].

Subsequently, damping functions for multicloud operators will be described in order to apply the multicloud method to problems involving hypersonic shock.

Finally, the proposals, three-dimensional multicloud method is developed in this study. The developed method is tested for nodal-centered and cell-centered method. Comparisons for explicit and implicit time schemes are presented at first. Subsequently, the comparison of acceleration effects based on different coarsening strategies and meshless methods will be presented to identify the most effective approach. Furthermore, numerical experiments are tested from simple geometries such as ONERA-M6 wing to significantly complex geometries such as DLR-F6 wing body nacelle pylon model[41] in order to highlight the efficacy of the proposed method.

Chapter 2

Multicloud

2.1 Coarsening Strategy

A multicloud coarsening strategy is performed in a meshless fashion. A fundamental idea is the usage of local point clouds. The local point cloud is the set of nearest point as shown in Figure 2.1. The multicloud coarsening strategy is described in Algorithm 1.

The local point cloud shown in Figure 2.1 is used for not only the next level coarsening procedure but also meshless discretization on the coarse level.

Therefore, the multicloud coarsening procedure can be adopted if initial local point clouds are established. The methodology of defining initial local point clouds is as follows

2.1.1 Node coarsening strategy

For node-centered methods, governing equations are solved at each node(a vertex of a cell). The edge between nodes act as interfaces which fluxes across. Such that

Algorithm 1 Original multicloud coarsening

For each point v_k, Set $\Gamma(v_k)$ = .True., where Γ denotes logical function which defines validity on the next coarse level

Define the set $L_1(v_i) = \{v_k | v_k$ is nearest point around $v_i\}$ which is called local point cloud as shown in Figure 2.2a

while $i \leq N_c$ **do**, where N_c is the number of computational points

  if $\Gamma(v_i)$ is .True. **then**

    $\Gamma(L_1(v_i))$ =.False.

  end if

end while

Figure 2.2b presents the resulting coarse level point cloud. Points whose $\Gamma(v_i)$ =.False. are filled with nothing

Form coarse level local clouds $L_1^{k+1}(v_i)$ for each $\Gamma(v_i)$ =.True.. Figure 2.2c represents the local point cloud on the coarse level.

Perform the coarsening through coarse level local point clouds until the coarsest level is reached

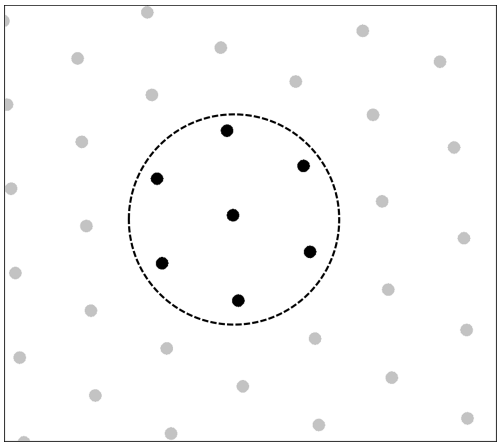
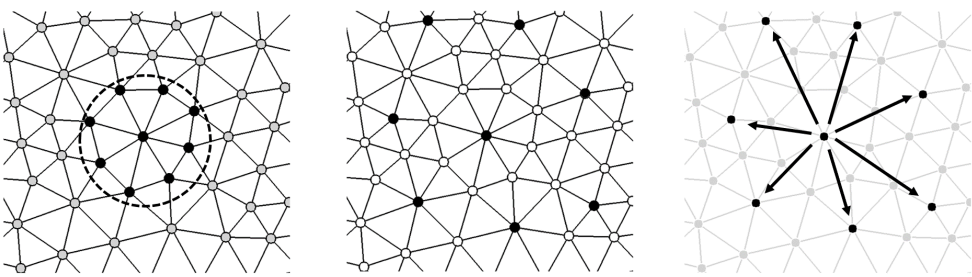


Figure 2.1: The definition of a local point cloud



(a) The local point cloud (b) validity determination (c) Coarse level clouds

Figure 2.2: The multicloud coarsening procedure

nearest neighbors can be composed by nodes forming edges as shown in Figure 2.3.

2.1.2 Cell coarsening strategy

As long as the local point cloud $L_1(v_i)$ is defined, the coarsening strategy can be applied to any type of method, such as meshless, NFV, and CFV. In the original study, the coarsening procedure was described for node-centered method whereas cell coarsening strategy was unclear. Thus, a cell coarsening procedure should be

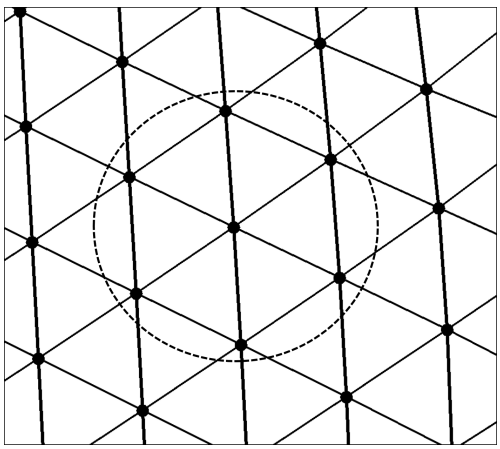


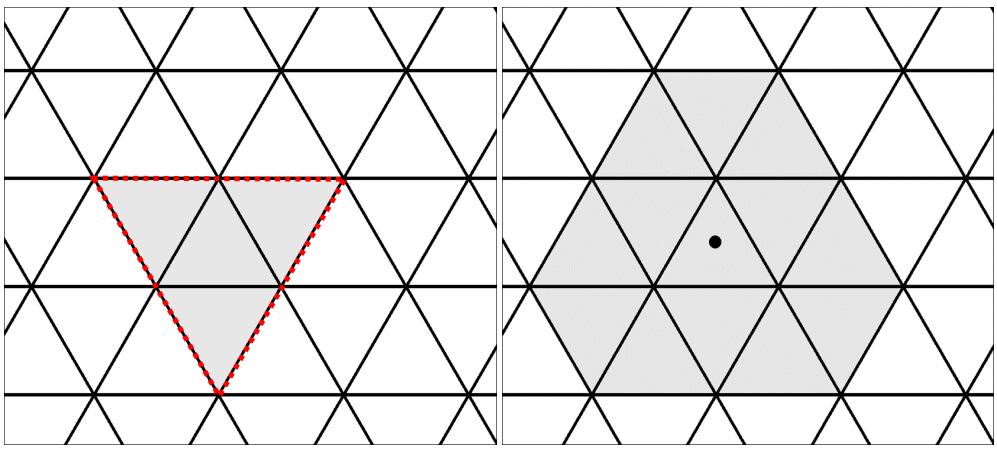
Figure 2.3: The definition of a initial local point cloud for node-centered methods

established for cell-centered method. For this reaosn $L_1(v_i)$ may be defined as the adjacent cells, as shown in Figure 2.5a for CFV, because the nodes that form the edge in NFV correspond to interfacing cells in CFV. The use of common face cells is named as the common face approach (CFA). The multicloud coarsening procedure, which is performed based on L_1 obtained from CFA, yields the results shown in Figure 2.5c. The resulting point distribution, however, produces an excessive number of points on the coarse level, which significantly disagreed with the ideal coarsening rates. The insufficient amount of neighboring cell information included in the definition of the local point clouds might be the reason for the excessive point distribution on the coarse level. For node coarsening, theoretically, a node for triangular grids can be connected to at least four nodes to an infinite number of nodes allowing for adequate collection of nearest point information. On the contrary, the number of interfacing cells is strictly limited by the type of polyhedra, as shown in Figure 2.5a. The limitation obviously degrades coarsening rates for cell coarsening. Therefore, CFA might not be suitable for the concept of local point clouds in coarsening, due

to the lack of information. Unsatisfactory coarsening rates hinder acceleration and increase computational time for multicloud operators.

Consequently, despite its convenience, multicloud for CFV may not be suitable for practical CFD problems. To address the issue of insufficient size of local point clouds, the nodes of each cell is utilized. When defining the local point cloud for an arbitrary cell c_i in coarsening, nodes of a cell is used to find nearest cells. Specifically, as shown in Figure 2.5b, every cell that shares nodes with c_i is collected to form the local point cloud of c_i. This approach is referred to as the 'common node approach' (CNA), where the local point cloud for an arbitrary cell $L_1(c_i)$ is defined by collecting all cells that share the nodes of cell c_i. By defining this new local point cloud, the same coarsening strategy can be used without any further modifications. The resulting point distribution is shown in Figure 2.5d. Compared to the CFA coarsening, it appears that far fewer points remain with the CNA coarsening strategy. Furthermore, unlike the CFA method, the CNA method ensures even point distribution by taking into account the neighboring cells precisely. Furthermore, The average distance for nearest points, which corresponds to grid size, is much farther for CNA than that of CFA as seen in Figure 2.5f and 2.5e. In general, The larger grid size grants the more error damping if coarse level discretization is carried out robustly. CNA coarsening can be compared to ideal agglomeration straightforwardly as shown in Figure 2.4. For ideal agglomeration coarsening, four cells merged to a cell as shown in Figure 2.4a, whereas a cell center can remove neighboring twelve cells in CNA coarsening as shown in Figure 2.4b. It denotes that CNA coarsening yields overly coarsened domain than that of ideal coarsening, which provides superior acceleration effect if coarse level discretization is performed robustly.

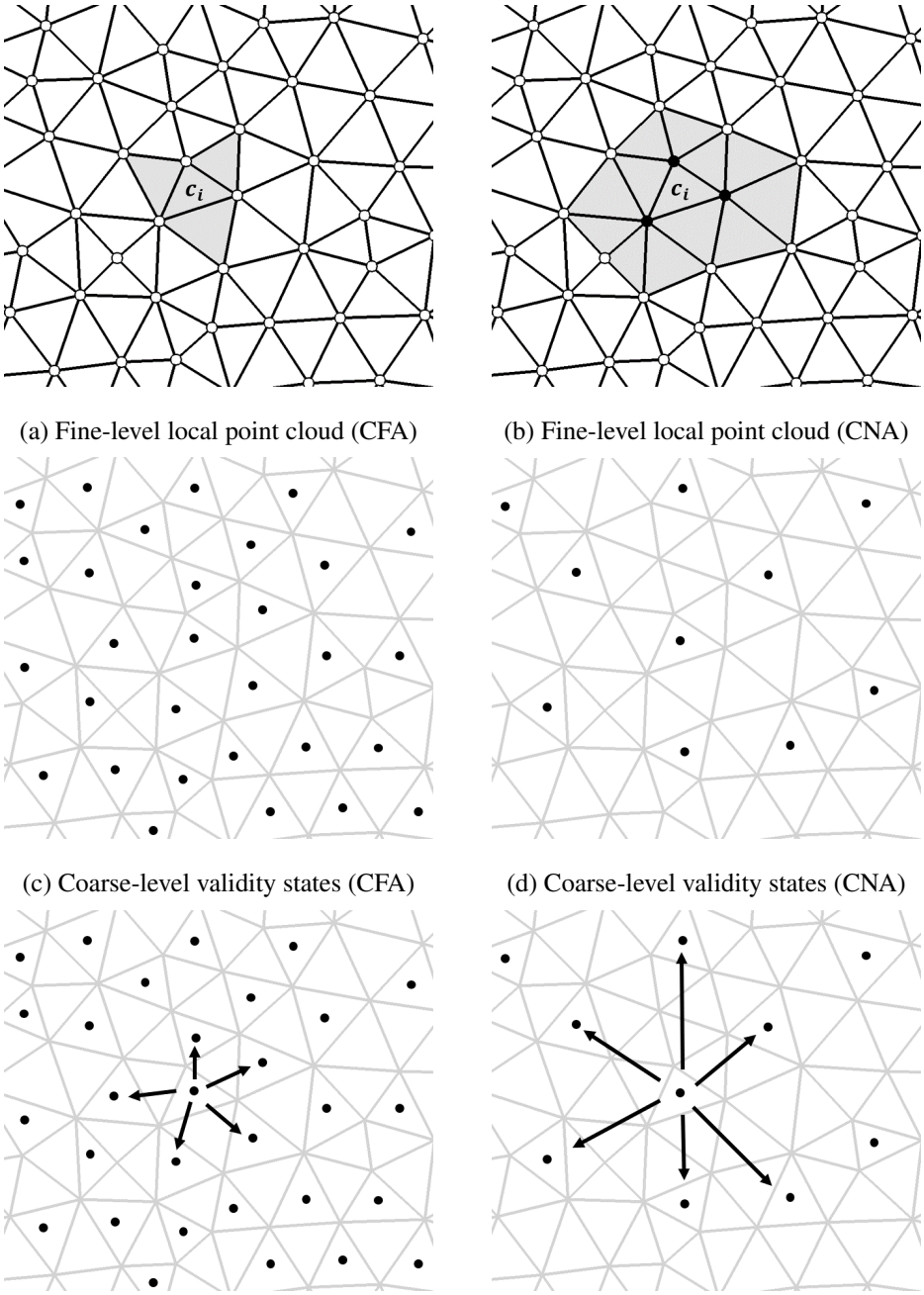
To compare the coarsening procedures for an practical case, the results obtained



(a) Ideal coarsening rates (b) CNA coarsening rates

Figure 2.4: Comparison between ideal and CNA coarsening

from CNA, CFA, and agglomeration methods were presented. Several agglomeration methods have been proposed in the literature [25, 42, 19], but for our study, the agglomeration method proposed by Jones and Vassilevski[19] was evaluated specifically. Figure 2.6a shows the finest triangular grid, while Figures 2.6b, 2.6d depict an agglomerated grid based on the algorithm proposed by Jones and Vassilevski[19], and coarsened point distributions obtained using CNA and CFA, respectively. As can be seen from Figure 2.6, CFA coarsening results in the least coarsened domain, whereas agglomeration and CNA provide approximately ideal coarsening rates. In addition, CNA may result in not only the most well-distributed but also the most coarsened domain for the next level because Jones and Vassilevski's agglomeration procedure may be dependent on the indexing order due to its element selection algorithm based on integer weight[19]. Furthermore, the results obtained from various meshless methods and coarsening strategy will be presented in this study.



(a) Fine-level local point cloud (CFA)

(b) Fine-level local point cloud (CNA)

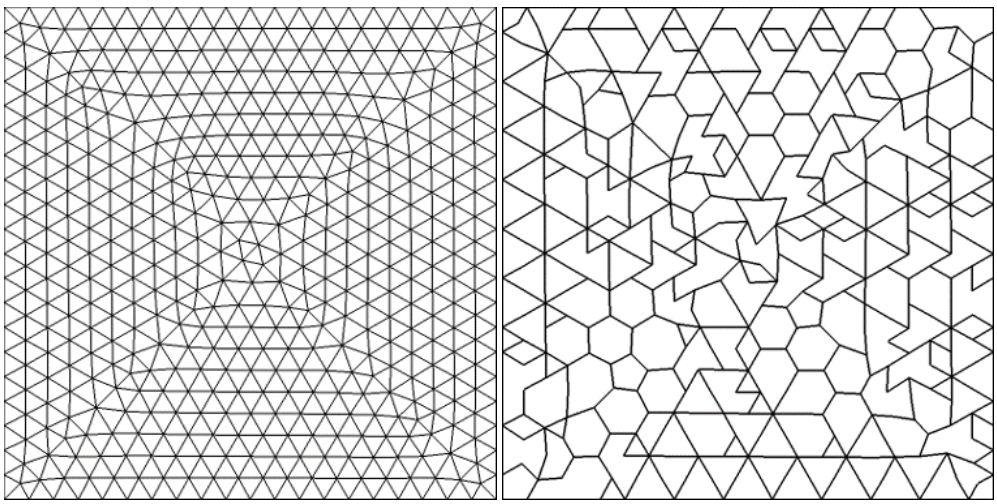
(c) Coarse-level validity states (CFA)

(d) Coarse-level validity states (CNA)

(e) Coarse-level local point cloud (CFA)

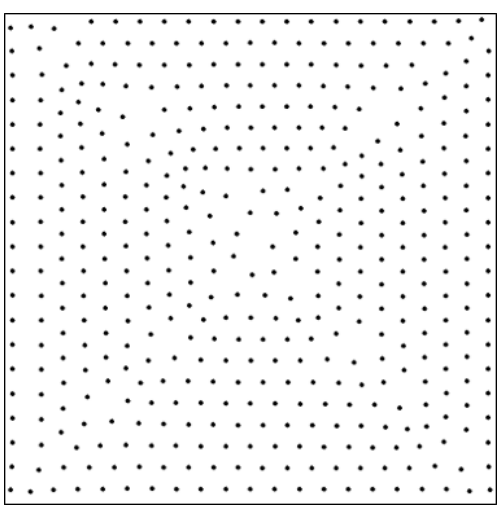
(f) Coarse-level local point cloud (CNA)

Figure 2.5: Cell coarsening procedures for each method

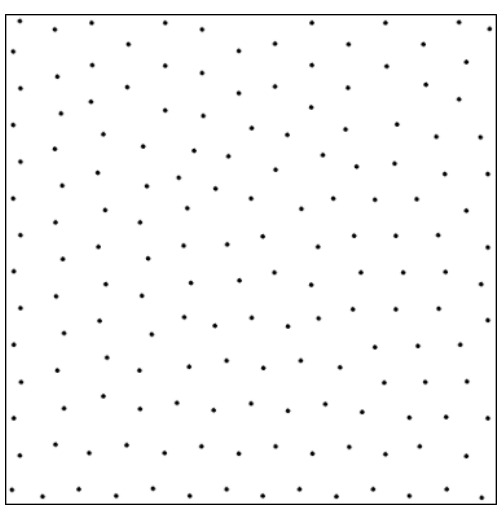


(a) Finest grid (924 cells)

(b) Level 2 by Jones[19] (218 cells)



(c) Level 2 by CFA (446 points)



(d) Level 2 by CNA (162 points)

Figure 2.6: Coarsening results comparison

2.1.3 Directional coarsening

High-aspect-ratio grids are necessarily involved when solving high Reynolds number flows. However, Su[43] reported that meshless methods can encounter misalignment issues on highly skewed grids resulting from high-aspect-ratio geometries. The issues cause serious stability problem to the multicloud method for high-aspect-ratio grids. Generally, aspect-ratio that is higher than 200 exhibit instability by empirical data. For the previous mutligrid method, the problems cause by high-aspect-ratio grids requires additional treatments such as implicit line relaxation and directional coarsening[22].

In this study, Directional coarsening strategy was developed for the multicloud method in order to address the issues resulting from high-aspect-ratio grids. In this strategy, a layer structure for viscous region must be predefined such as prism. The layer structure is prerequisite in solving high Reynolds flows. In Directional coarsening, viscous layer coarsening should be prioritized. Then, viscous layer coarsening is carried out only along the normal to surface direction in order to alleviate high-aspect-ratio. In order to coarsen grid in a unidirectional way, viscous grids should be labelled by its marching direction. Layer cells are assigned their layer number along their marching direction. The remaining cells are assigned their layer number as negative 1. Figure 2.8 illustrates the assignment procedure. Then, a rule is established that cells with positive layer numbers cannot be blanked by other cells. With the established rule, the directional coarsening procedure follows the Algorithm 2.

Algorithm 2 with the layer number rule results in Figure 2.9. By directional coarsening, only marching direction coarsening is carried out, and then the layer structure is preserved, which alleviates grid stretching.

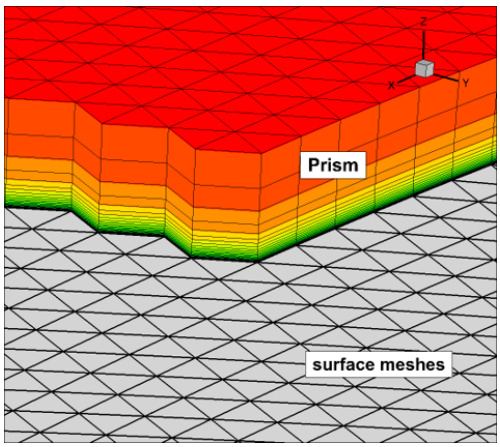


Figure 2.7: Prism grids for high Reynolds number flow

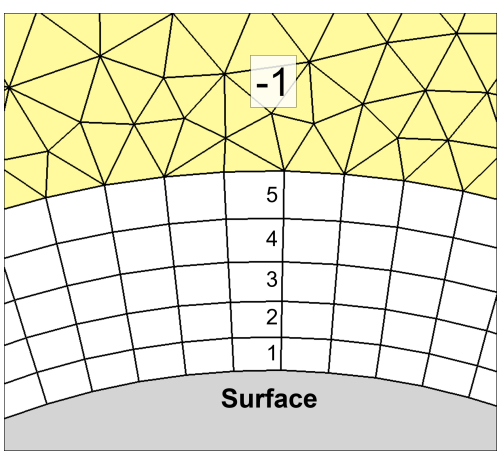


Figure 2.8: Illustration of id labelling

Algorithm 2 Directional coarsening

1: Assign layer number for all cells.

2: blank cells whose layer number does not equal to $2^k n - 1$ $(n = 1, 2, 3, ...)$.

3: Perform the general coarsening procedure for cells whose layer number is maximum or negative.

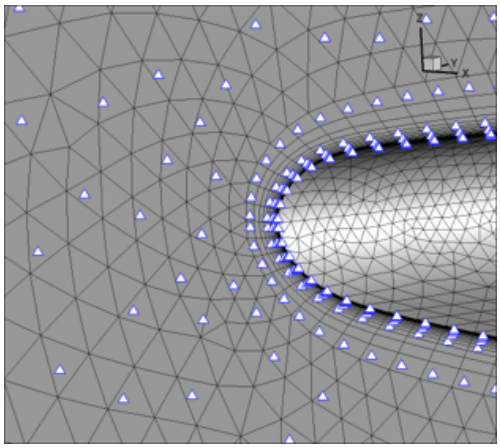
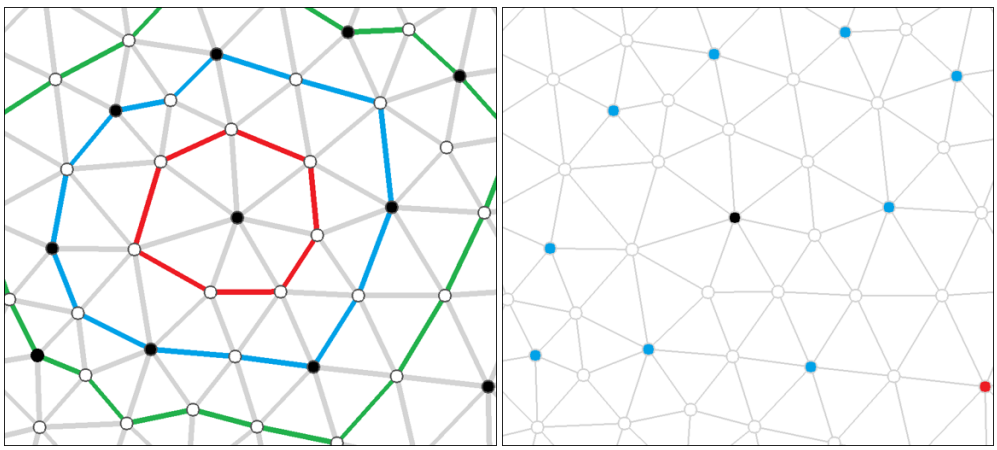


Figure 2.9: Resulting coarse level point by directional coarsening



(a) Range of searching (b) Resulting coarse level local point cloud

Figure 2.10: Illustration of procedure of forming coarse level local point cloud

2.1.4 Composing coarse level local point clouds

After coarsening, local point clouds for the next coarse level must be constructed. Searching algorithm is can be simply established by adopting a meshless fashion. Similar to coarsening procedure, fine level local point clouds are used in finding coarse level local point clouds. For an arbitrary coarse level point, searching process is performed by a radial fashion. the points on its local point cloud are the first candidates as indicated as red edge in Figure 2.10a. Then, the points on their local point cloud as indicated as blue edge in Figure 2.10a are the second candidates. Finally, the last scanning is performed from the second candidates as indicated as in green edge in Figure 2.10a. From the candidates, All the valid point are collected as a coarse level local point cloud as shown in Figure 2.10b. Constructed local point clouds are taken advantage of every procedure in multicloud such as meshless discretization, coarsening and forming coarse level local point clouds.

2.2 Multicloud Operator

The Full Approximation Stroage (FAS) algorithm of Brandt[2] is applied to the multicloud method. The FAS algorithm is described in Algorithm 3. An advantage of the Multicloud method in applying the FAS algorithm is that solution interpolation is not required for coarse-level points. Considering retriagulation as shown in Figure 2.11a, the state variables of coarse cell must be interpolated by its neighboring cells. On the contrary, in the multicloud method, the state variables of coarse points can be directly transferred from fine level coincident cell-averaged or nodal values without the need for solution interpolation as shown in Figure 2.11b.

Algorithm 3 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 $\boldsymbol{q}_0^{k+1}$.

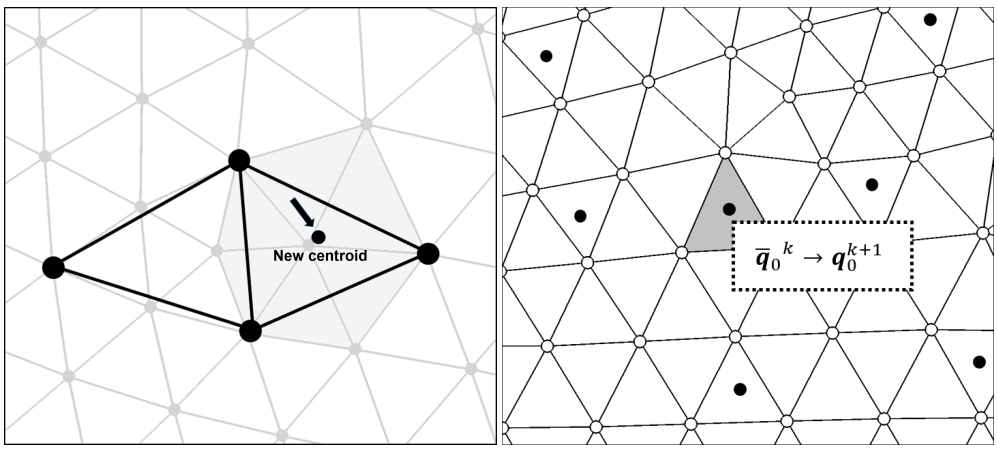
2: Compute forcing function $\boldsymbol{P}_m$ using fine-level residuals of nearest points as in Eq.(2.1).

3: Perform iteration on the coarse level with restricted residuals $\boldsymbol{R}^{k+1} + \boldsymbol{P}_m$

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 Figure 2.12b.

For the multicloud method, restriction and prolongation are performed in a meshless fashion such that neither volume nor surface information is used. Residual forcing function for multicloud is given as

$$\boldsymbol{P}_m = \boldsymbol{T}_R \boldsymbol{R}_{k-1}(\boldsymbol{q}_{k-1}) - \boldsymbol{R}_k(\boldsymbol{q}_k^{(0)}) \tag{2.1}$$



(a) Illustration of solution interpolation (b) Illustration of solution transfer

Figure 2.11: Illustration of procedure of forming coarse level local point cloud

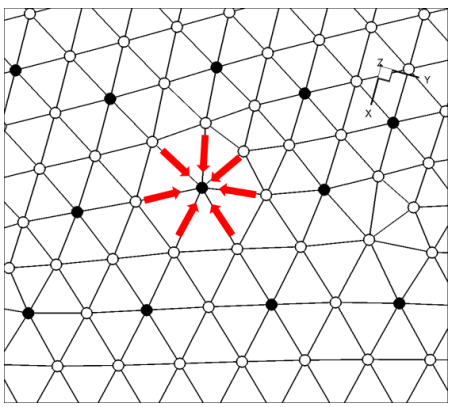
where,

$$\boldsymbol{T}_R^T = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_N \end{bmatrix}, \boldsymbol{R} = \begin{bmatrix} R_{0,1} & R_{0,1} & \dots & R_{0,m} \\ R_{1,1} & R_{1,2} & \dots & R_{1,m} \\ \vdots & \vdots & \ddots & \vdots \\ R_{N,1} & R_{N,2} & \dots & R_{N,m} \end{bmatrix} \tag{2.2}$$

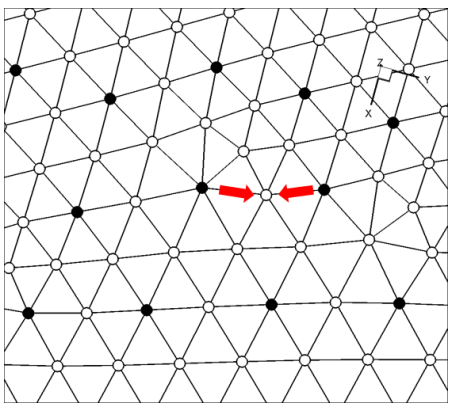
In Eq.(2.2), β is given as a distance-based function since multicloud doesn't require grid information. Those multicloud operators are demonstrated to exhibit a high degree of efficacy[26]. The functions are used only with minor modifications to consider new coarsening and three-dimensional space in this study. Then, β is given as

$$\beta_i = \begin{cases} \left(\frac{1-\beta_0}{\sum_{k \in L_1(0)} c_k} \right) c_i & \text{if } i \neq 0 \\ (ds_k / ds_{k+1})^{\mathrm{d}} & \text{if } i = 0 \end{cases} \tag{2.3}$$

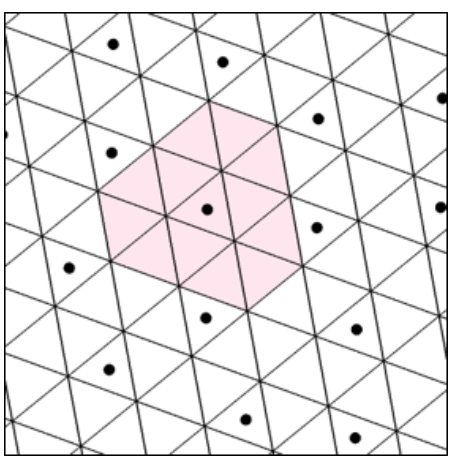
where,



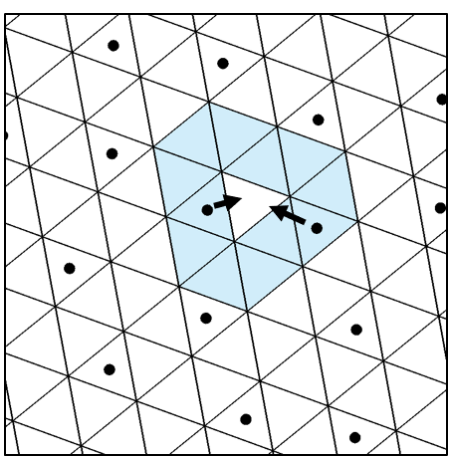
(a) restriction stencils (NFV)



(b) prolongation stencils (NFV)



(c) restriction stencils (CFV)



(d) prolongation stencils (CFV)

Figure 2.12: Illustration of stencil for multicloud operators

$$c_i = \begin{cases} \mid \boldsymbol{r}_i - \boldsymbol{r}_0 \mid^{-1} & \text{if } i \neq 0 \\ 1 & \text{if } i = 0 \end{cases} \tag{2.4}$$

$L_1(0)$ is the set of discrete points in its fine level local point cloud as shown in Figure 2.12a. In CNA coarsening, $L_1(0)$ would be expanded as shown in Figure 2.12c. Such that, multicloud operators are independent of coarsening procedures. In Eq.(2.3), ds is average length between point 0 and its local point cloud. Such that β_0 is cube of the ratio of average length between fine and coarse levels. Since β_0 is similar function to volume ratio of conventional multigrid methods, the cube of ds ratio should be considered for three-dimensional flows.

In Eq.(2.3), β_i is exactly same with the original multicloud method. As seen in Eq.(2.4), c_i is inverse distance between point 0 and i.

Then, the solutions are updated by relaxation on the coarse level. The updated solutions correct fine level solutions. In the multicloud method, coarse level state variables are directly transferred to coincident fine level points, which is called "inherited point", such that only fine level points whose coincident coarse level points are nonexistent, which is called "orphan point", should be corrected.

The state variables of orphan points are corrected by its neighboring inherited points that are located in $L(0)$ as shown in Figure 2.12b and 2.12d. Then solutions are corrected by inverse distance weighting.

$$q_0^+ = q_0 + \frac{\sum_{i \in L(0)} c_i (q_i^+ - q_i^{(0)})}{\sum_{i \in L(0)} c_i} \tag{2.5}$$

2.3 multicloud operators for hypersonic flows

Multicloud operators cannot be directly applied to hypersonic flows since standard multigrid methods leads to divergence near strong shock formed by hypersonic flows. In order to avoid the divergence, various studies have been suggested. Kim[6] proposed prolongation damping using pressure based shock sensing method. Radespiel[44] presented restriction damping method by detecting shock using pressure. Gerlinger[45] modified Radespiel's restriction damping using TVD and pressure based sensor combined with chemical reaction sensor for analysis of turbulent combustion. However, the damping methods were formulated based on structured grids such that the methods cannot directly be implemented to unstructured multigrid or multicloud method. In this section, new damping formulations for multicloud method are described to apply multicloud method to problems with strong shock.

2.3.1 Damped restriction

Locally damped restriction method was developed to prevent divergence near strong shock. Damped restriction for fully coarsened structured methods is given as

$$\boldsymbol{R}^{k-1} = \boldsymbol{T}_{k-1,k}\boldsymbol{R}^{k} = \sum_{l=1}^{4} \boldsymbol{R}_{l}^{k} \max\left[0, \max\left(1-\kappa_{l}\right)\right] \tag{2.6}$$

As seen in Eq.(2.6), residual of coarse cell can be expressed as sum of damped residuals of parental fine cells as shown in Figure 2.13. κ_l in Eq.(2.6) denotes shock sensor function which is defined as

$$\kappa = \max\left(\nu_{i,j}^{\xi}, \nu_{i-1,j}^{\xi}, \nu_{i+1,j}^{\xi}, \nu_{i,j}^{\eta}, \nu_{i,j-1}^{\eta}, \nu_{i,j+1}^{\eta}\right) \tag{2.7}$$

with

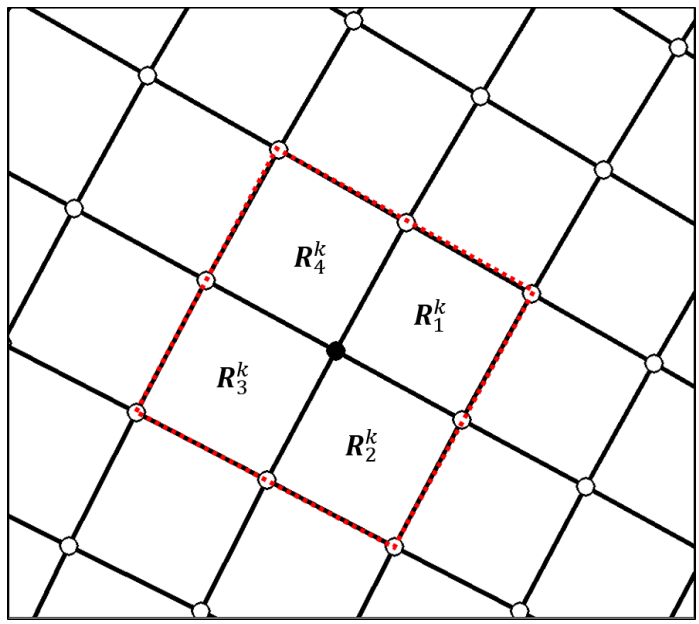


Figure 2.13: Restriction stencil for structured grids

$$\nu_{i,j}^{\xi} = \frac{|p_{i+1,j} - 2p_{i,j} + p_{i-1,j}|}{(1-\chi)\left(|p_{i+1,j} - p_{i,j}| + |p_{i,j} - p_{i-1,j}|\right) + \chi\left(p_{i+1,j} + 2p_{i,j} + p_{i-1,j}\right)} \tag{2.8}$$

where χ is values between 0.8 and 1. Eq.(2.8) is mixture between pressure-based and TVD-based shock sensor. Consequently, residual is less transferred where strong shock is located between cells by Eq.(2.6). Since $i-1$ and $i+1$ cells are not defined for both unstructured and meshless methods, new formulation is required for damped restriction. The new meshless damped restriction is given as

$$\boldsymbol{P}_m = \boldsymbol{T}_R^{\kappa}\boldsymbol{R}_{k-1}(\boldsymbol{q}_{k-1}) - \boldsymbol{R}_k(\boldsymbol{q}_k^{(0)}) \tag{2.9}$$

with

$$\boldsymbol{T}_R^{\kappa} = \begin{bmatrix} \beta_0 & \kappa_1\beta_1 & \cdots & \kappa_N\beta_N \end{bmatrix} \tag{2.10}$$

where

$$\kappa_i = \min\left(1, \frac{P_0}{P_i}\right) \tag{2.11}$$

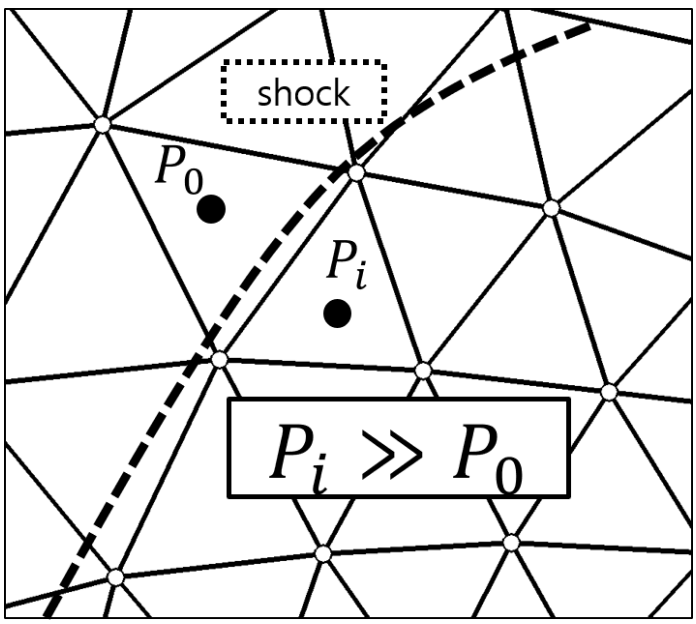


Figure 2.14: Illustration of meshless damped restriction

As shown in Figure 2.14, if strong shock is located between 0 and i, the residual of i rarely transfers the residuals to point 0. Initial instability can be cured by simply introducing damping function[45].

2.3.2 Damped prolongation

In order to converge solution near strong shock, damped prolongation is also necessary. For stablization of the multicloud method, prolongation damping proposed by kim[6] and Zhu[7] is formulated in a meshless fashion. As seen in Figure 2.15, P_0 is interpolated by nearest points P_1^+, P_2^+, P_3^+. But P_2^+ is located at the postshock position. Thus, it is desirable to damp the interpolation value from P_2^+ to avoid negative

pressure. To search for points where are located at the otherside of the shock wave, pressure weighted function is used

$$\omega_{p,i} = \min\left(1, \frac{P_0}{P_i^+}\right) \tag{2.12}$$

Then, prolongation operator with damping is as follows

$$q_0^+ = q_0 + \frac{\sum_{i \in L(0)} \omega_{p,i} c_i (q_i^+ - q_i^{(0)})}{\sum_{i \in L(0)} c_i} \tag{2.13}$$

From Eq.(2.12) and (2.13), $\omega_{p,i}$ goes to 0, where abrupt pressure change occur.

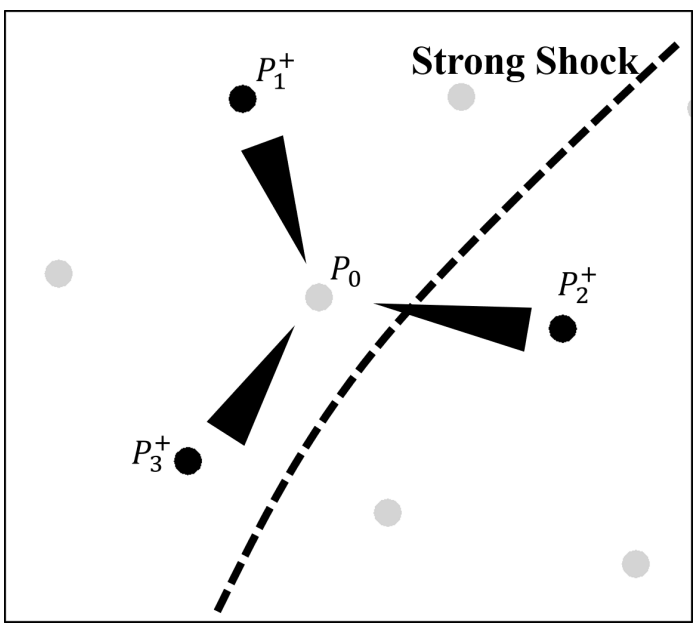


Figure 2.15: Illustration of shock sensing

Chapter 3

Numerical methods

3.1 Governing equations

3.1.1 Navier-Stokes equations

In this study, three-dimensional Navier-stokes equations were considered to verify the effect of an implicit multicloud convergence accelerator. The equations are expressed as follows:

$$\frac{\partial \boldsymbol{q}}{\partial t} + \frac{\partial \boldsymbol{f}}{\partial x} + \frac{\partial \boldsymbol{g}}{\partial y} + \frac{\partial \boldsymbol{h}}{\partial z} = \frac{M_\infty}{Re_\infty}\left(\frac{\partial \boldsymbol{f}_v}{\partial x} + \frac{\partial \boldsymbol{g}_v}{\partial y} + \frac{\partial \boldsymbol{h}_v}{\partial z}\right) \tag{3.1}$$

where $\boldsymbol{q}$ represents the set of conservative variables $\begin{bmatrix} \rho & \rho u & \rho v & \rho w & \rho E \end{bmatrix}^T$, and the convective fluxes $\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{h}$ are expressed as

$$\boldsymbol{f} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ \rho uH \end{bmatrix}, \boldsymbol{g} = \begin{bmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vw \\ \rho vH \end{bmatrix}, \boldsymbol{h} = \begin{bmatrix} \rho w \\ \rho wu \\ \rho wv \\ \rho w^2 + p \\ \rho wH \end{bmatrix} \tag{3.2}$$

In Eq.3.2, p is the pressure and ρ, 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,

$$E = \frac{p}{(\gamma - 1)\rho} + \frac{1}{2}\left(u^2 + v^2 + w^2\right) \tag{3.3}$$

Then, the viscous fluxes are

$$\begin{aligned}
\boldsymbol{f}_v &= \begin{bmatrix} 0 & \tau_{xx} & \tau_{xy} & \tau_{xz} & u\tau_{xx} + v\tau_{xy} + w\tau_{xz} - Q_x \end{bmatrix}^T \\
\boldsymbol{g}_v &= \begin{bmatrix} 0 & \tau_{yx} & \tau_{yy} & \tau_{yz} & \tau_{yx} + v\tau_{yy} + w\tau_{yz} - Q_y \end{bmatrix}^T \\
\boldsymbol{h}_v &= \begin{bmatrix} 0 & \tau_{zx} & \tau_{zy} & \tau_{zz} & \tau_{zx} + v\tau_{zy} + w\tau_{zz} - Q_z \end{bmatrix}^T
\end{aligned} \tag{3.4}$$

where the shear stress and heat flux terms are

$$\begin{aligned}
\tau_{xx} &= 2\mu u_x + \lambda(u_x + v_y + w_z) \\
\tau_{yy} &= 2\mu v_y + \lambda(u_x + v_y + w_z) \\
\tau_{zz} &= 2\mu w_z + \lambda(u_x + v_y + w_z) \\
\tau_{xy} &= \tau_{yx} = \mu(u_y + v_x) \\
\tau_{xz} &= \tau_{zx} = \mu(u_z + w_x) \\
\tau_{yz} &= \tau_{zy} = \mu(v_z + w_y) \\
Q_x = -\kappa T_x, Q_y &= -\kappa T_y, Q_z = -\kappa T_z
\end{aligned} \tag{3.5}$$

where, T is temperature and effective viscosity μ is given as $\mu_l + \mu_t$, where μ_l and μ_t denote laminar viscosity and turbulent viscosity respectively. Also conductivity $\kappa = \kappa_l + \kappa_t$ The laminar viscosity computed by Sutherland's law and laminar conductivity are given as

where, C_p is the constant pressure specific heat $T_0 = 110.K$ and $Pr_l = 0.72$ are Sutherland's constant and the laminar Prandtl number respectively. In this study, Turbulent viscosity and conductivity are given by Spalart and Allmaras one-equation turbulence model [46, 47]. The one-equation model is also included in multicloud computation

$$\frac{\mu_l}{\mu_\infty} = \left(\frac{T}{T_\infty}\right)^{\frac{3}{2}} \frac{T_\infty + T_0}{T + T_0}, \quad \kappa_l = \frac{C_p \mu_l}{Pr_l} \tag{3.6}$$

3.1.2 Spalart-Allmaras Turbulence Model

In this study, Spalart-Allmaras one equation model is used. Before proceeding to the S-A model, the eddy viscosity $\tilde{\nu}$ is given as follows:

$$\nu_t = \tilde{\nu} f_{v1},\ f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3},\ \chi = \frac{\tilde{\nu}}{\nu} \tag{3.7}$$

The compressible Form of S-A is then as follows

$$\frac{\partial \rho \tilde{\nu}}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} \tilde{\nu}) - \rho(P - D) - \frac{1}{\sigma} \nabla \cdot [\rho(\nu + \tilde{\nu} \nabla \tilde{\nu}] - \frac{c_{b2}}{\sigma} \rho (\nabla \tilde{\nu})^2 + \frac{1}{\sigma}(\nu + \tilde{\nu}) \nabla \rho \cdot \nabla \tilde{\nu} \tag{3.8}$$

where P and D represent production and destruction, respectively, and are defined as follows

$$P = c_{b1}(1 - f_{t2}) \tilde{S} \tilde{\nu}, \quad D = \left(c_{w1} f_{w1} - \frac{c_{b1}}{\kappa^2} f_{t2}\right) \left[\frac{\tilde{\nu}}{d}\right]^2 \tag{3.9}$$

with

$$\tilde{S} = S + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}, \quad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}, \quad S = |\nabla \times \boldsymbol{V}| \tag{3.10}$$

where d is the minimum distance to the wall. Then, the remaining unknowns are given as follows

$$f_w = g\left[\frac{1+c_{w3}^6}{g^6+c_{w3}^6}\right] \quad g = r + c_{w2}(r^6 - r),\ r = \min\left(\frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2}, 10\right) \tag{3.11}$$

Remaining constants are

$$\begin{aligned} c_{b1} = 0.1355,\ \sigma = 2/3,\ c_{b2} = 0.622,\ \kappa = 0.41,\ c_{w1} = c_{b1} \\ \kappa^2 + (1+c_{b2})/\sigma,\ c_{w2} = 0.3,\ c_{w3} = 2 \end{aligned} \tag{3.12}$$

3.2 Spatial discretization

3.2.1 Cell finite volume

Eq. (3.1) may be semi-discretized by cell finite volume methods as follows

$$\frac{\partial \bar{\boldsymbol{q}}_0}{\partial t} + \sum_i^n \left(\boldsymbol{G}_{0i} \cdot \frac{\boldsymbol{S}_{0i}}{V_0}\right) = 0 \tag{3.13}$$

where $\bar{\boldsymbol{q}}$ is cell-averaged state vector $\boldsymbol{S}_{0i}$ and V_0 denote the cell interface area vector and cell volume respectively. Then, $\boldsymbol{G}_{0i}$ can be given by various flux schemes.

3.2.2 Least Squares Method

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

$$\hat{\phi}(\boldsymbol{r}_i) \approx \phi(\boldsymbol{r}_0) + \Delta\boldsymbol{r}_{0i} \cdot \nabla\phi(\boldsymbol{r}_0) \tag{3.14}$$

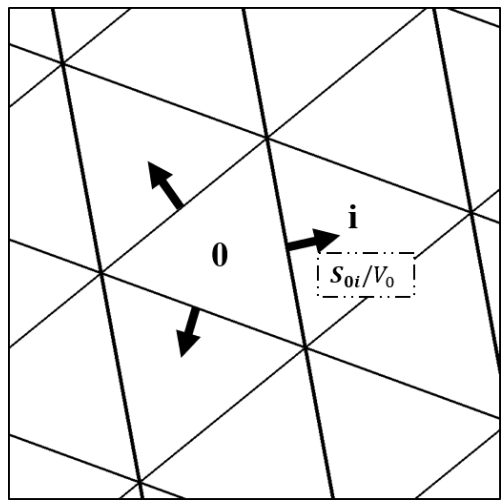


Figure 3.1: Illustration of cell finite volume method

where $\Delta \boldsymbol{r}_{0i} = \boldsymbol{r}_i - \boldsymbol{r}_0$ and Eq. (3.14) may be recast as

$$\Delta \phi_{0i} \approx \boldsymbol{p}^T (\Delta \boldsymbol{r}_{0i}) \cdot \nabla \phi(\boldsymbol{r}_0) \tag{3.15}$$

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

$$\boldsymbol{p}(\boldsymbol{r}) = [x\, y\, z]^T \tag{3.16}$$

To estimate the gradient of $\phi(\boldsymbol{r}_0)$, the least squares problem is established using its nearest points (See Fig.2.1):

$$J = \sum_i^N \omega_{0i} \left[\boldsymbol{p}^T (\Delta \boldsymbol{r}_{0i}) \cdot \nabla \hat{\phi}(\boldsymbol{r}_0) - \Delta \phi_{0i} \right]^2 \tag{3.17}$$

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.2.1.

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

$$\frac{\partial J}{\partial \nabla\hat{\phi}(\boldsymbol{r}_0)} = 0 \tag{3.18}$$

Eq.(3.18) can be written as

$$\sum_i^N \omega_{0i}\boldsymbol{p}(\Delta\boldsymbol{r}_{0i}) \cdot \boldsymbol{p}^T(\Delta\boldsymbol{r}_{0i})\nabla\hat{\phi}(\boldsymbol{r}_0) = \sum_i^N \omega_{0i}\boldsymbol{p}(\Delta\boldsymbol{r}_{0i})\Delta\phi_{0i} \tag{3.19}$$

Eq.(3.19) may be simplified as

$$\boldsymbol{S}\nabla\hat{\phi}(\boldsymbol{r}_0) = \boldsymbol{T}\Delta\phi_{0i} \tag{3.20}$$

where

$$\boldsymbol{S} = \begin{bmatrix} \sum_i^N \omega_{0i}\Delta x_{0i}^2 & \sum_i^N \omega_{0i}\Delta x_{0i}\Delta y_{0i} & \sum_i^N \omega_{0i}\Delta x_{0i}\Delta z_{0i} \\ \sum_i^N \omega_{0i}\Delta y_{0i}\Delta x_{0i} & \sum_i^N \omega_{0i}\Delta y_{0i}^2 & \sum_i^N \omega_{0i}\Delta y_{0i}\Delta z_{0i} \\ \sum_i^N \omega_{0i}\Delta z_{0i}\Delta x_{0i} & \sum_i^N \omega_{0i}\Delta z_{0i}\Delta y_{0i} & \sum_i^N \omega_{0i}\Delta z_{0i}^2 \end{bmatrix} \tag{3.21}$$

$$\boldsymbol{T} = \begin{bmatrix} \omega_{01}\Delta\boldsymbol{r}_{01} & \omega_{02}\Delta\boldsymbol{r}_{02} & \cdots & \omega_{0N}\Delta\boldsymbol{r}_{0N} \end{bmatrix} \tag{3.22}$$

By Eq.(3.21) and (3.22), the estimation of the gradient can be expressed as:

$$\nabla\hat{\phi}(\boldsymbol{r}_0) = \boldsymbol{S}^{-1}\boldsymbol{T}\Delta\phi_{0i} \tag{3.23}$$

where

$$\boldsymbol{S}^{-1}\boldsymbol{T} = \begin{bmatrix} a_{01} & a_{02} & \cdots & a_{0N} \\ b_{01} & b_{02} & \cdots & b_{0N} \\ c_{01} & c_{02} & \cdots & c_{0N} \end{bmatrix} \tag{3.24}$$

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

$$\frac{\partial \phi(\boldsymbol{r}_0)}{\partial x} \approx \sum_{i}^{N} a_{0i} \Delta \phi_{0i} \tag{3.25}$$

$$\frac{\partial \phi(\boldsymbol{r}_0)}{\partial y} \approx \sum_{i}^{N} b_{0i} \Delta \phi_{0i} \tag{3.26}$$

$$\frac{\partial \phi(\boldsymbol{r}_0)}{\partial z} \approx \sum_{i}^{N} c_{0i} \Delta \phi_{0i} \tag{3.27}$$

3.2.3 Geometric Conservative Least Squares Method

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

$$\sum_{i=1}^{N} a_{0i} = 0, \sum_{i=1}^{N} b_{0i} = 0, \sum_{i=1}^{N} c_{0i} = 0 \tag{3.28}$$

$$\begin{aligned} \sum_{i=1}^{N} a_{0i} \Delta x_{0i} = 1, \sum_{i=1}^{N} b_{0i} \Delta x_{0i} = 0, \sum_{i=1}^{N} c_{0i} \Delta x_{0i} = 0 \\ \sum_{i=1}^{N} a_{0i} \Delta y_{0i} = 0, \sum_{i=1}^{N} b_{0i} \Delta y_{0i} = 1, \sum_{i=1}^{N} c_{0i} \Delta y_{0i} = 0 \\ \sum_{i=1}^{N} a_{0i} \Delta z_{0i} = 0, \sum_{i=1}^{N} b_{0i} \Delta z_{0i} = 0, \sum_{i=1}^{N} c_{0i} \Delta z_{0i} = 1 \end{aligned} \tag{3.29}$$

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

$$\Lambda = J + \sum_{p=1}^{3} \mu_p M_p + \sum_{p=1}^{3} \sum_{q=1}^{3} \nu_{p,q} N_{p,q} \tag{3.30}$$

where J is the object function denoted in Eq. (3.17), whereas M and N take the form

$$
\begin{aligned}
M_1 = \sum_{i=1}^{N} a_{0i} = 0,\ M_2 = \sum_{i=1}^{N} b_{0i} = 0,\ M_3 = \sum_{i=1}^{N} c_{0i} = 0 \\
N_{1,1} = \sum_{i=1}^{N} a_{0i}\Delta x_{0i} = 1,\ N_{1,2} = \sum_{i=1}^{N} b_{0i}\Delta x_{0i} = 0,\ N_{1,3} \sum_{i=1}^{N} c_{0i}\Delta x_{0i} = 0 \\
N_{2,1} = \sum_{i=1}^{N} a_{0i}\Delta y_{0i} = 0,\ N_{2,2} = \sum_{i=1}^{N} b_{0i}\Delta y_{0i} = 1,\ N_{2,3} \sum_{i=1}^{N} c_{0i}\Delta y_{0i} = 0 \\
N_{3,1} = \sum_{i=1}^{N} a_{0i}\Delta z_{0i} = 0,\ N_{3,2} = \sum_{i=1}^{N} b_{0i}\Delta z_{0i} = 0,\ N_{3,3} \sum_{i=1}^{N} c_{0i}\Delta z_{0i} = 1
\end{aligned}
\tag{3.31}
$$

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 $\nabla\hat{\phi}_0$, μ_p and $\nu_{p,q}$. $\nabla\Lambda = 0$ can be written in matrix form, as follows:

$$
A\boldsymbol{x} = \boldsymbol{b} \tag{3.32}
$$

where

$$
\boldsymbol{A} = \begin{bmatrix} \boldsymbol{D} & \boldsymbol{E} \\ \boldsymbol{E}^T & \boldsymbol{0} \end{bmatrix}, \boldsymbol{D} = \begin{bmatrix} \boldsymbol{S} & 0 & \dots & 0 \\ 0 & \boldsymbol{S} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \boldsymbol{S} \end{bmatrix}, \boldsymbol{E} = \begin{bmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \\ \vdots \\ \boldsymbol{e}_N \end{bmatrix} \tag{3.33}
$$

$$
\boldsymbol{e}_i = \begin{bmatrix} \boldsymbol{I} & \Delta x_{0i}\boldsymbol{I} & \Delta y_{0i}\boldsymbol{I} & \Delta z_{0i}\boldsymbol{I} \end{bmatrix} \tag{3.34}
$$

$$\boldsymbol{x} = \begin{bmatrix} a_{01} \\ b_{01} \\ c_{01} \\ \vdots \\ a_{0N} \\ b_{0N} \\ c_{0N} \\ \mu_1 \\ \mu_2 \\ \mu_3 \\ \nu_{1,1} \\ \nu_{1,2} \\ \vdots \\ \nu_{3,3} \end{bmatrix}, \boldsymbol{b} = \begin{bmatrix} \omega_{01}\Delta\boldsymbol{r}_{01}^T \\ \vdots \\ \omega_{0N}\Delta\boldsymbol{r}_{0N}^T \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{3.35}$$

where $\boldsymbol{I}$ is a 3×3 identity matrix and $\boldsymbol{S}$ is the matrix in Eq. (3.21). As a consequence, $\boldsymbol{D}$ is a $3N \times 3N$ matrix, because S is a 3×3 matrix. The meshless coefficients may be obtained by multiplying the inverse matrix of $\boldsymbol{A}$ by $\boldsymbol{b}$.

The meshless coefficients derived from Eq. (3.32) satisfy geometric conservation laws because the geometric conservative conditions in Eq.(3.28) are constrained when solving the least squares problem. It is known that geometric conservative meshless coefficients are robust, even in randomly distributed point clouds [35].

3.3 Flux schemes

3.3.1 AUSMPW+

AUSMPW+ was developed by Kim[48] to increase accuracy in capturing shocks without sacrificing robustness. The numerical flux obtained by AUSMPW+ is given by as

$$\boldsymbol{F}_{\frac{1}{2}} = \bar{M}_L^+ c_{\frac{1}{2}} \boldsymbol{\Phi_L} + \bar{M}_R^- c_{\frac{1}{2}} \boldsymbol{\Phi_R} + (P_L^+]\boldsymbol{P}_L + P_R^-]\boldsymbol{P}_R) \tag{3.36}$$

$$\boldsymbol{\Phi} = \begin{bmatrix} \rho & \rho u & \rho v & \rho w & \rho H \end{bmatrix}, \boldsymbol{P} = \begin{bmatrix} 0 & pn_x & pn_y & pn_z & 0 \end{bmatrix} \tag{3.37}$$

where n_x, n_y, n_z are the normalized cell area vectors. Then, $\bar{M}_{L,R}^{\pm}$ for $m_{\frac{1}{2}} = M_L^+ + M_R^-$ is given as

$$\bar{M}_L^+ = \begin{cases} M_L^+ + M_R^- \left[(1-\omega_{pw})(1+f_R) - f_L\right] & , m_{\frac{1}{2}} \geq 0 \\ \omega_{pw} M_L^+ (1+f_L) & , m_{\frac{1}{2}} < 0 \end{cases} \tag{3.38}$$

$$\bar{M}_R^- = \begin{cases} \omega_{pw} M_R^- (1+f_R) & , m_{\frac{1}{2}} \geq 0 \\ M_R^- + M_L^+ \left[(1-\omega_{pw})(1+f_L) + f_L - f_R\right] & , m_{\frac{1}{2}} < 0 \end{cases} \tag{3.39}$$

with pressure weighting

$$\omega_{pw} = 1 - \min\left(\frac{p_L}{p_R}, \frac{p_R}{p_L}\right)^3 \tag{3.40}$$

Then, $f_{L,R}$ is given by

$$f_{L,R} = \left(\frac{p_{L,R}}{p_s} - 1\right) \tag{3.41}$$

where

$$p_s = p_L^+ p_L + p_R^- p_R \tag{3.42}$$

The split Mach number is defined by

$$M^{\pm} = \begin{cases} \pm\frac{1}{4}(M \pm 1)^2 & , |M| \leq 1 \\ \pm\frac{1}{2}(M \pm |M|)^2 & , |M| > 1 \end{cases} \tag{3.43}$$

$$p^{\pm} = \begin{cases} \pm\frac{1}{4}(M \pm 1)^2(2 \mp M) & , |M| \leq 1 \\ \pm\frac{1}{2}(1 \mp sign(M)) & , |M| > 1 \end{cases} \tag{3.44}$$

The Mach number of each side is

$$M_{L,R} = \frac{U_{L,R}}{c_{s,1/2}} \tag{3.45}$$

U is the velocity component that is parallel to meshless coefficient or cell interface area vector for FVM with its corresponding neighbor. And the speed of sound $c_{s,1/2}$ is given by

$$c_{s,1/2} = \begin{cases} \frac{c_s^{*2}}{\max(|U_L|, c_s^*)} & , \frac{1}{2}(U_L + U_R) > 0 \\ \frac{c_s^{*2}}{\max((|U_R|, c_s^*)} & , \frac{1}{2}(U_L + U_R) < 0 \end{cases} \tag{3.46}$$

where

$$c_s^* = \sqrt{2(\gamma - 1)/(\gamma + 1)H_{\text{normal}}} \tag{3.47}$$

$$H_{\text{normal}} = \frac{1}{2}\left(H_L - \frac{1}{2}V_L^2 + H_R - \frac{1}{2}V_R^2\right) \tag{3.48}$$

3.3.2 AUSM$^+$-up

AUSM$^+$-up is also used as a flux scheme for unstructured cell finite volume method as the flux scheme shows excellent convergence behavior for all speed flows. The numerical flux of AUSM-family is given by

$$\boldsymbol{f}_{1/2} = \dot{m}_{1/2}\boldsymbol{\psi}_{L/R} + \boldsymbol{p}_{1/2} \tag{3.49}$$

where, $\dot{m}_{1/2} = \rho u$ and $\boldsymbol{\psi} = (1, u, H)^T$

the mass flux given by AUSM$^+$-up is expressed as

$$\dot{m}_{1/2} = \begin{cases} a_{1/2}M_{1/2}\rho_L & , M_{1/2} > 0 \\ a_{1/2}M_{1/2}\rho_R & , M_{1/2} \leq 0 \end{cases} \tag{3.50}$$

The mach number functions are given as follows

$$M_{1/2} = M^+_{(4)}(M_L) + M^-_{(4)}(M_R) - \frac{0.25}{f_a}\text{max}(1 - \bar{M}^2, 0)\frac{p_R - p_L}{\rho_{1/2}a^2_{1/2}} \tag{3.51}$$

$$\rho_{1/2} = (\rho_L + \rho_R)/2 \tag{3.52}$$

$$\bar{M}^2 = \frac{u_L^2 + u_R^2}{2a^2_{1/2}} \tag{3.53}$$

$$M_0^2 = \text{min}(1, \text{max}(\bar{M}^2, M^2_\infty)) \in [0, 1] \tag{3.54}$$

$$f_a(M_0) = M_0(1 - M0) \in [0, 1] \tag{3.55}$$

$$M^\pm_{(1)}(M) = \frac{1}{2})(M \pm |M|), M^\pm_{(2)}(M) = \pm\frac{1}{4}(M \pm 1)^2 \tag{3.56}$$

$$M^{\pm}_{(4)}(M) = \begin{cases} M^{\pm}_{(1)} & , |M| \geq 1 \\ M^{\pm}_{(2)}(1 \mp 16\beta M^{\mp}_{(2)} & , |M| < 1 \end{cases} \tag{3.57}$$

$a_{1/2}$ in Eq.(3.53) is given as

$$a_{1/2} = \min(\hat{a}_L, \hat{a}_B) \tag{3.58}$$

$$\hat{a} = \frac{a^{*2}}{\max(a^*, |u|)}, a^{*2} = \frac{2(\gamma - 1)}{\gamma + 1} H \tag{3.59}$$

An the pressure flux is as follows

$$p_{1/2} = P^{+}_{(5)}(M_L)p_L + P^{-}_{(5)}(M_R)p_R - 0.75P^{+}_{(5)}P^{-}_{(5)}(\rho_L + \rho_R)(f_a a_{1/2})(u_R - u_L) \tag{3.60}$$

where

$$P^{\pm}_{(5)}(M) = \begin{cases} \frac{1}{M} M^{\pm}_{(1)} & , |M| \geq 1 \\ M^{\pm}_{(2)} \left[(\pm 2 - M) \mp 16\alpha M^{\mp}_{(2)} \right] & , |M| < 1 \end{cases} \tag{3.61}$$

with

$$\alpha = \frac{3}{16}(-4 + 5f_a^2) \in \left[-\frac{3}{4}, \frac{3}{16} \right] \tag{3.62}$$

$$\beta = \frac{1}{8} \tag{3.63}$$

3.4 Limiter

3.4.1 TVD schemes

In this study, total variational diminishing (TVD) is applied for accurate solutions. The face value $\phi_{i+1/2}$ is adjusted using the flux limiter function $\psi(r)$. And the flux

limiter function is a non-linear function of r. r for a structured grid is given as

$$r_{1+1/2} = \frac{\phi_i - \phi_{i-1}}{\phi_{i+1} - \phi_i} \tag{3.64}$$

The face value is then adjusted as

$$\phi_{i+1/2} = \phi_i + \frac{1}{2}\psi(r_{i+1/2})(\phi_{i+1} - \phi_i) \tag{3.65}$$

Among many TVD schemes, MINMOD limiter is used in this study owing to its robustness with acceptable accuracy[49]. MINMOD limiter is expressed as

$$\psi(r) = \max\left(0, \min(1, r)\right) \tag{3.66}$$

For unstructured grids, the left and right values need to be modified since $i - 1$ and $i + 1$ may not exist for unstructured grids. The modified notation for unstructured grids of Eq. (3.65) is given as

$$\phi_f = \phi_C + \frac{1}{2}\psi(r_f)(\phi_D - \phi_C) \tag{3.67}$$

where

$$r_f = \frac{\phi_C - \phi_U}{\phi_D - \phi_C} \tag{3.68}$$

f, C, D and U are illustrated in Figure 3.2.

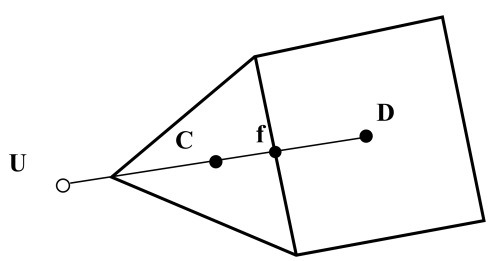


Figure 3.2: Node notation for unstructured TVD schemes

a virtual node U is defined by exact r formulation [50], which given as

$$r_f = \frac{(\phi_D - \phi_U) - (\phi_D - \phi_C)}{\phi_D - \phi_C} \tag{3.69}$$

In Eq. (3.69), ϕ_C and ϕ_D are given values, while ϕ_U should be estimated, as U represents a virtual node, as shown in Figure 3.2. Then $\phi_D - \phi_U$ may be estimated as

$$(\phi_D - \phi_U) = \nabla\phi_C \cdot \boldsymbol{r}_{UD} = 2\nabla\phi_C \cdot \boldsymbol{r}_{CD} \tag{3.70}$$

Then, Eq. (3.69) may be recast as

$$r_f = \frac{2\nabla\phi_C \cdot \boldsymbol{r}_{CD}}{\phi_D - \phi_C} - 1 \tag{3.71}$$

r can be used for both unstructured and meshless discretization.

3.4.2 Venkatakrishnan limiter

The face flux by the limiter function of Venkatakrishnan is given as

$$\phi_f = \phi_C + \psi_{\text{ven}} \nabla\phi_C \cdot \Delta\boldsymbol{r}_{fC} \tag{3.72}$$

$$\psi_{\text{ven}} = \min\left(\psi_{Ci}\right) \tag{3.73}$$

where i denotes the nodes that compose cell C as shown in Figure 3.3.

ψ_{Ci} is given by

$$\psi_{Ci} = \frac{1}{\Delta_-}\left[\frac{(\Delta_+^2 + \epsilon^2)\Delta_- + 2\Delta_-^2\Delta_+}{\Delta_+^2 + 2\Delta_-^2 + \Delta_-\Delta_+ + \epsilon^2}\right] \tag{3.74}$$

$$\Delta_+ = \max(\phi_D) - \phi_C \tag{3.75}$$

$$\Delta_{-} = \nabla\phi_C \cdot \Delta \boldsymbol{r_{Ci}} \tag{3.76}$$

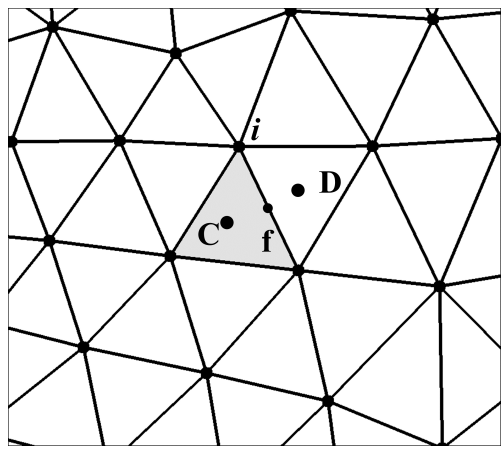


Figure 3.3: Node notation for Venkatakrishnan limiter

In Eq.(3.75), D is a adjacent cell index.

3.4.3 Gradient calculation

In this study, two methods are used to estimate the gradient: the first is the least squares method, which is described in Section 3.2.2, and the second is the node-based Green-Gauss theorem. Thus, only node-based Green-Gauss Theorem is described in this Section.

The gradient of function ϕ using Green-Gauss Theorem is given by

$$\nabla\phi_C \approx \frac{1}{V_C}\sum_f \phi_f \boldsymbol{A}_f \tag{3.77}$$

V_C denotes the volume of cell C, and f represents the interface index between C and its neighboring cells. And A_f denotes a interface area vector. For node-based Green-Gauss theorem, ϕ_f is the average of the nodal values that compose interface

f. The nodal values are estimated by cells which share node with face f as shown in Figure 3.4.

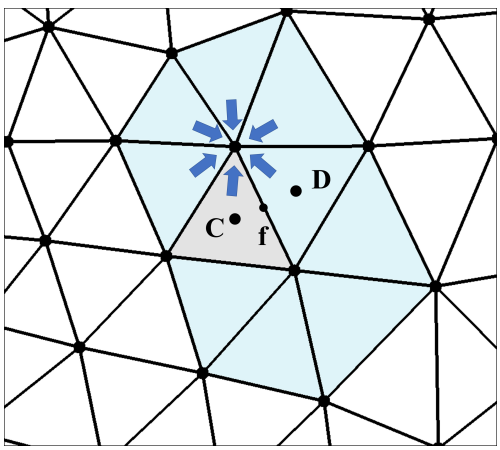


Figure 3.4: Illustration of nodal value interpolation

3.5 Time integration

3.5.1 Local time stepping

To accelerate convergence for steady state problems, local time stepping may be employed in this study. A local time step is estimated based on the spectral radius of the convective and viscous flux Jacobian. the spectral radius of convective and viscous flux an arbitrary cell or node $'0'$ may be expressed as

$$\Lambda_c = \sum_{i=1}^{n} \left(|\boldsymbol{V}_0 \cdot \boldsymbol{A}_{0i}| + \parallel \boldsymbol{A}_{0i} \parallel \right) c_0 \tag{3.78}$$

where $\boldsymbol{A}_{0i}$ denotes $\boldsymbol{S_{0i}}/V_0$ in Eq. (3.13) and meshless coefficient vector (a_{0i}, b_{0i}, c_{0i}) for CFV and meshless discretization respectively. And $\boldsymbol{V}_0$ and c_0 denote the velocity vector and speed of sound at point 0.

$$\Lambda_v = \frac{\gamma^{2/3} M_\infty}{Re_\infty Pr} \sum_{i=1}^{n} \frac{\mu_{0i+1/2}}{\rho_{0i+1/2}} \parallel \boldsymbol{A}_{0i} \parallel^2 \tag{3.79}$$

where $\mu_{0i+1/2}$ and $\rho_{0i+1/2}$ is average viscosity and density between 0 and i. Then, local time step is given as

$$\Delta t_0 = \frac{CFL}{\Lambda_c + \Lambda_v} \tag{3.80}$$

3.5.2 Explicit Runge-Kutta

Eq. (3.1) can be explicitly discretized as

$$\frac{\Delta \boldsymbol{q}_0^n}{\Delta t_0} = \boldsymbol{R}(\boldsymbol{q}_0^n) \tag{3.81}$$

The four-stage explicit Runge-Kutta time integration is employed in this study[51].

$$\begin{aligned}
\boldsymbol{q}_0^0 &= \boldsymbol{q}_0^n \\
\boldsymbol{q}_0^{(1)} &= \boldsymbol{q}_0^0 + \alpha_1 \Delta t \boldsymbol{R}(\boldsymbol{q}_0^0) \\
\boldsymbol{q}_0^{(2)} &= \boldsymbol{q}_0^1 + \alpha_2 \Delta t \boldsymbol{R}(\boldsymbol{q}_0^1) \\
\boldsymbol{q}_0^{(3)} &= \boldsymbol{q}_0^2 + \alpha_3 \Delta t \boldsymbol{R}(\boldsymbol{q}_0^2) \\
\boldsymbol{q}_0^{(4)} &= \boldsymbol{q}_0^3 + \alpha_4 \Delta t \boldsymbol{R}(\boldsymbol{q}_0^3) \\
\boldsymbol{q}_0^{n+1} &= \boldsymbol{q}_0^{(4)}
\end{aligned} \tag{3.82}$$

3.5.3 LU-SGS for unstructured grids

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. Eq. (3.1) can be discretized implicitly on an arbitrary cell (or node) 0 as follows:

$$\frac{\Delta \boldsymbol{q}_0^n}{\Delta t_0} + \sum_i^N \boldsymbol{H}_{0i}^{n+1} = 0 \tag{3.83}$$

where, N is the number of nearest neighboring cells or nodes. The flux function $\boldsymbol{H}_{0i}$ between 0 and i can be expressed in various manners. The flux function using Meshless discretization may be expressed as

where,

$$\begin{aligned} \boldsymbol{H}_{0i} = a_{0i}(\boldsymbol{f}_i - \boldsymbol{f}_0) + b_{0i}(\boldsymbol{g}_i - \boldsymbol{g}_0) + c_{0i}(\boldsymbol{h}_i - \boldsymbol{h}_0) \\ - \left[a_{0i}(\boldsymbol{f}_{v,i} - \boldsymbol{f}_{v,0}) + b_{0i}(\boldsymbol{g}_{v,i} - \boldsymbol{g}_{v,0}) + c_{0i}(\boldsymbol{h}_{v,i} - \boldsymbol{h}_{v,0})\right] \end{aligned} \tag{3.84}$$

In this study, the flux at the midpoint is used, resulting in the following expression for Eq.(3.84):

$$\begin{aligned} \boldsymbol{H}_{0i} = 2\left[a_{0i}(\boldsymbol{f}_{i+1/2} - \boldsymbol{f}_0) + b_{0i}(\boldsymbol{g}_{i+1/2} - \boldsymbol{g}_0) + c_{0i}(\boldsymbol{h}_{i+1/2} - \boldsymbol{h}_0)\right] \\ -2\left[a_{0i}(\boldsymbol{f}_{v,i+1/2} - \boldsymbol{f}_{v,0}) + b_{0i}(\boldsymbol{g}_{v,i+1/2} - \boldsymbol{g}_{v,0}) + c_{0i}(\boldsymbol{h}_{v,i+1/2} - \boldsymbol{h}_{v,0})\right] \end{aligned} \tag{3.85}$$

For GC-LSM, Eq.(3.83) can be recast as

$$\begin{aligned} \frac{\Delta \boldsymbol{q}_0^n}{\Delta t_0} + \sum_i^N 2\Bigg[\Big(a_{0i}\boldsymbol{f}_{i+1/2} + b_{0i}\boldsymbol{g}_{i+1/2} + c_{0i}\boldsymbol{h}_{i+1/2}\Big) - \\ \Big(a_{0i}\boldsymbol{f}_{v,i+1/2} + b_{0i}\boldsymbol{g}_{v,i+1/2} + c_{0i}\boldsymbol{h}_{v,i+1/2}\Big)\Bigg] = 0 \end{aligned} \tag{3.86}$$

Since, $\sum_i^N a_{0i} = 0, \sum_i^N b_{0i} = 0, \sum_i^N c_{0i} = 0$. Then, $\boldsymbol{H}_{0i}$ by GC-LSM may be expressed as

$$\begin{aligned} \boldsymbol{H}_{0i} = 2\Bigg[\Big(a_{0i}\boldsymbol{f}_{i+1/2} + b_{0i}\boldsymbol{g}_{i+1/2} + c_{0i}\boldsymbol{h}_{i+1/2}\Big) - \\ \Big(a_{0i}\boldsymbol{f}_{v,i+1/2} + b_{0i}\boldsymbol{g}_{v,i+1/2} + c_{0i}\boldsymbol{h}_{v,i+1/2}\Big)\Bigg] \end{aligned} \tag{3.87}$$

The flux function $\boldsymbol{H}_{0i}$ of finite volume methods is given as

$$\boldsymbol{H}_{0i} = \sum_{i}^{N} \left((\boldsymbol{G}_{0i} - \boldsymbol{G}_{v,0i}) \cdot \frac{\boldsymbol{S}_{0i}}{V_0} \right) \tag{3.88}$$

The convective and viscous fluxes are represented by $\boldsymbol{G}$ and $\boldsymbol{G}_v$, respectively, in the expression for the flux function $\boldsymbol{H}_{0i}$. $\boldsymbol{S}_{0i}$ and V_0 are the interface surface vector and volume of the cell 0 respectively. As seen in Eq. (3.87) and (3.88), $\frac{1}{2}(a_{0i}, b_{0i}, c_{0i})$ and $\boldsymbol{S}0i/V0$ correspond to each other. Thus, Eq.(3.84) is universal for meshless and finite volume methods.

As a result, $\boldsymbol{H}_{0i}$ may be linearized independent of methods. Then, $\boldsymbol{H}_{0i}$ is expressed as

$$\boldsymbol{H}_{0i}^{n+1} = \boldsymbol{H}_{0i}^{n} + \boldsymbol{A}_{0i}^{+} \Delta \boldsymbol{q}_0 + \boldsymbol{A}_{0i}^{-} \Delta \boldsymbol{q}_i \tag{3.89}$$

where

$$\boldsymbol{A}_{0i}^{\pm} = \frac{1}{2} \left(\boldsymbol{A}_{0i} \pm \lambda_{0i} I \right) \tag{3.90}$$

$$\boldsymbol{A} = \partial \boldsymbol{H} / \partial \boldsymbol{q} \tag{3.91}$$

where λ_{0i} is the eigenvalue of the Jacobian matrix $\boldsymbol{A}_{0i}$. By Eq.(3.89), (3.83) may be written as

$$\left(\frac{1}{\Delta \tau_0} + \frac{1}{2} \sum_{i}^{N} \lambda_{0i} \right) \boldsymbol{I} \Delta \boldsymbol{q}_0 + \sum_{i}^{N} \boldsymbol{A}_{0i}^{-} \Delta \boldsymbol{q}_i - \sum_{i}^{N} \boldsymbol{A}_{0i} \Delta \boldsymbol{q}_0 = - \sum_{i}^{N} \boldsymbol{H}_{0i}^{n} \tag{3.92}$$

For finite volume methods, the sum of Jacobian matrix $\boldsymbol{A}_{0i}$ in Eq. (3.92) must be zero due to the geometric conservation law. Generally, conventional meshless discretization cannot satisfy $\sum_i^N \boldsymbol{A}_{0i} = 0$ since the methods are not geometric conservative discretization, whereas GC-LSM naturally satisfy $\sum_i^N \boldsymbol{A}_{0i} = 0$ due to its geometric conservative features. Due to this, the original multicloud method used to implemented to only for explicit schemes since the meshless discretization used does not satisfy geometric conservation law. Consequently, Eq.(3.92) may be recast as

$$\left(\frac{1}{\Delta\tau_0} + \frac{1}{2}\sum_i^N \lambda_{0i}\right)\boldsymbol{I}\Delta\boldsymbol{q}_0 + \sum_{i\in L(0)}^{N} \boldsymbol{A}_{0i}^{-}\Delta\boldsymbol{q}_i + \sum_{i\in U(0)}^{N} \boldsymbol{A}_{0i}^{-}\Delta\boldsymbol{q}_i = \boldsymbol{R}_0^n \tag{3.93}$$

where $L(0)$ and $U(0)$ are the set of nearest points whose indices are less than and greater than those for point 0, respectively. Consequently, Eq.(3.93) 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:

$$\Delta\boldsymbol{q}_0^* = \boldsymbol{D}_0^{-}\left[\boldsymbol{R}_0^n - \sum_{i\in L(0)}^{n} \boldsymbol{A}_{0i}^{-}\Delta\boldsymbol{q}_i^*\right] \tag{3.94}$$

$$\Delta\boldsymbol{q}_0 = \Delta\boldsymbol{q}_0^* - \boldsymbol{D}_0^{-}\sum_{i\in U(0)}^{N} \boldsymbol{A}_{0i}^{-}\Delta\boldsymbol{q}_i \tag{3.95}$$

where

$$\boldsymbol{D}_0 = \left(\frac{1}{\Delta\tau_0} + \frac{1}{2}\sum_i^N \lambda_{0i}\right)\boldsymbol{I} \tag{3.96}$$

Eq.(3.95) and (3.94) are forward and backward sweeps, respectively. Thus, LU-SGS may be implemented to the multicloud method without formulating new algorithm complexity by using only GC-LSM.

3.5.4 LU-SGS for non-conservative meshless discretization

In this thesis, LU-SGS for non-conservative meshless discretization is also formulated to compare the performances between GC-LSM and LSM discretization. Eq.(3.92) without the geometric conservation law can be expressed as:

$$\left(\frac{1}{\Delta\tau_0}+\frac{1}{2}\sum_{i}^{N}\lambda_{0i}\right)\boldsymbol{I}\Delta\boldsymbol{q}_0+\sum_{i\in L(0)}^{N}\boldsymbol{A}_{0i}^{-}\Delta\boldsymbol{q}_i+\sum_{i\in U(0)}^{N}\boldsymbol{A}_{0i}^{-}\Delta\boldsymbol{q}_i-\sum_{i}^{N}\boldsymbol{A}_{0i}\Delta\boldsymbol{q}_0=\boldsymbol{R}_0^n \tag{3.97}$$

Then Eq.(3.97) may be recast as:

$$\boldsymbol{D}_0\Delta\boldsymbol{q}_0+\boldsymbol{L}\Delta\boldsymbol{q}_i+\boldsymbol{U}\Delta\boldsymbol{q}_i+\tilde{\boldsymbol{D}}_0\Delta\boldsymbol{q}_0=\boldsymbol{R}_0^n \tag{3.98}$$

where,

$$\boldsymbol{L}=\sum_{i\in L(0)}\boldsymbol{A}_{0i}^{-},\ \ \boldsymbol{U}=\sum_{i\in U(0)}\boldsymbol{A}_{0i}^{-} \tag{3.99}$$

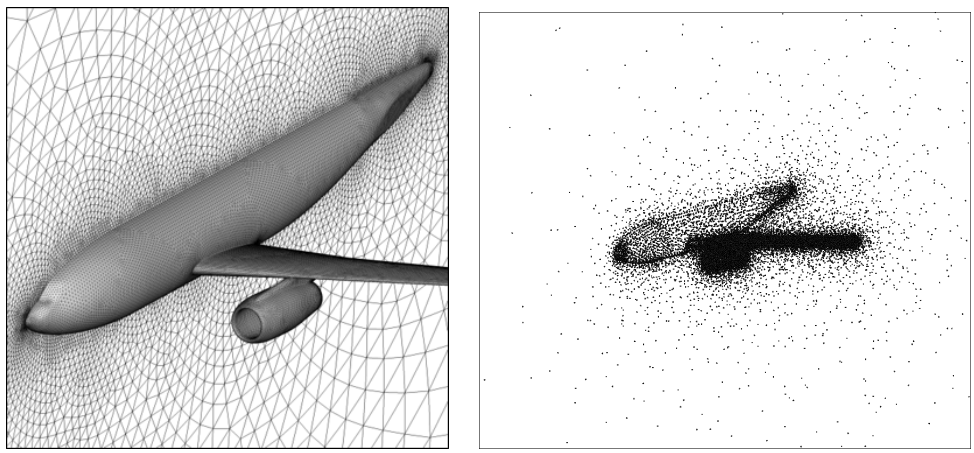
$$\boldsymbol{D}_0=\left(\frac{1}{\Delta\tau_0}+\frac{1}{2}\sum_{i}^{N}\lambda_{0i}\right)\boldsymbol{I},\ \ \tilde{\boldsymbol{D}}_0=\frac{1}{2}\sum_{i}^{N}\boldsymbol{A}_{0i} \tag{3.100}$$

Eq.(3.98) can be described as:

$$(\boldsymbol{L}+\boldsymbol{D}_0)(\boldsymbol{D}_0+\tilde{\boldsymbol{D}}_0)^{-1}(\boldsymbol{U}+\boldsymbol{D}_0)\Delta\boldsymbol{q}_0=\boldsymbol{R}_0^n \tag{3.101}$$

Finally, Eq.(3.101) can be solved as:

$$\Delta\boldsymbol{q}_0^*=\boldsymbol{D}_0^{-}\left[\boldsymbol{R}_0^n-\sum_{i\in L(0)}^{n}\boldsymbol{A}_{0i}^{-}\Delta\boldsymbol{q}_i^*\right]\ \ \text{: Forward sweep} \tag{3.102}$$



(a) Configuration of f6wbnp and its flow grid (b) f6wbnp's level 3 flow point domain

Figure 3.5: Illustration of flow grid for f6wbnp

$$\boldsymbol{D}_0 \Delta \boldsymbol{q}_0^* = \boldsymbol{D}_0 \Delta \boldsymbol{q}_0^* + \tilde{\boldsymbol{D}}_0 \Delta \boldsymbol{q}_0^* \quad \text{: Diagonal sweep} \tag{3.103}$$

$$\boldsymbol{D}_0 \Delta \boldsymbol{q}_0 = \boldsymbol{D}_0 \Delta \boldsymbol{q}_0^* - \sum_{i \in U(0)}^{N} \boldsymbol{A}_{0i}^- \Delta \boldsymbol{q}_i \quad \text{: Backward sweep} \tag{3.104}$$

Non-conservative meshless LU-SGS (M-LU) was tested for LSM meshless discretization. The flow grids is 3 level CNA coarsened grid of Figure 3.5a. The flow point domain is illustrated in Figure 3.5b. The reason level 3 domain is chosen is that unstructured LU-SGS (U-LU) generally works well when points are well-distributed, such as in an unstructured grid, even for non-conservative meshless discretization. In order to demonstrate M-LU, the points must be randomly distributed, as the coarse-level point distribution is always random in the multicloud method. The flow condition is Mach 0.75 with AOA $0.49°$. The CFL condition for M-LU is 15, while for

U-LU it is 4.2, as a value higher than 4.2 for U-LU leads to divergence right after the few iterations. 4.2 is threshold CFL numebr for U-LU with LSM.

Figure 3.6 illustrates convergence histrories for M-LU and U-LU. U-LU leads to converges the order of -2 even for much less CFL number than M-LU. Whereas, M-LU converges stably to the order of -15 with higher convergence rates than U-LU. It is evident that M-LU works perfectly for non-conserative meshless dicretization with randomly distributed points.

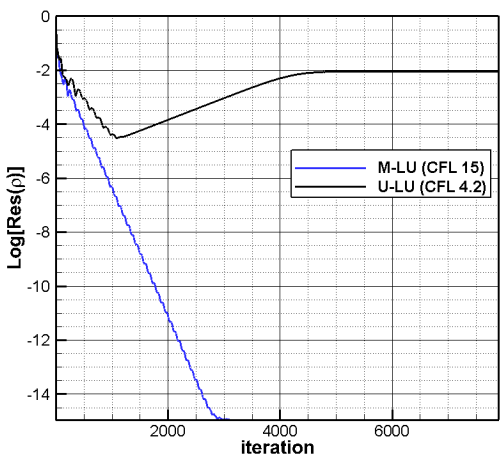


Figure 3.6: Comparison between M-LU and U-LU with LSM discretization

Chapter 4

Numerical Results

4.1 Comparisons for explicit and implicit time integration

Unlike the previous Multicloud methods, An implicit version of multiclut method method using LU-SGS is developed in this study. The purpose of this section is to verify the performance of the implicit multicloud by comparing to the explicit multicloud method. In this study, Runge-Kutta 4th order time integration method is employed as a explicit scheme. Cases involving tetrahedral grids in three-dimensional configurations were tested to analyze the acceleration performance according to time integration methods. The first case is ONERA M6 and The second case is DLR-F6 wing body base configuration with transonic flows. GC-LSM is used to compute both fine and coarse level solutions in this section. AUSM$^+$-up is used for flux estimation with MINMOD limiter for both cases.

4.1.1 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 Figure 4.1b. 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 Figure 4.1c. The coarsening results are shown in Figure 4.1d. 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 to the single RK4 method, such that the work associated with RK4 is unity, as shown in Table 4.1. LU-SGS spends much less computational coat than RK4 since RK4 should calculate 4 residuals for 4 each step. Hence, single-grid LU-SGS spends computation cost 0.39 of RK4. In the multicloud method, the computational cost of four-level V cycle is the factor of 1.5-1.6 of single-method. Hence, the given problem, time per iteration of RK4 four-level V cycle was the factor of 1.59 of single RK4. And LU-SGS with four-level V cycle was the factor of 0.62. Every compuational cost was estimated with repect to single RK4. In fact, the computational cost spent on single RK4 is higher than four-level V cycle LU-SGS.

The convergence history is presented in Figure 4.2a and 4.2b. Figure 4.2a displays the density residual per iteration on the finest grid, and Figure 4.2b displays the density residual per unit work. In Figure 4.2a, the four-level procedure with LU-SGS

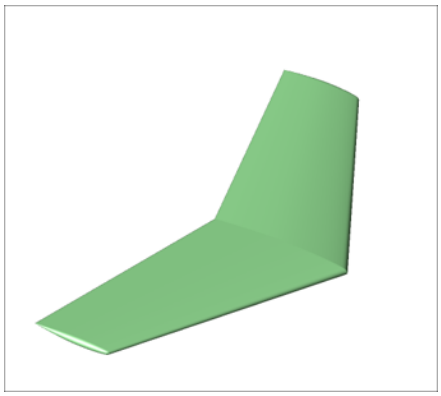
exhibits dramatic convergence speedup with respect to the number of iterations. Four levels with RK4 also exhibited substantial speedup, exceeding that of single LU-SGS. However, the unit work associated with RK4 exceed those associated with LU, such that the convergence history with respect to the number of unit work shows a different result compared to the history with respect to the number of iterations, as shown in Figure 4.2a. 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[52]. The surface pressure results are compared with the experimental results[53] in Figure 4.3 4.8. 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[54]. Finally, the four-level procedure and single LU display satisfactory agreement with each other as well as with the experimental results.

Scheme	RK4 SG	RK4 MC	LU SG	LU MC
Unit work	1.00	1.59	0.39	0.62
CFL	1	1	20	20

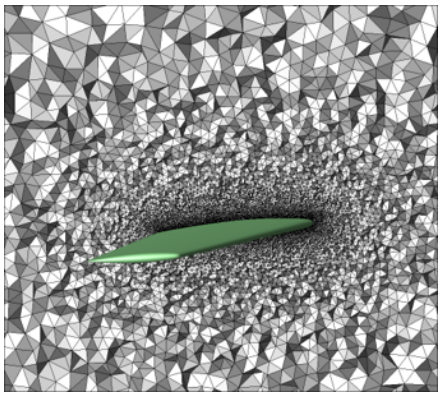
Table 4.1: Units of work and CFL for each case

4.1.2 DLR-F6 wing body configuration

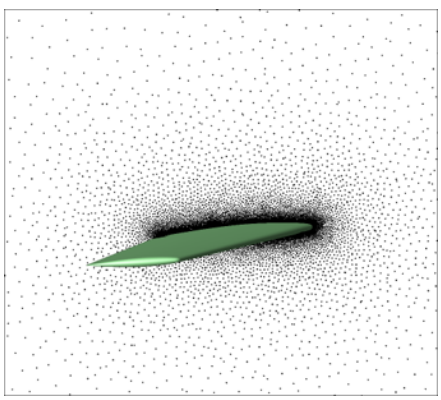
To highlight the performance of the implicit multicloud method, the flow around the baseline wing-body DLR-F6[41], whose geometry is complicated, was analyzed. The Mach number of the flow was 0.75 with an angle of attack of $0.49°$. In Figure 4.10,



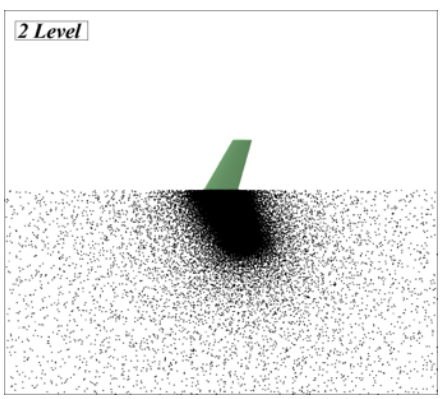
(a) ONERA M6 configuration



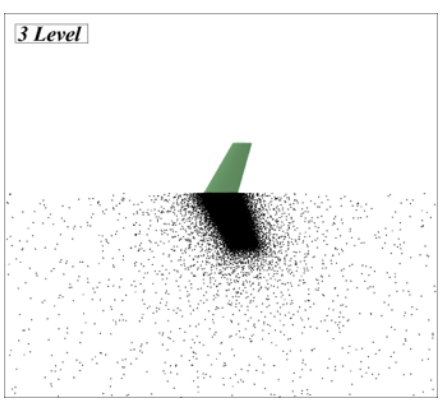
(b) Tetrahedral grids for ONERA M6



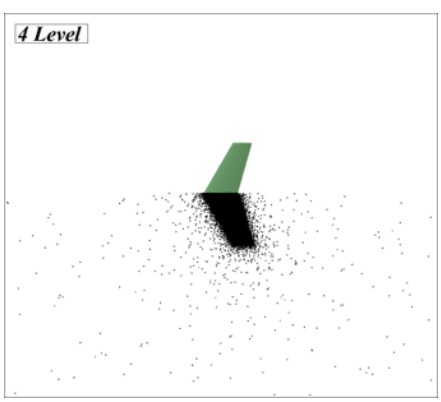
(c) Global point cloud obtained from nodes of tetrahedral grids



(d) Coarse level cloud for level 2

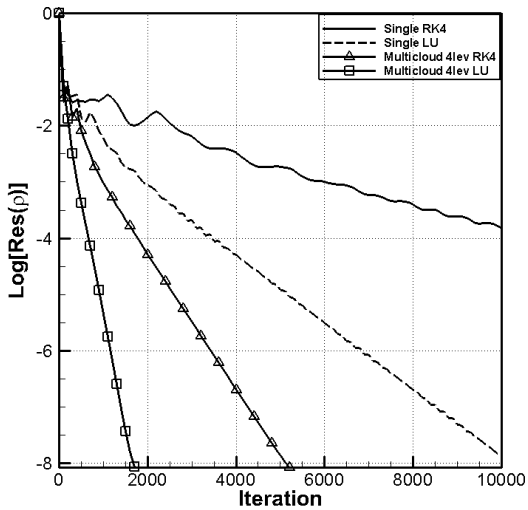


(e) Coarse level cloud for level 3

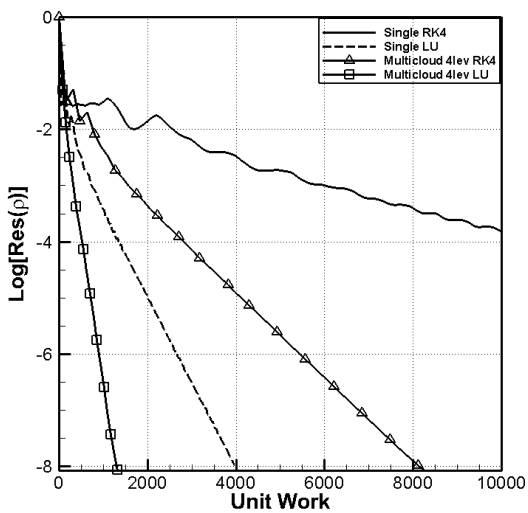


(f) Coarse level cloud for level 4

Figure 4.1: Coarsening results for ONERA M6



(a) Convergence history in term of iteration



(b) Convergence history in term of unit work

Figure 4.2: Convergence histories for ONERA M6

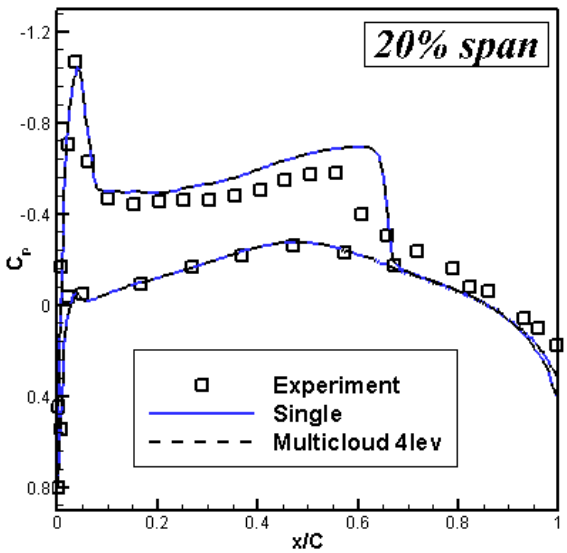


Figure 4.3: Surface pressure along y=0.2η

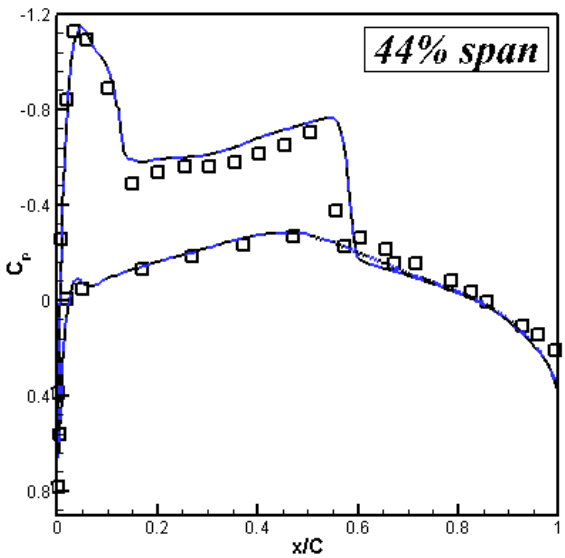


Figure 4.4: Surface pressure along y=0.44η

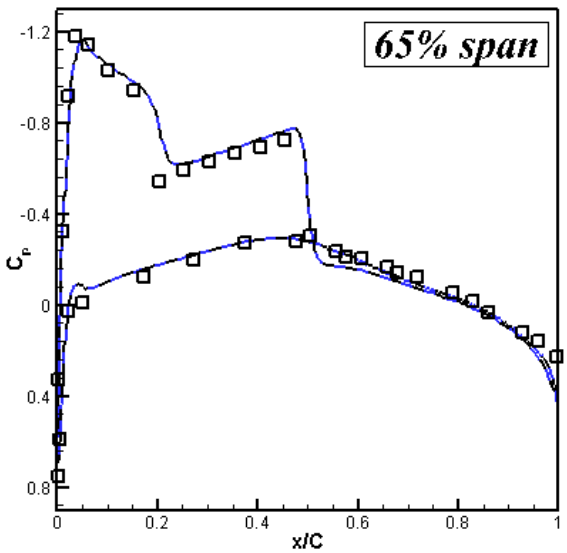


Figure 4.5: Surface pressure along y=0.65η

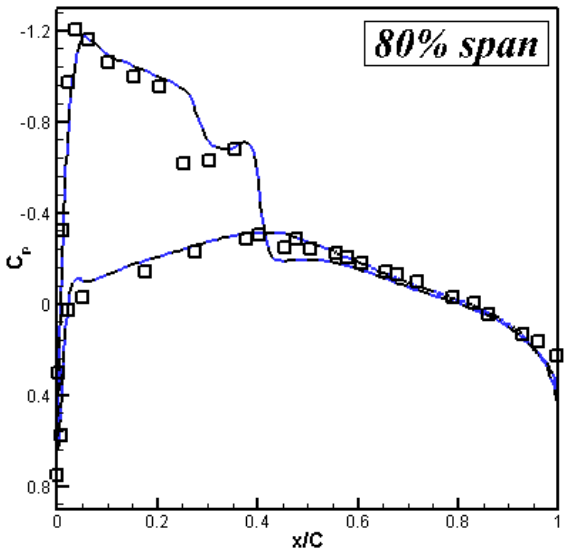


Figure 4.6: Surface pressure along y=0.8η

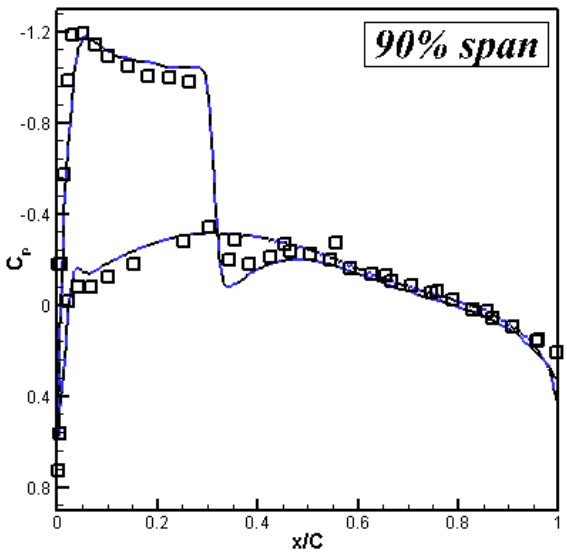


Figure 4.7: Surface pressure along y=0.9η

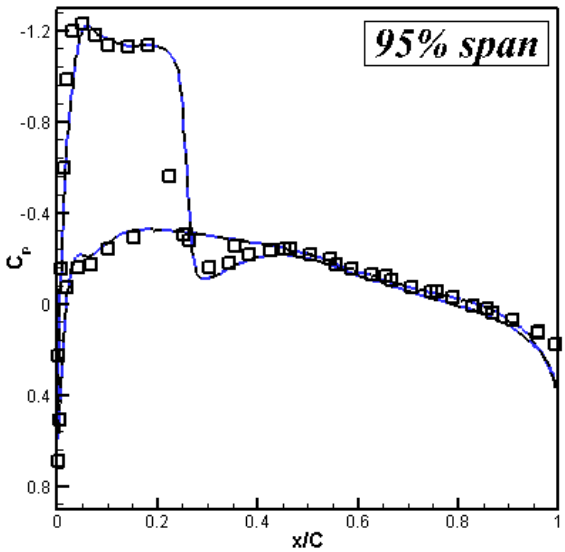


Figure 4.8: Surface pressure along y=0.95η

the finest and coarse level clouds are shown. It appears that the coarsening procedure works well, even for complex geometries.

In Figure 4.11, 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 Figure 4.12 and 4.13, the results of surface pressure are compared for the four-level, single, and experimental[55] cases. As seen in Figure 4.12 and 4.13, 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[54]. 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[41]. 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. As a result, the implicit multicloud method provide significantly dramatic acceleration effect compared to that of explicit method even for complex configurations.

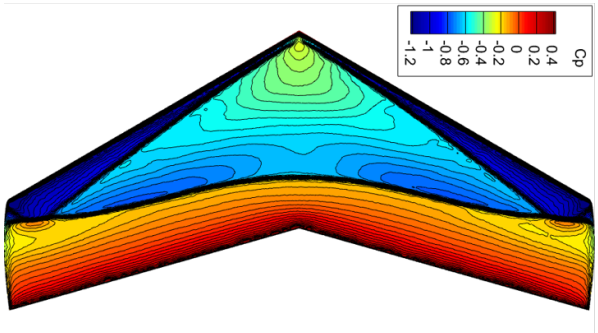


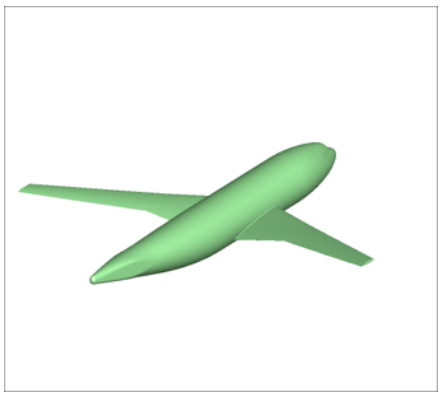
Figure 4.9: Surface pressure contour for ONERA M6

4.2 Application to non-primal grid system

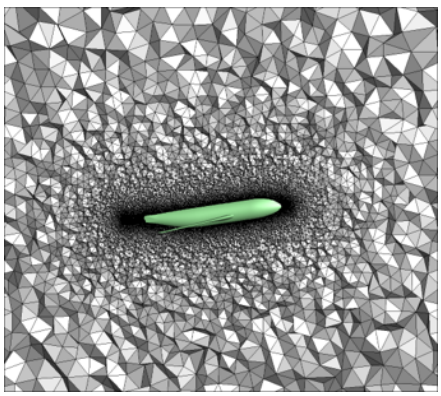
The previous results are based on tetrahedral grids. Local point clouds for primal grids, which are represented as tetrahedron and prism, are generally well-defined. Thus, an octree-structured grid system[56, 57] with prism (OctP) is tested to demonstrate the versatility and robustness of the multicloud method. Each level domain is discretized by GC-LSM based on Octree-prism grids. The nodes of volumes are used for meshless discretization. AUSMPW+ and minmod limiter were used and LU-SGS was used for the time integration.

4.2.1 NACA0012

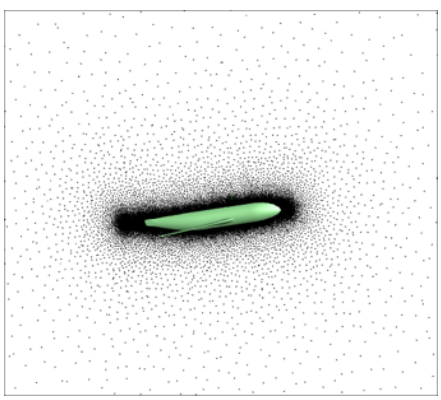
In order demonstrate the efficacy of the multicloud method. NACA0012 airfoil with 2D version of OctP (QuaP) is tested. As shown Figure 4.15, the node of multiscale Cartesian and O-type mixed grid are used for meshless domain. Numerical experiments are tested for subsonic flows (Mach 0.5 and AOA $3°$ and transonic flows (Mach 0.85 and AOA $1°$). Figure 4.16 shows that convergence history with respect to the iteration for subsonic flows. As seen from Figure 4.15, even though the resulting coarse



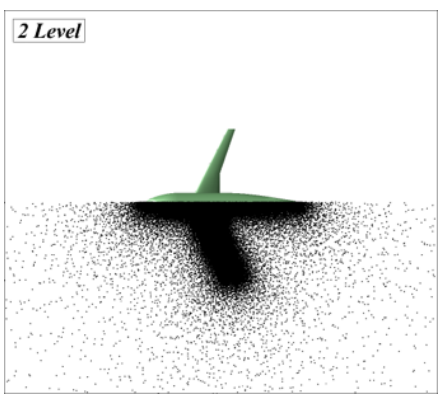
(a) ONERA M6 configuration



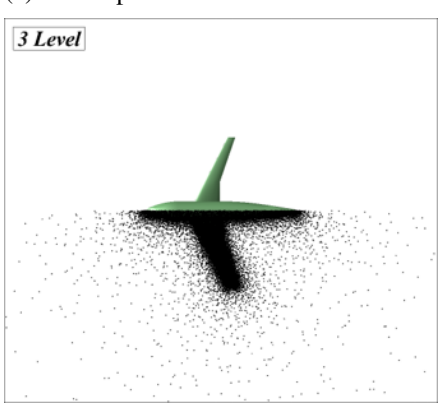
(b) Tetrahedral grids for DLR-F6



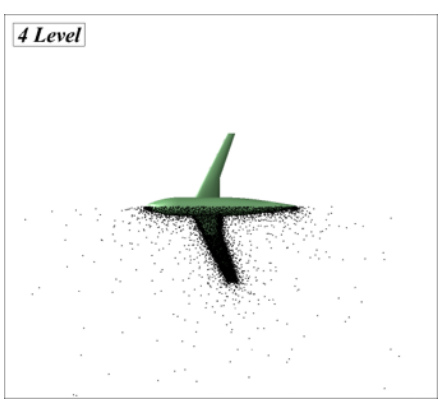
(c) Global point cloud based on the nodes



(d) Coarse level cloud for level 2

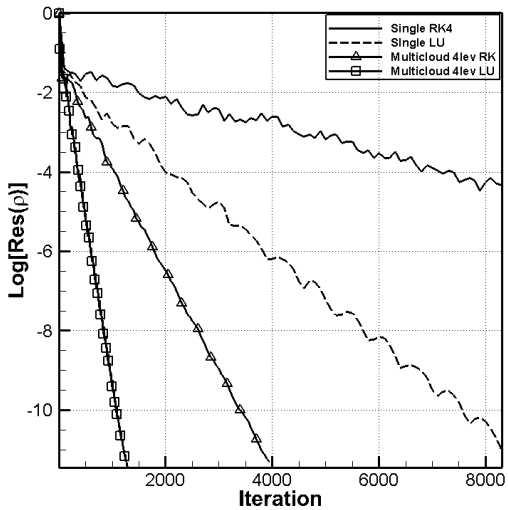


(e) Coarse level cloud for level 3

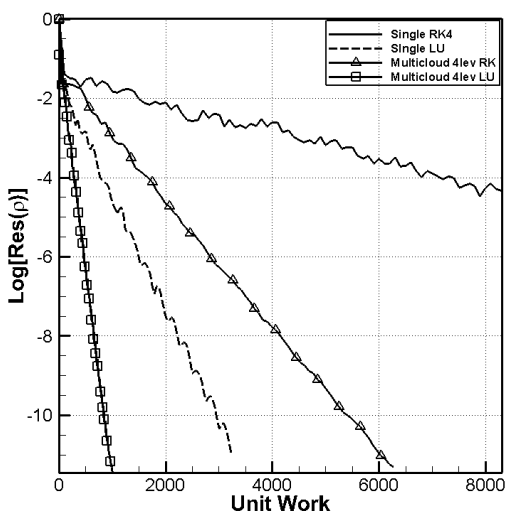


(f) Coarse level cloud for level 4

Figure 4.10: Coarsening results for DLR-F6



(a) Convergence history in term of iteration



(b) Convergence history in term of unit work

Figure 4.11: Convergence histories for DLR-F6

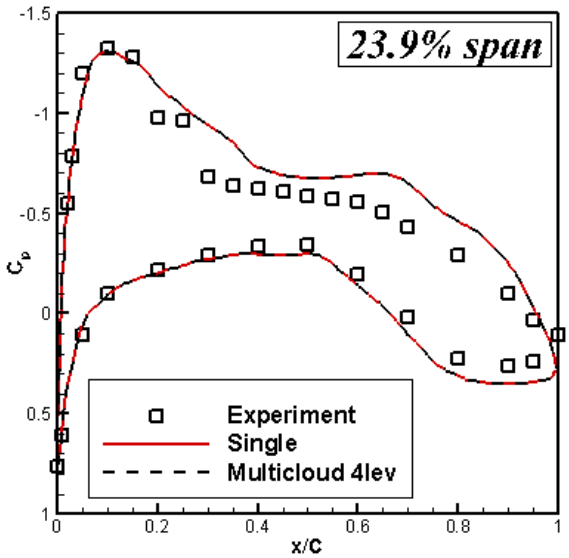


Figure 4.12: Surface pressure along $y = 0.239\eta$

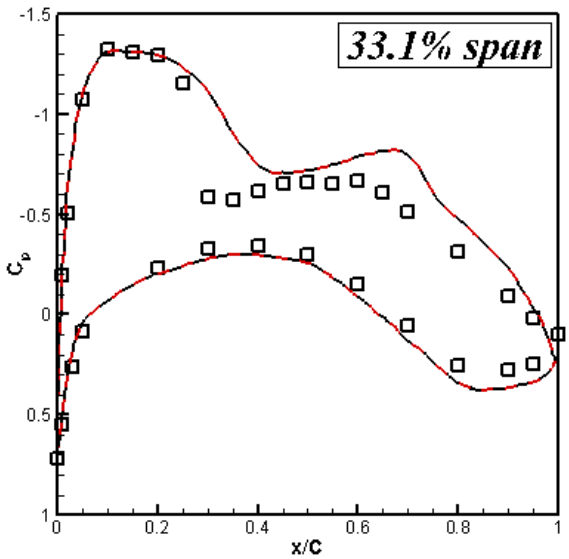


Figure 4.13: Surface pressure along $y = 0.331\eta$

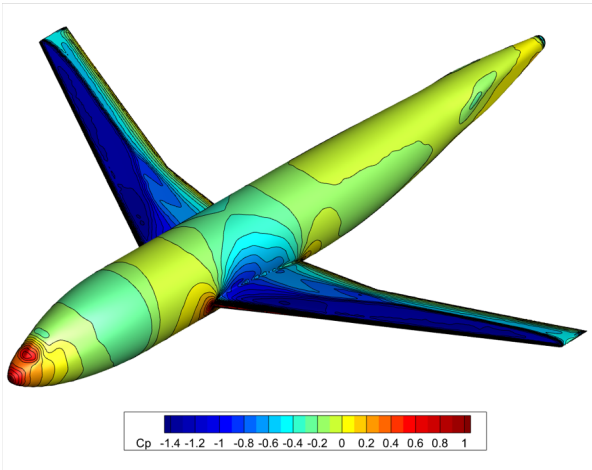


Figure 4.14: Surface pressure contour for dlr-f6 wing body

domain is randomly distributed, the convergence acceleration effect is still dramatic without solution differences(See Figure 4.17). The transonic condition is also investigated using the same point clouds. Figure 4.18 shows the convergence history for transonic flows. A slight decrease in acceleration is observed due to the presence of shock waves, which can degrade the acceleration effect. Nevertheless, the effect of speed up is significant. The effectiveness of the multicloud method is demonstrated, even for points that are significantly randomly distributed from the results.

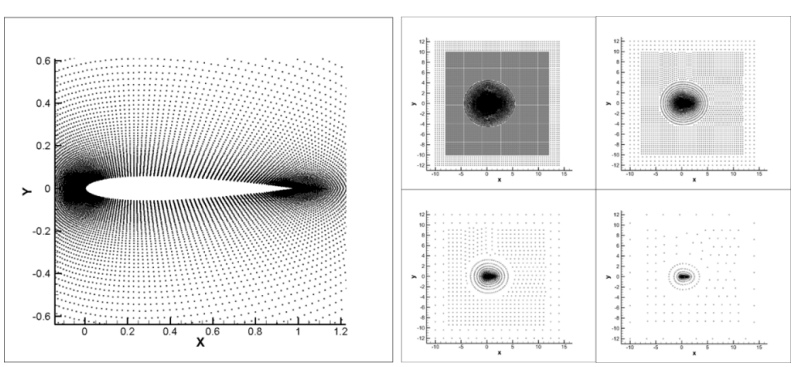


Figure 4.15: Illustration of computational domains for NACA0012

Table 4.2: The number of point comparisons for each level for NACA0012

level	the number
1(points)	32,589
2(points)	8,155
3(points)	1,996
4(points)	485

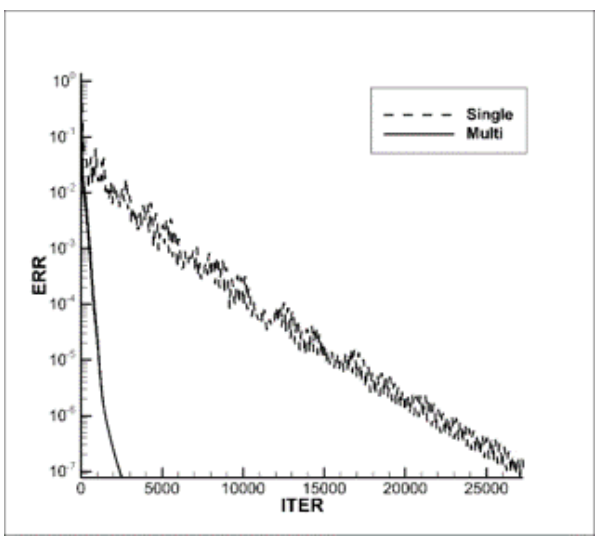


Figure 4.16: Convergence history for NACA0012, M=0.5 AOA=3° (QuaP)

4.2.2 ONERA M6

To verify the validity of the method in three dimension, ONERA M6 with OctP grids is tested. The coarsening results are shown in Figure 4.20. The number of points on fine level is 2,215,030. Then, the number of points reduced as 380,957, 64,278, 11,142 as the level decreases. The multicloud coarsening method is effective for OctP grids, whereas it can be challenging to use traditional geometric multigrid methods.

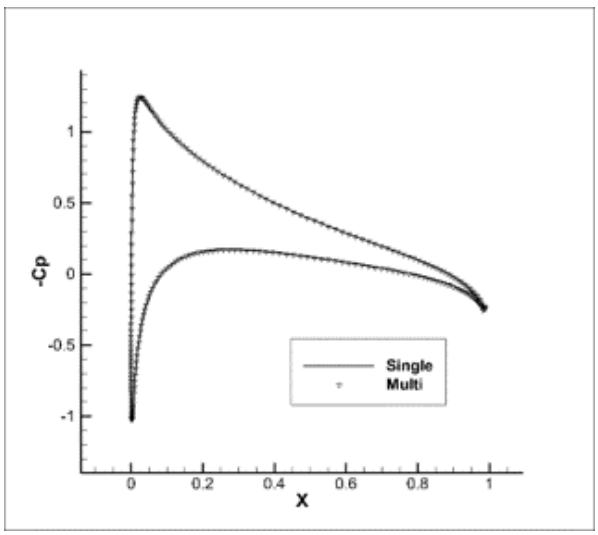


Figure 4.17: Surface pressure for NACA0012, M=0.5 AOA=3° (QuaP)

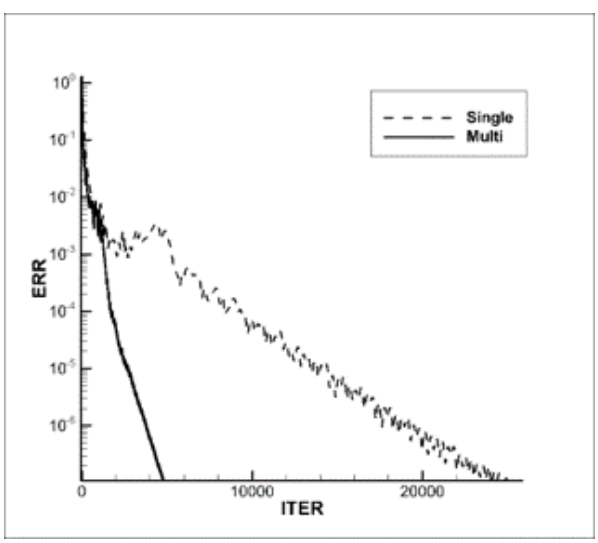


Figure 4.18: Convergence history for NACA0012, M=0.85 AOA=1° (QuaP)

In addition to the coarsening results, it is necessary to demonstrate the acceleration effect on convergence. Figure 4.21 represents convergence history in terms of iteration. The multicloud results show a convergence acceleration factor of five compared

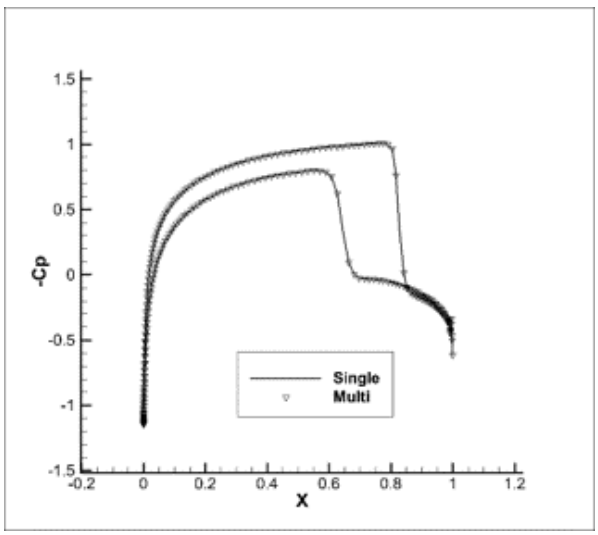
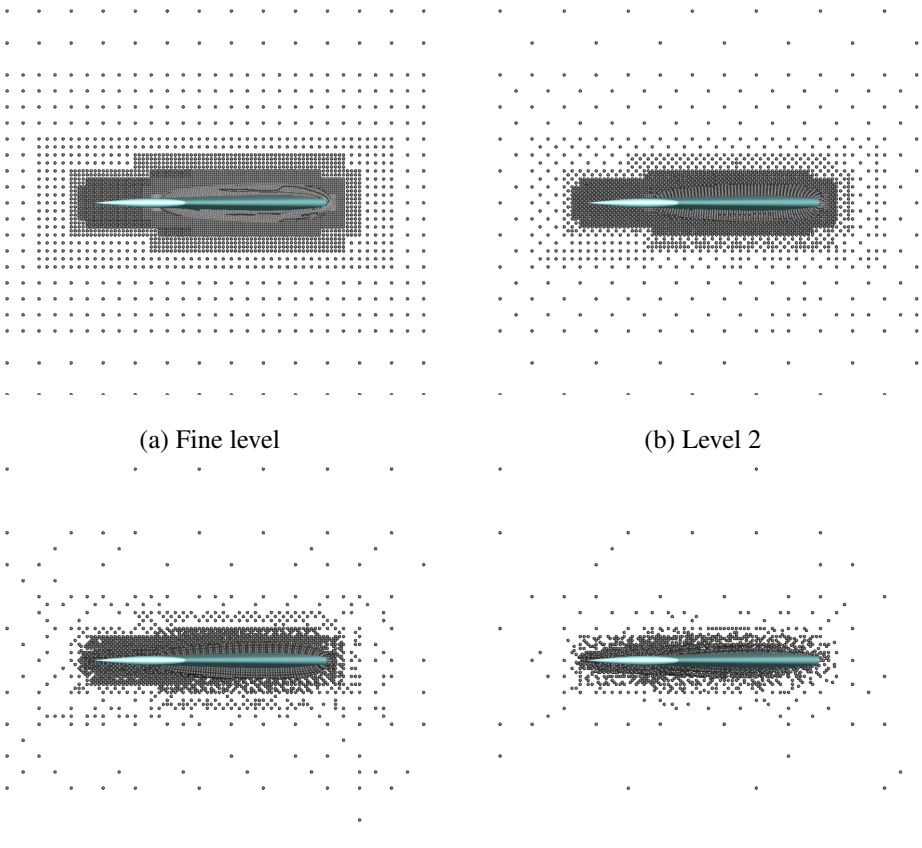


Figure 4.19: Surface pressure for NACA0012, M=0.85 AOA=1° (QuaP)

to single grid. The time spent per iteration of the multicloud method is a factor of 1.5 compared to that of the single-grid method. Figure 4.22-4.27 show the comparison between single and multilevel solutions. It is evident that two solutions rarely show the discrepancy. Thus, it is evident that the multicloud method provides an effective convergence acceleration effect for any grid type. It seems that the multicloud definitely provides the satisfactory convergence acceleration regardless of grid types and dimension.



(a) Fine level

(b) Level 2

(c) Level 3

(d) Level 4

Figure 4.20: Each level point distribution based on octree-prism for ONERA M6

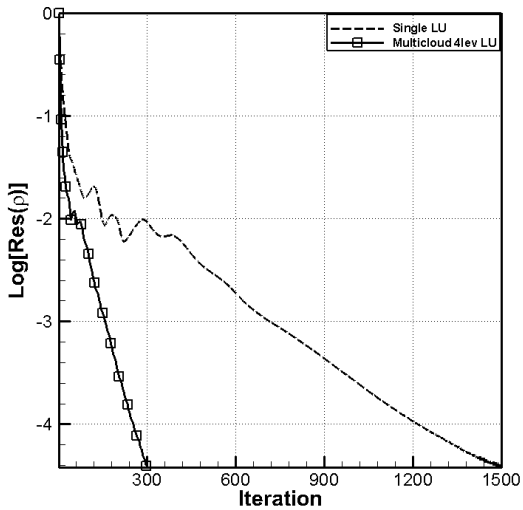


Figure 4.21: Convergence history for ONERA M6 (OctP)

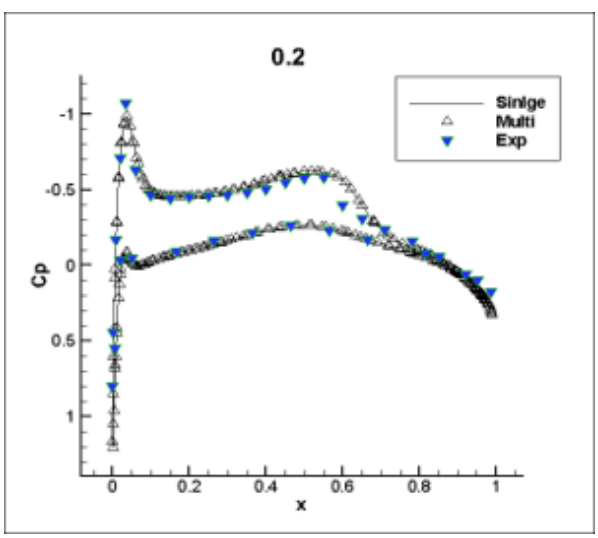


Figure 4.22: Surface pressure along y=0.2η

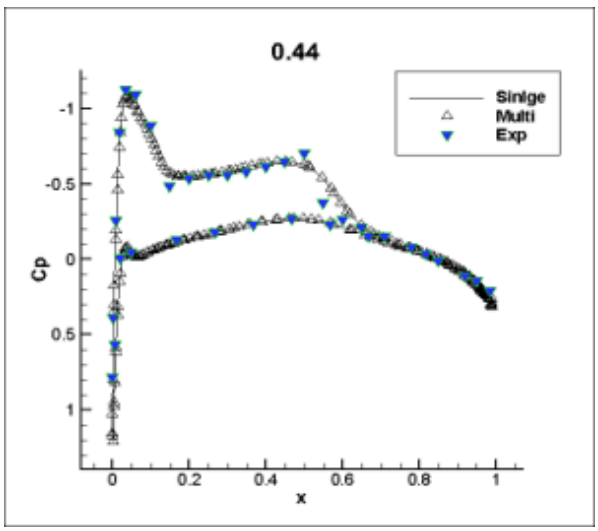


Figure 4.23: Surface pressure along y=0.44η

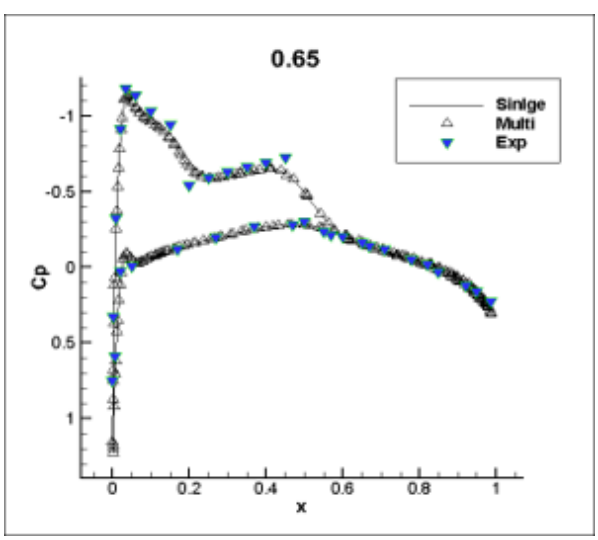


Figure 4.24: Surface pressure along y=0.65η

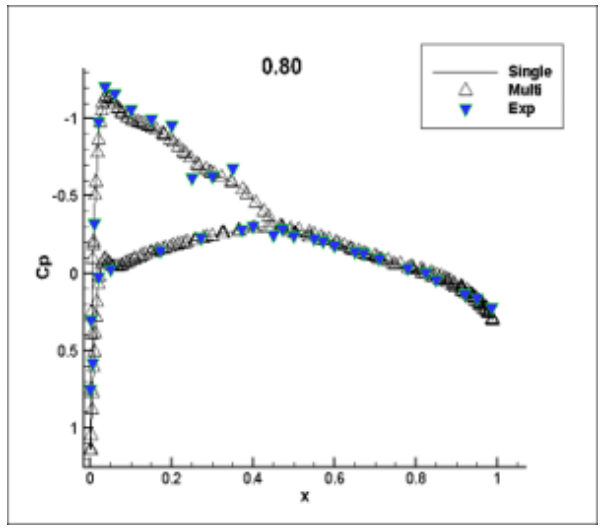


Figure 4.25: Surface pressure along y=0.8η

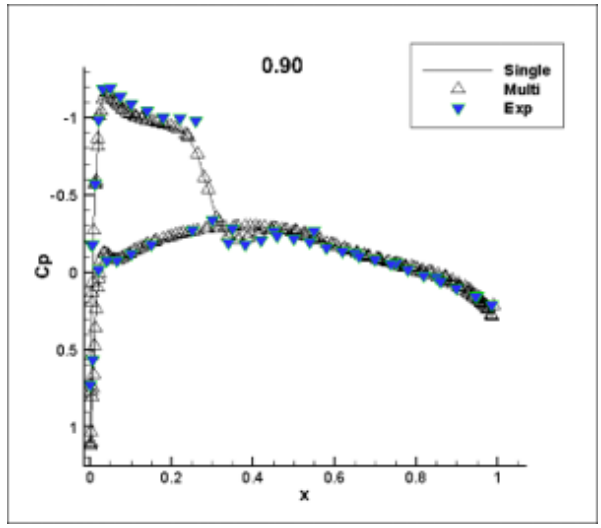


Figure 4.26: Surface pressure along y=0.9η

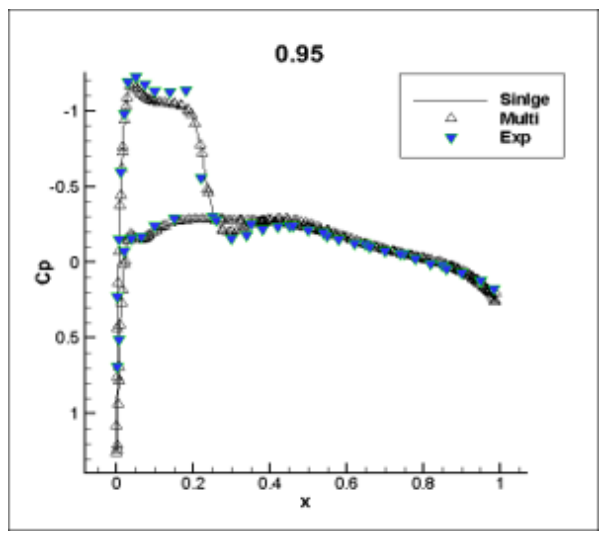


Figure 4.27: Surface pressure along y=0.95η

4.3 Cell coarsening strategy comparisons

In this section, the acceleration performance of different cell coarsening procedures is analyzed. The difference between CFA, the basic approach, and CNA, the improve approach, is presented. Cell finite volume discretization is employed for fine level domain and GC-LSM meshless discretization is employed for coarse level domain. AUSM$^+$-up is used on the all levels as a numerical flux scheme and LU-SGS is also used for all levels. Numerical experiments are tested for the three-configurations, ONERA M6, DLR-F6 wing body (f6wb) and DLR-F6 wing body nacelle pylon (f6wbnp). Especially, f6wbnp geometry is considerably complicated that the robustness of multicloud method might be highlighted.

4.3.1 DLR-F6 wbnp configuration

The first validation case is DLR-F6 wing body nacelle pylon(f6wbnp) configuration that was used for AIAA drag prediction workshop[41]. The geometries were based on those of Laflin et al.[41]. f6wbnp has a significantly complex geometry, as shown in Figure 4.28. For this reason, the robustness of CNA coarsening is uncertain for other non-conservative meshless methods. In this test, comparisons are provided not only between CFA and CNA but also between LSM and GC-LSM to demonstrate the performance of GC-LSM with CNA coarsening. AUSM+up is used for every case. And Venkakrishnan limiter is used for CFV exclusively.

Then, four-cases are tested for the demonstration as shown in Table 4.4. M-LU is combined with non-conservative schemes and U-LU is combined with conservative schemes. Every multicloud strategy is four-level V cycle with CFL condition 15. The convergence histories for each case are presented in Figure 4.29. As seen in Figure

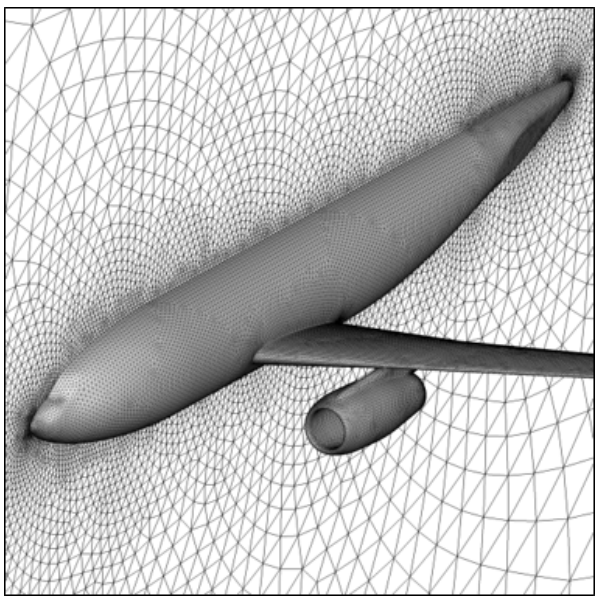


Figure 4.28: Configuration of f6wbnp and its flow grid

4.29, CFA with LSM and CNA with GC-LSM showcase dramatic convergence acceleration than single-grid method. It is evident that CNA grants much higher level of acceleration since average distance among point is further than that of CFA. However, CNA with LSM cannot converge to the same level of CNA with GC-LSM. This is because non-conservative meshless discretizations is not guaranteed its stability for highly coarsened domains such as CNA. It seems that LSM discretization generated more error than GC-LSM on CNA coarsened domain. Figure 4.30 shows the surface density contour obtained by fully converged results computed by each meshless discretization on CNA level 2 domain by the single grid method. As seen in Figure 4.30a & 4.30b, LSM discretization generated more error on the surface than GC-LSM. Those errors involved by LSM might inhibit convergence in the multi-cloud method. Thus, only CNA coarsening with GC-LSM might guarantee dramatic acceleration effect than any other methods.

Furthermore, Figure 4.31 illustrate convergence histories between Single-grid, CFA and CNA with respect to walltime, which denotes the wall clock time. CNA 4 level converges only in 2000 seconds while that of CFA 4 level is 3900 seconds. It is 15000 seconds for single-grid method. The speed-up for CNA 4 levels is the factor of 7.5 while that of CFA 4 levels is 3.84. CNA with GC-LSM provides twice the speed-up than that of CFA 4 level. The acceleration effect is significantly dramatic for the highly complex configuration of f6wbnp. As seen in Figure 4.32 and 4.33, no solution discrepancy appears.

Table 4.3: The number of point comparisons for each level

level	CNA	CFA
1(cells)	5,809,122	
2(points)	229,918	2,123,670
3(points)	46,806	252,575
4(points)	10,170	48,951

Temporal	Coarsening	Spatial
U-LU	Single grid	CFV
M-LU	CFA 4 level	1 : CFV & 2-4 : LSM
M-LU	CNA 4 level	1 : CFV & 2-4 : LSM
U-LU	CNA 4 level	1 : CFV & 2-4 : GC-LSM

Table 4.4: Description for test cases

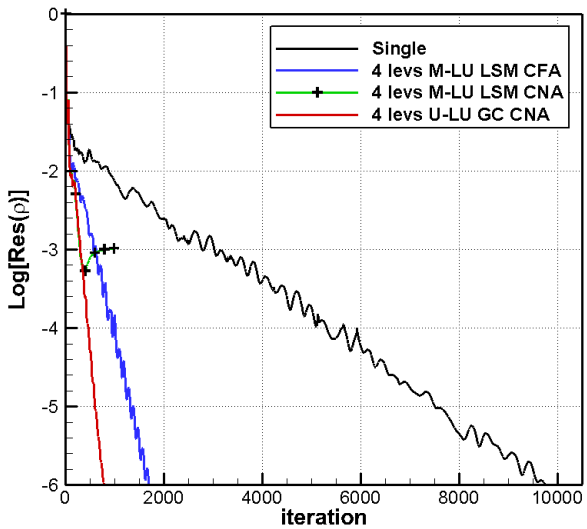
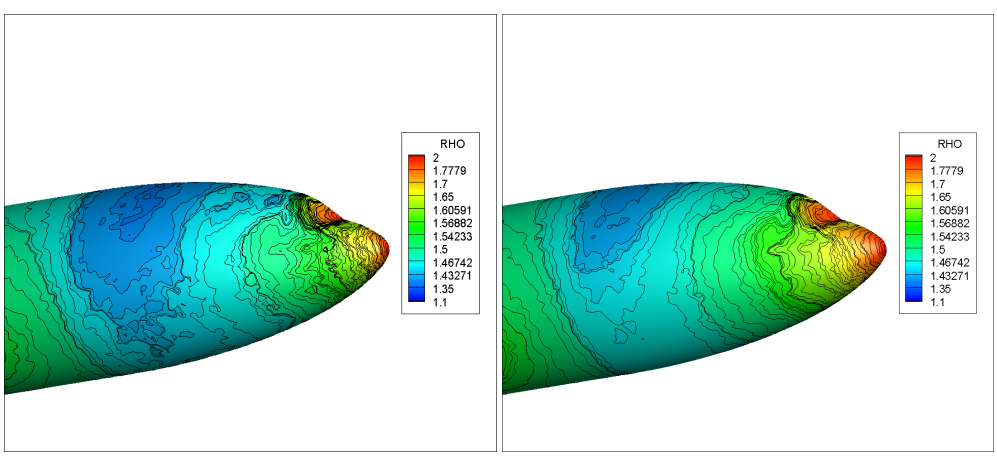


Figure 4.29: Convergence histories for f6wbnp in term of iteration



(a) LSM (b) GC-LSM

Figure 4.30: Surface results of fully converged solutions on level 2 point domains

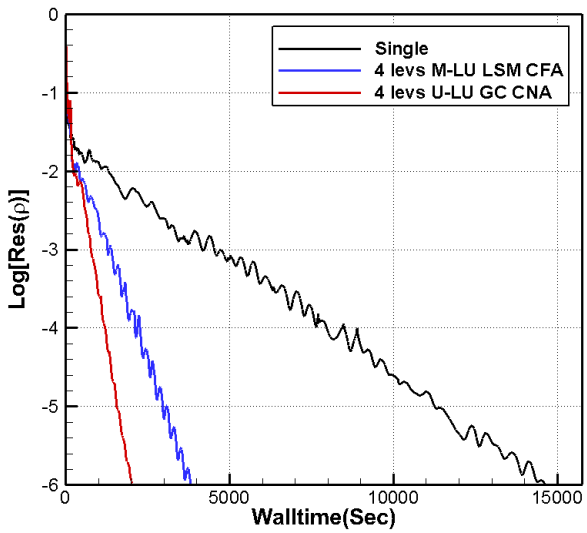


Figure 4.31: Convergence history of f6wbnp in terms of the walltime

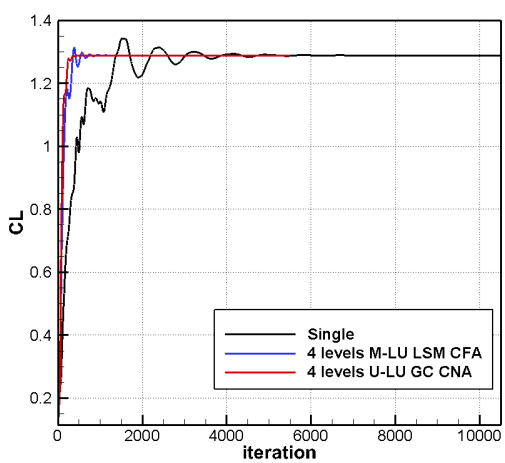


Figure 4.32: Convergence history of lift coefficient for f6wbnp in terms of the iteration

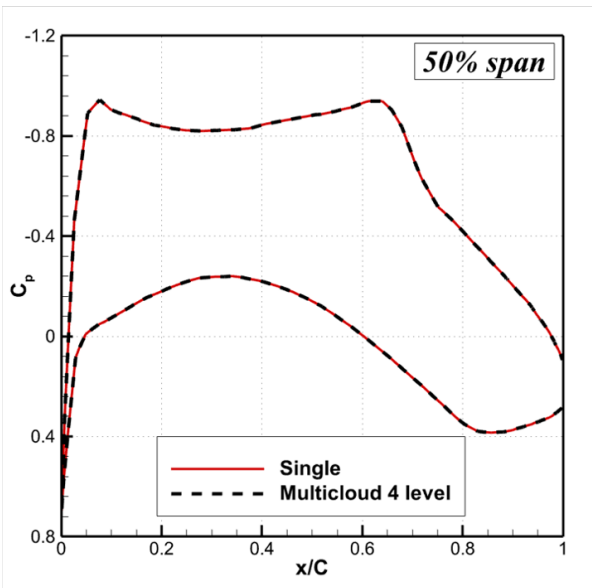


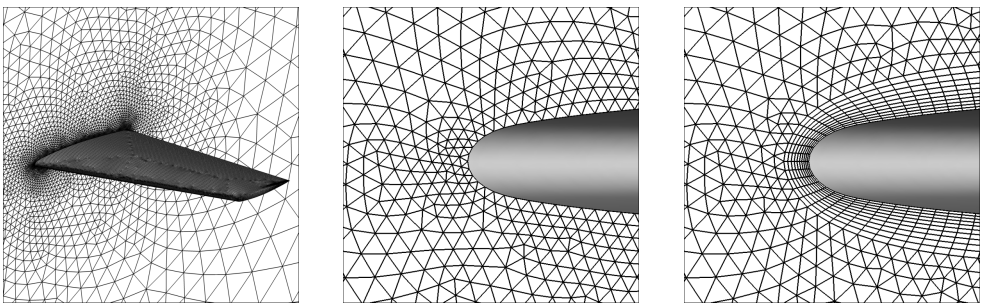
Figure 4.33: Surface pressure comparison for f6wbnp

4.3.2 ONERA M6

To verify the performance of the CNA coarsening procedure, the transonic flow around ONERA M6($M = 0.8395$ and $\alpha = 3.06°$) was analyzed. To demonstrate the mesh transparency of CNA coarsening, tetrahedral grids (TG) and hybrid grids (HG), which are mixtures of tetrahedrons and prisms, were tested, as shown in Figure 4.34. Discreitzation of fine level was carried out by CFV for every case. AUSM+up was used for numerical flux compuation with unstructured MINMOD limiter. Table 4.5 lists the numbers of points for each case. Figure 4.35 illustrates the coarsening results of the tetrahedral grid for both the CNA (TG-CNA) and CFA (TG-CFA) systems. CFL number was 5 for TG and 2 for HG for LU-SGS. As described in Table 4.5, the CNA coarsening procedure provides coarser point clouds than the CFA. At level 4, the coarsening rate was 0.00187, which was better than the ideal coarsening rate of 0.00195 obtained in the three-dimensional space, whereas the CFA coarsening rate was 0.00680. Furthermore, the CFA strategy shows unsatisfactory coarsening rates at Level 2, which is a dominant factor in determining the success of multigrid methods in terms of time and iterations. Figure 4.36-4.38 show the convergence histories of TG for each coarsening strategy. As mentioned above, the CFV method was used at the finest level for all cases; therefore, meshless discretization was used only on the coarse-levels and four-level V cycle was applied. In Figure 4.36, the x-axis indicates the number of iterations required on a fine-level grid. As shown in Figure 4.36, both TG-CFA and TG-CNA show a dramatic acceleration effect compared with the single-grid method in terms of iterations. The CNA strategy reaches an order of -8.5 in approximately 2800 iterations, such that the required fine-level iterations are less than half of the 6500 iterations from the CNA strategy. Figure 4.37 shows the

wall time vs. the residuals. In addition, in terms of the wall time, TG-CNA showed greater convergence acceleration behavior than TG-CFA. The lift coefficient convergence histories are presented in Figure 4.38. Similar to the residual histories, the lift coefficient of TG-CNA converged significantly faster than those of the other methods. Figure 4.39-4.44 show comparisons of the surface pressures. As shown in the figures, the results from the four-level and single grid CNA rarely differ from the single grid results. Both the single- and four-level models also showed good agreement with the experimental results[53]. According to these findings, CNA greatly improves the speed of these systems without losing its robustness and accuracy, despite the scarcity of the point distribution on level 2.

To demonstrate the mesh-transparent characteristics of the CNA coarsening strategy, numerical experiments were performed using HG. In Table 4.5, the number of coarse-level points is described for the HG case (HG-CFA & HG-CNA). Figure 4.45-4.47 show the convergence histories. Similar to TG, CNA presented exceptional coarsening rates compared to CFA for HG. In comparison to the TG case, all the other cases exhibit slower convergence due to the presence of HG involving high-aspect-ratio grids, which can result in delays convergence. However, the acceleration effect is more significant than that of TG, which can be attributed to the regularity of the prism coarsening. The results indicate that CNA coarsening provides the most powerful acceleration effect regardless of grid type.

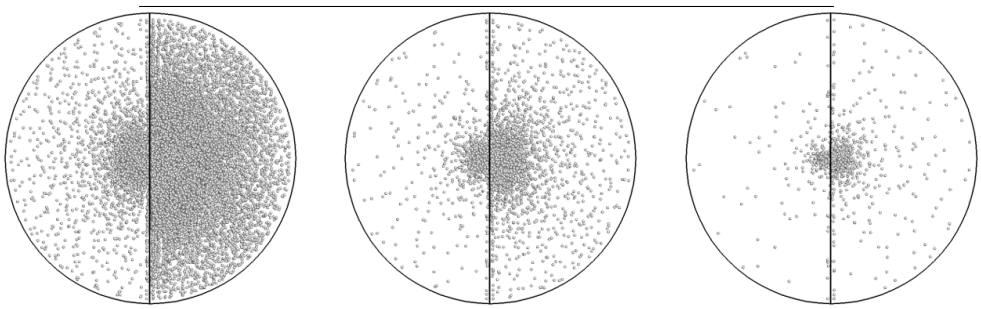


(a) (b) (c)

Figure 4.34: Illustration of the unstructured grid for the ONERA M6 wing. (a),(b) Tetrahedral grid (TG), (c) hybrid grid (HG).

Table 4.5: The number of point comparisons for each level

level	TG-CNA	TG-CFA	HG-CNA	HG-CFA
1(cells)	1,540,522		2,336,161	
2(points)	62,190	557,926	124,500	923,865
3(points)	12,084	56,371	22,626	117,623
4(points)	2,877	10,484	4,660	22,695



(a) Level 2 point cloud (b) Level 3 point cloud (c) Level 4 point cloud

Figure 4.35: Comparison of the coarsening results between TG-CNA (left) and TG-CFA (right)

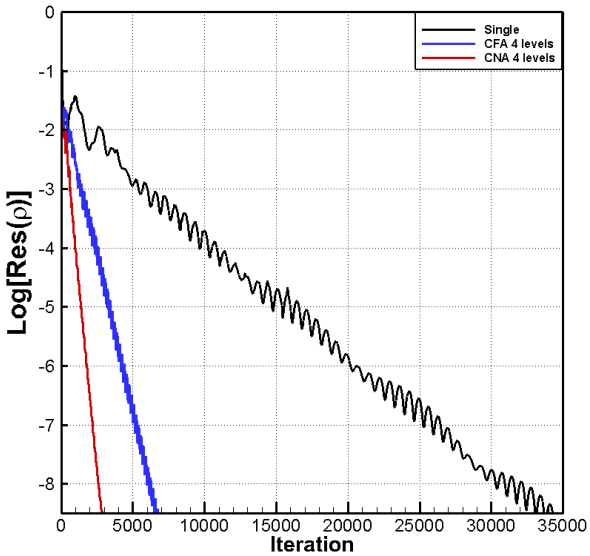


Figure 4.36: Convergence history in term of iteration for transonic ONERA M6 flows (Tet)

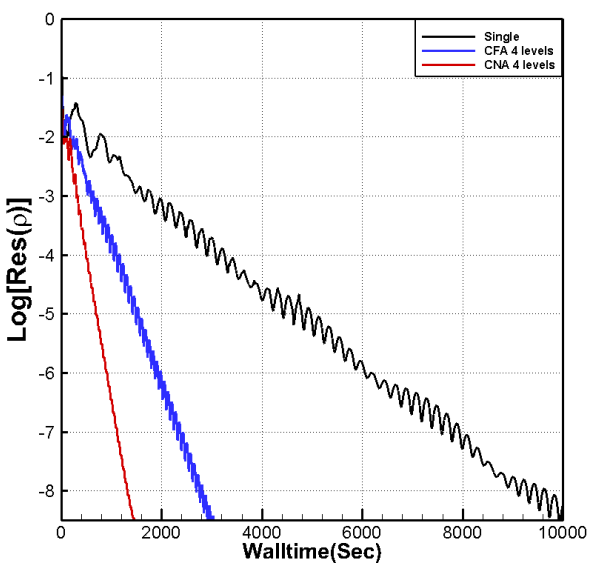


Figure 4.37: Convergence history in term of walltime for transonic ONERA M6 flows (Tet)

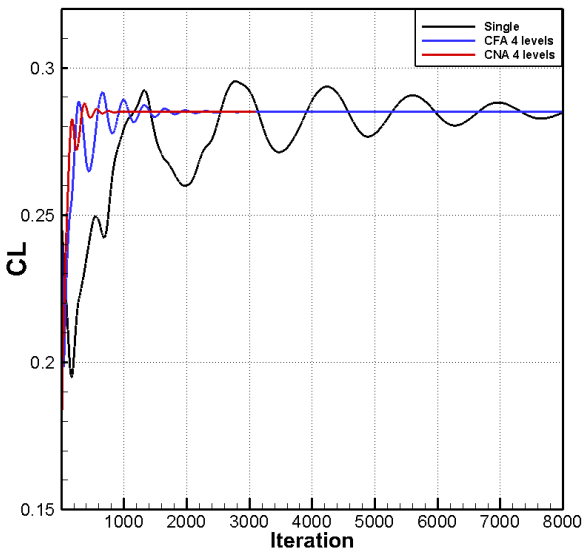


Figure 4.38: Lift convergence history in term of iteration for transonic ONERA M6 flows (Tet)

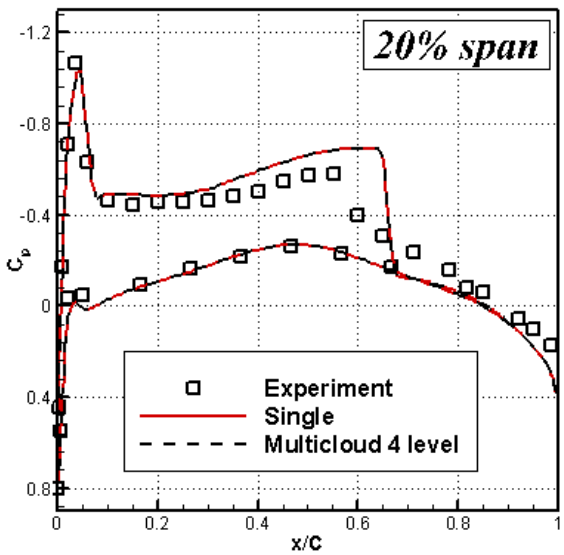


Figure 4.39: Surface pressure along y=0.2η

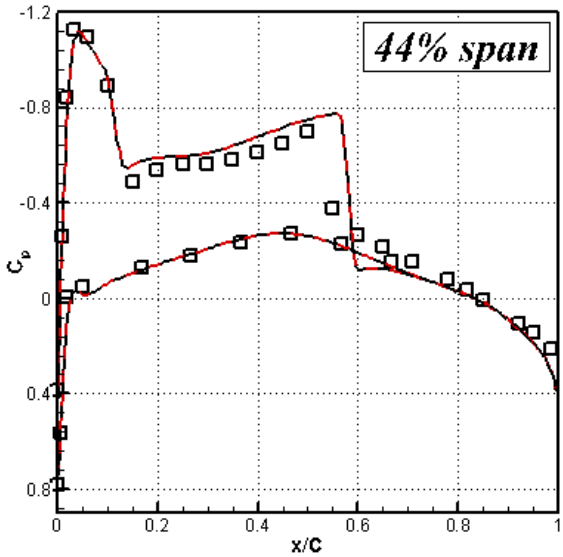


Figure 4.40: Surface pressure along y=0.44η

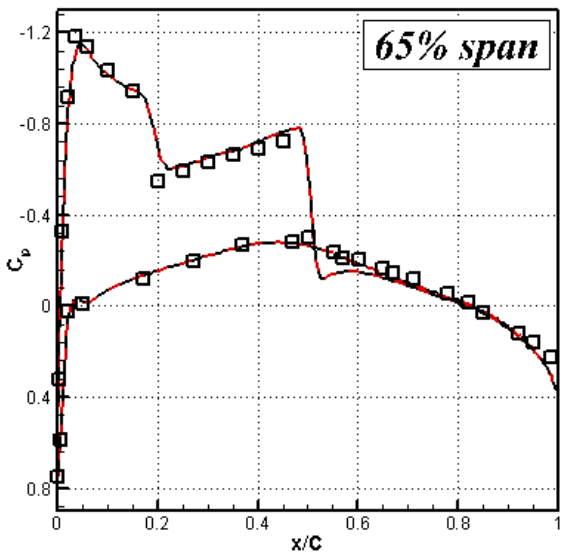


Figure 4.41: Surface pressure along y=0.65η

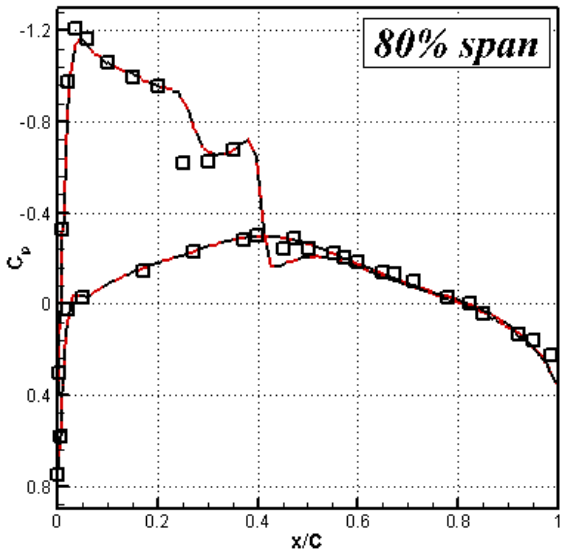


Figure 4.42: Surface pressure along y=0.8η

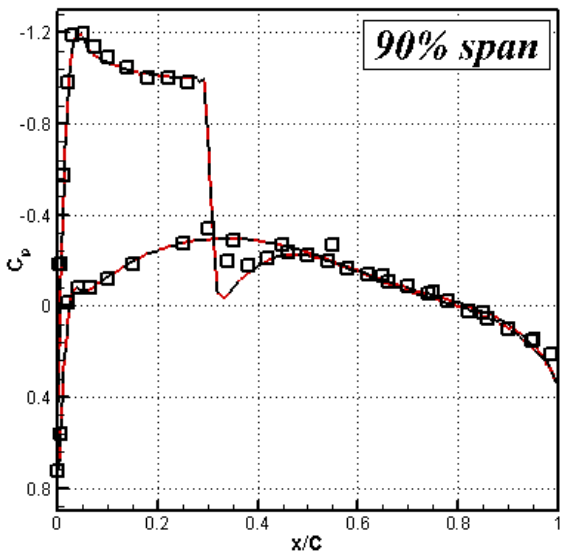


Figure 4.43: Surface pressure along y=0.9η

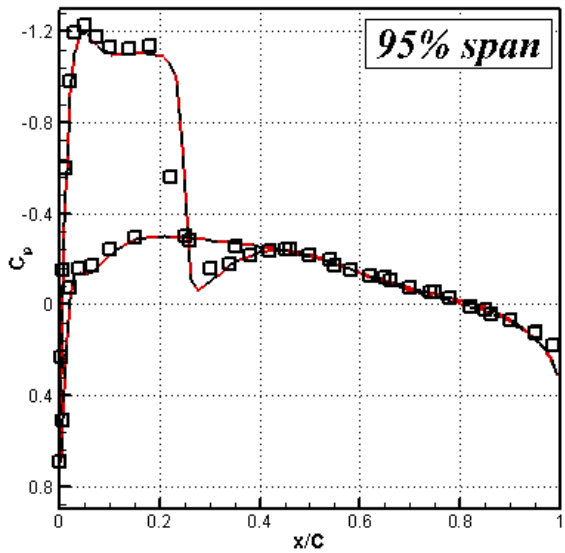


Figure 4.44: Surface pressure along y=0.95η

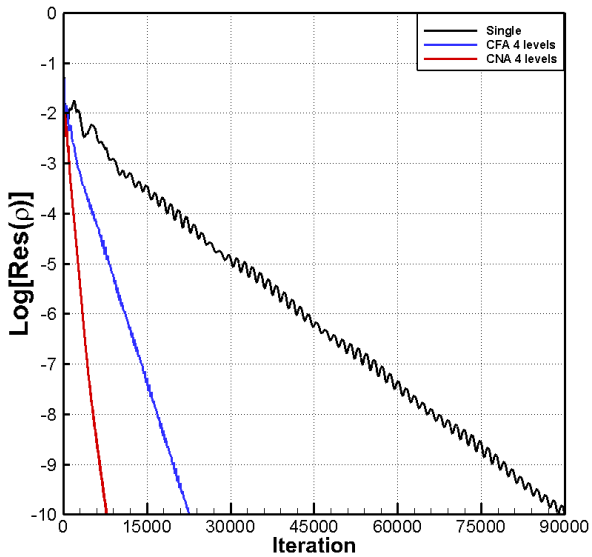


Figure 4.45: Convergence history in term of iteration for transonic ONERA M6 flows (Hybrid)

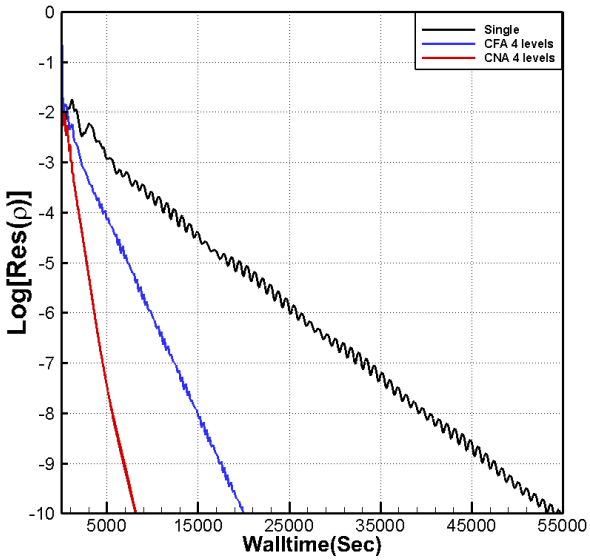


Figure 4.46: Convergence history in term of walltime for transonic ONERA M6 flows (Hybrid)

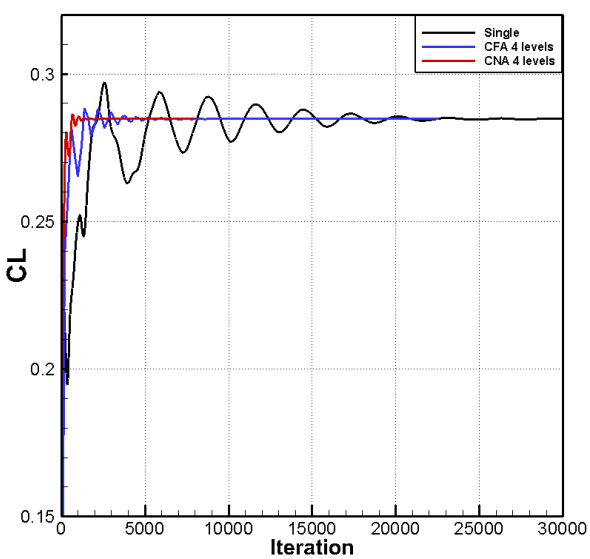


Figure 4.47: Lift convergence history in term of iteration for transonic ONERA M6 flows (Hybrid)

4.3.3 DLR-F6 wing body configuration

To highlight the performance of the CNA coarsening procedure for complex geometries, the DLR-F6 wing body (f6wb) model was tested. CFV was used for the finest level discretization with unstructured MINMOD limiter and LU-SGS with CFL 2. The geometries were also based on those of Laflin et al.[41]. The flow simulation was tested using both CNA and CFA. The tetrahedral grids for this case are shown in Figure 4.48. A numerical simulation was performed for $M = 0.75$ and $\alpha = 0.48°$. coarsening results on the surface grid are shown in Figure 4.49-4.51. Surface points are evenly distributed which provide the robust and effective multicloud computation. The convergence histories of f6wb are presented in Figure 4.52-4.54. The four-levels CNA strategy reached an order of -8 in approximately 4800 fine-level iterations, while CFA required approximately 9000 iterations. Both CNA and CFA achieved a high acceleration compared to a single grid (55000 iterations). Furthermore, the results of the multicloud and single-grid methods do not significantly differ, as indicated in Figure 4.54 and 4.55. The results of f6wb demonstrate that the CNA coarsening strategy works well for three-dimensional complex geometries.

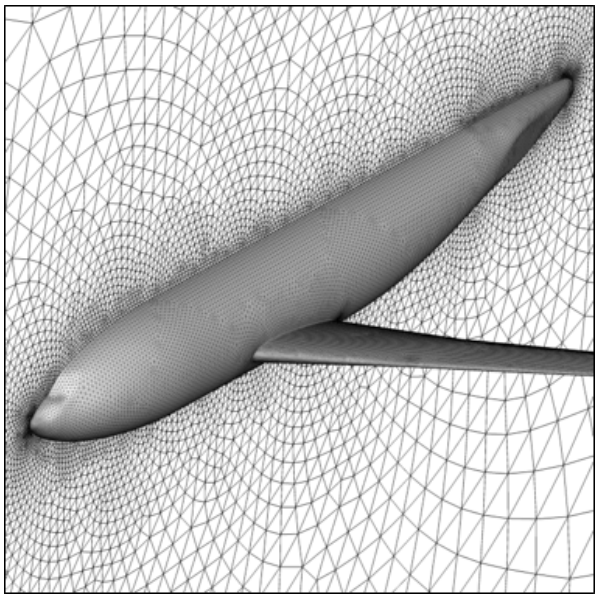


Figure 4.48: Configuration of f6wb and its flow grid

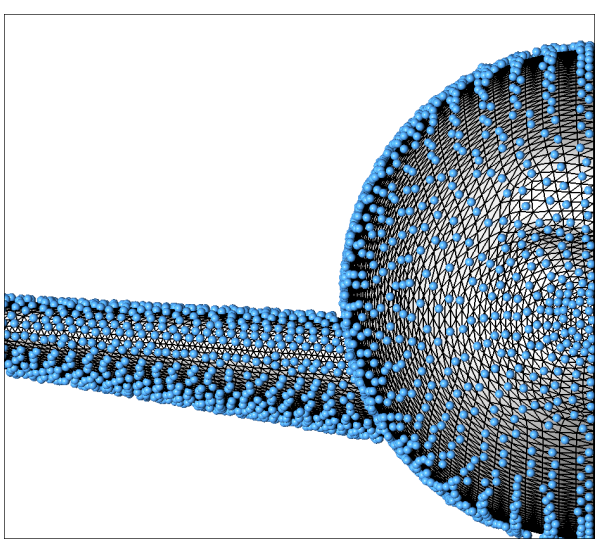


Figure 4.49: Frontal view of surface point distribution on level 2 for f6wb

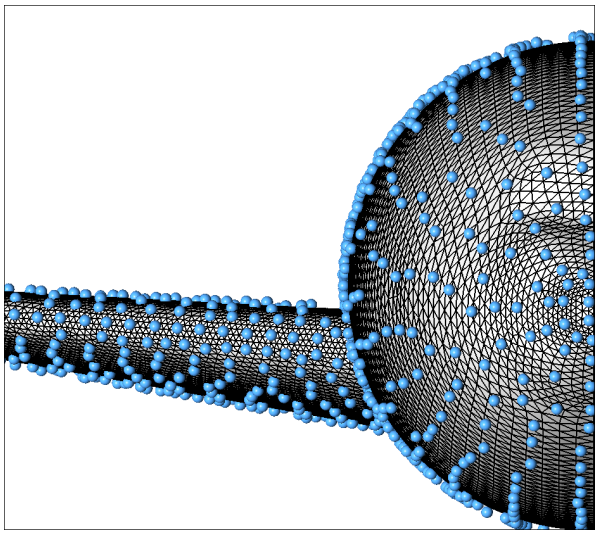


Figure 4.50: Frontal view of surface point distribution on level 2 for f6wb

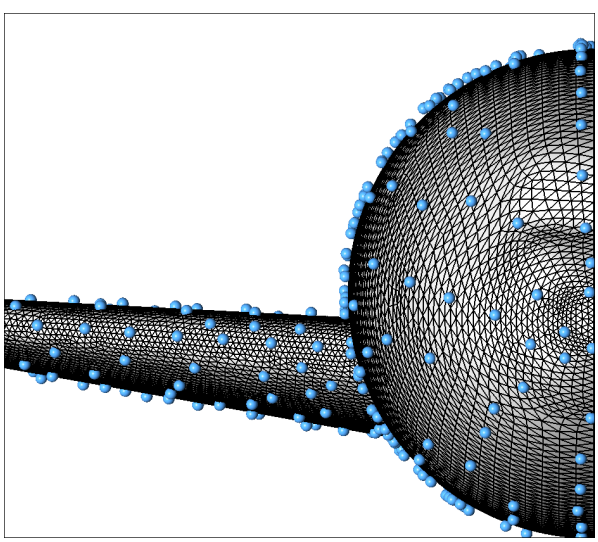


Figure 4.51: Frontal view of surface point distribution on level 4 for f6wb

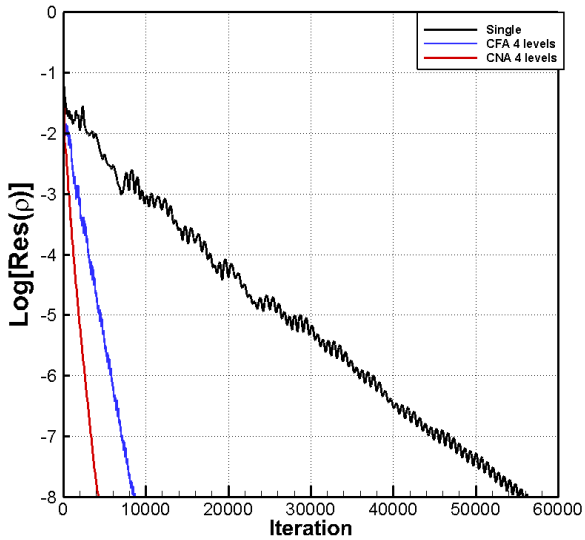


Figure 4.52: Convergence history of f6wb in terms of the iteration

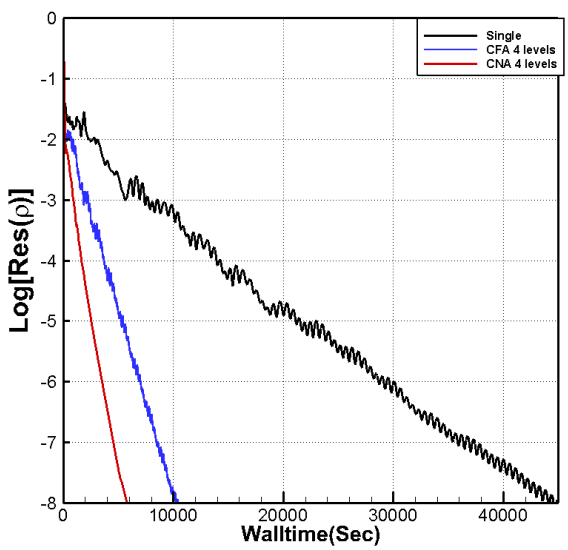


Figure 4.53: Convergence history of f6wb in terms of the walltime

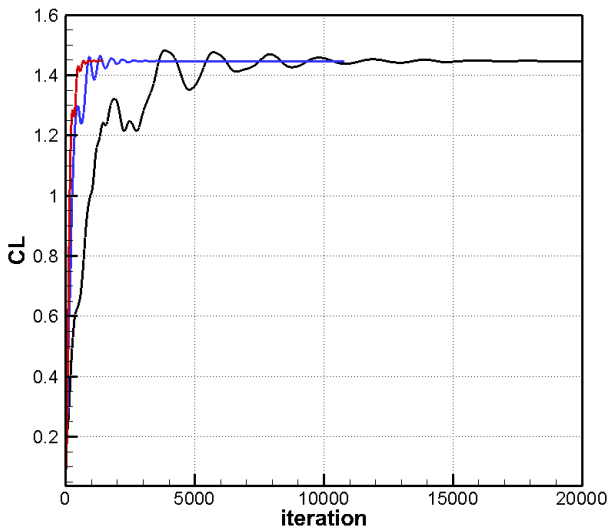


Figure 4.54: Convergence history of lift coefficient for f6wb in terms of iteration

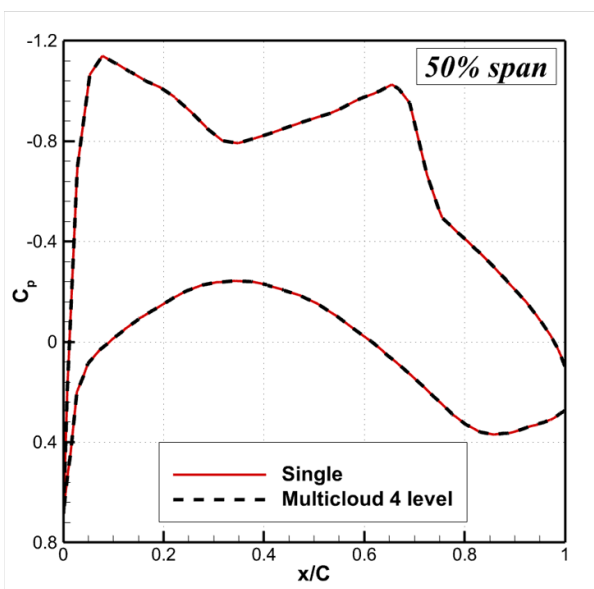


Figure 4.55: Surface pressure comparison for f6wb

4.4 Laminar flows

In this section, laminar flows were tested to demonstrate the validity of the multicloud method to viscous flows. NACA0012 and cylinder Laminar flows were tested. four-level V cycle with CNA coarening is applied to every test case.

4.4.1 NACA0012 Laminar flows

The first test case is Laminar flows around NACA0012 with Mach 0.5 and Reynolds number 5000. This test case is widely used to demonstrate developed numerical methods[58]. The goal of this case is not only comparing convergence histories but also scrutinizing the discrepancy of solution between single and multicloud method in viscous flows. To analyze viscous flows, blend of quadrature and triangle grids are used as seen in Figure 4.56. CNA coarsening was applied to the grids since CFV was used for discretization. the coarsening rates at level 2 is 0.168 and 0.31 for both 3 and 4 levels, which are the satisfactory coarsening rates for two-dimensional grids. Finally, four-level V cycle was used.

Finite discretization was carried out by CFV and AUSM+up with Venkatakrishnan limiter was used for numerical flux computation. LU-SGS was used with CFL condition 5. Estimation of gradient was computed by Nodal-based Green-Gauss Theorem for both reconstruction and viscous flux. Figure 4.57 and 4.58 illustrate convergence histories. As seen in Figure 4.57, the multicloud method needs 3537 iterations to achieve -14 order of convergence whereas the single method takes 77079 iterations. The speed-up is the factor of 22. This speed-up is by far the fastest case. It might be driven by viscous effect and grid quality. In term of walltime, the acceleration effect is rarely effected. The single grid case takes 2929 seconds whereas the

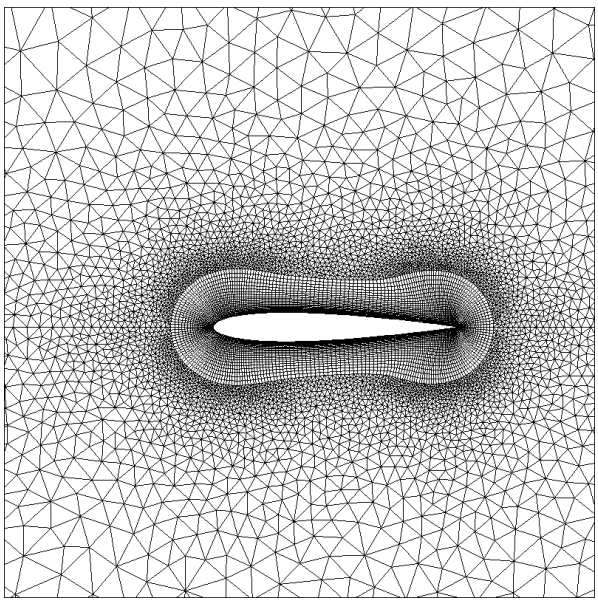


Figure 4.56: The flow grid for NACA0012

multicloud case take 235 seconds, which is the factor of 12.

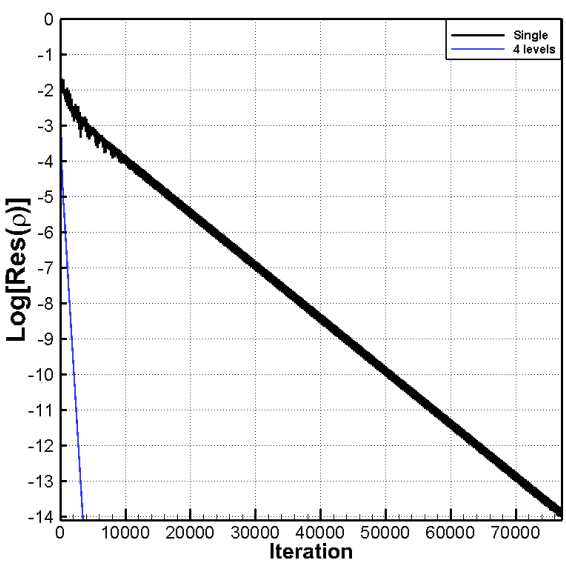


Figure 4.57: Density residual history for NACA0012 terms of iteration

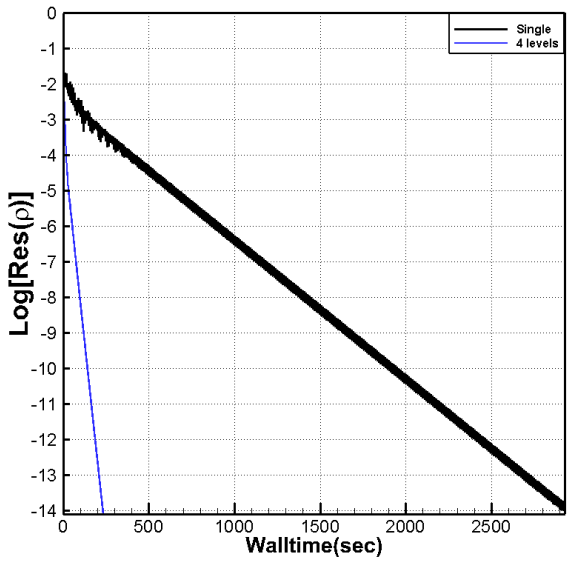
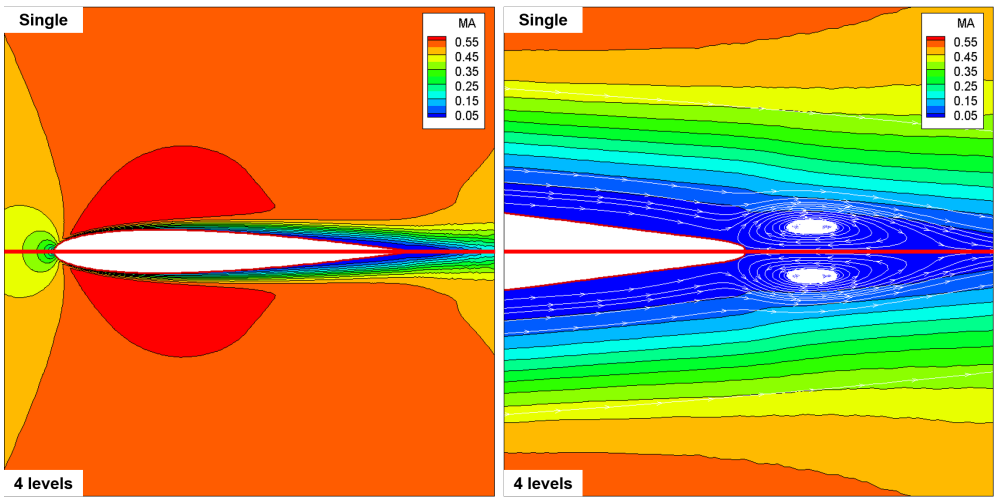


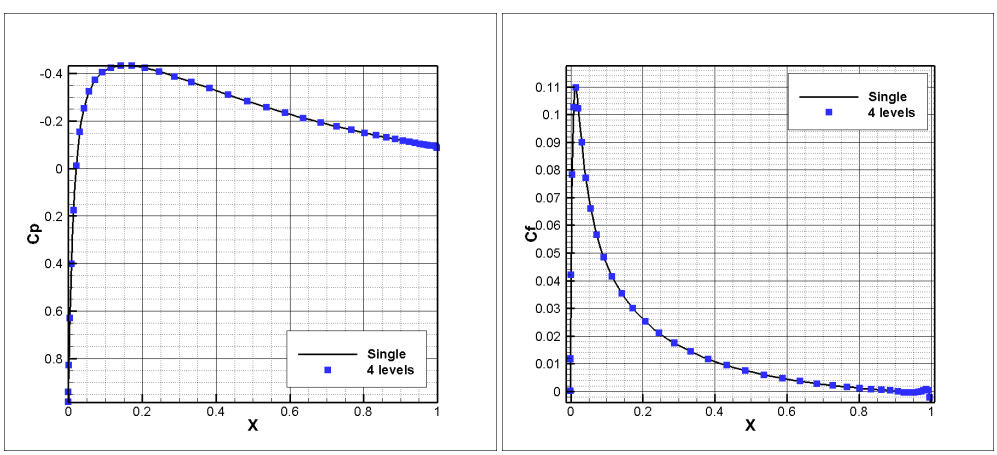
Figure 4.58: Density residual history for NACA0012 in terms of walltime

Figure 4.59 presents the comparisons of contour between the single and multi-cloud methods. As seen in Figure 4.59b, the separation exhibit excellent agreement between methods. Comparisons regarding surface coefficients are illustrated in Figure 4.60. As seen from the figures, surface pressure and skin friction coefficients exhibit no discrepancy even though the multicloud method convergence 12 times faster than the single grid method.



(a) Mach number contour (b) Streamline pattern in the trailing edge

Figure 4.59: Contour comparison for NACA0012 Laminar flows (Upper : Single, Nether : multicloud 4 levels



(a) Surface pressure coefficient (b) Skin friction coefficient

Figure 4.60: surface coefficient comparison for NACA0012 Laminar flows

4.4.2 Laminar flows around the cylinder (Re 40)

The next valdiation case for laminar flow is Mach 0.1 flow around the cylinder with Reynolds number 40. laminar cylinder flows is the famous benchmark for validation of laminar flows[59]. Re 40 cylinder flows is the famous test case for steady laminar flows without vortex shedding. Figure 4.61 illustrates the flow grid used in this case, which is the blend of quadrature and triangle grids. AUSM+up with Venkatakrishnan limiter and LU-SGS with CFL 2 were used. Furthermore, CNA coarsening with GC-LSM was applied with four-level V cycle.

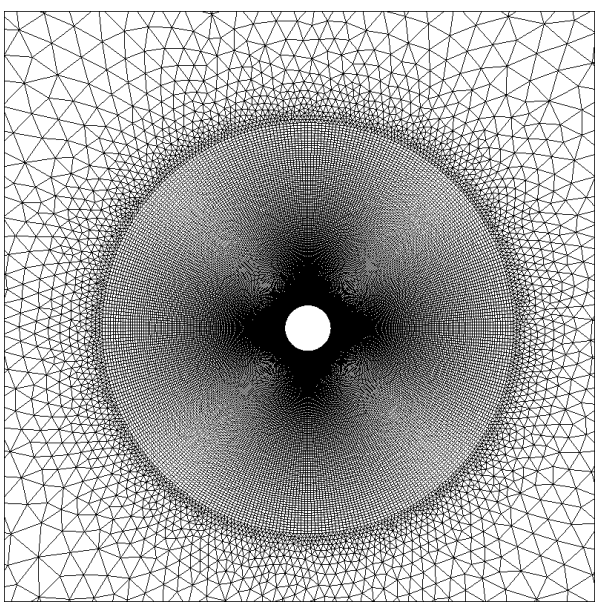


Figure 4.61: Flow grid for Re 40 cylinder flows

Convergence histories are plotted in Figure 4.62 and 4.63. The single-grid method takes 175,000 iterations to convergence which is extreme higher iteration numbers than those of the multicloud method (19295 iterations). In term of walltime, the multicloud method spend 13,168 seconds to convergence whereas walltime spent on the

single-grid method is 68,249 seconds. It is the factor of five. Re 40 cylinder flows exhibit less acceleration effect than Re 5000 NACA0012 laminar flows. It seems that the separation in the downstream that is shown in Figure 4.64 requires more iterations to be stabilized. As seen in Figure 4.64, the comparison of the location of separation in the downstream between the single-grid and multicloud show excellent agreements. In fact, the L2 norm of rho solution difference is less than 1E-10. Furthermore, Figure 4.65 illustrates comparisons of skin friction and pressure coefficient along the surface. It is evident that the two methods provide the same solution though the multicloud provides the solution 5 time faster.

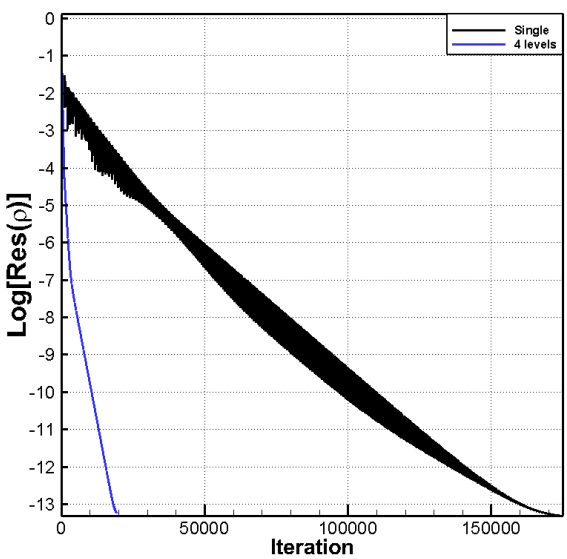


Figure 4.62: Density residual history of Re 40 cylinder flow in terms of iteration

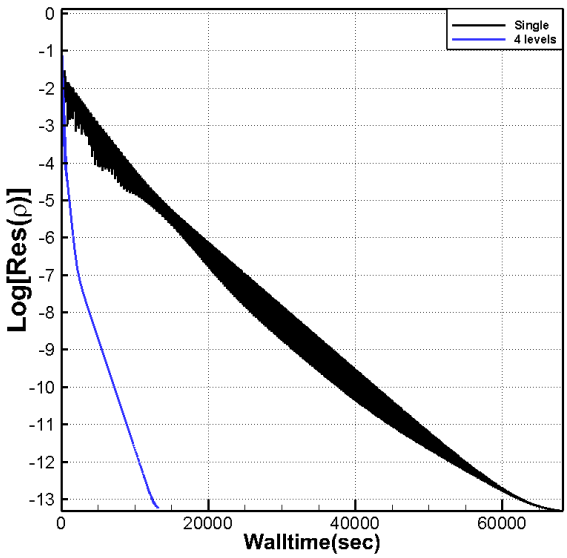


Figure 4.63: Density residual history of Re 40 cylinder flow in terms of walltime

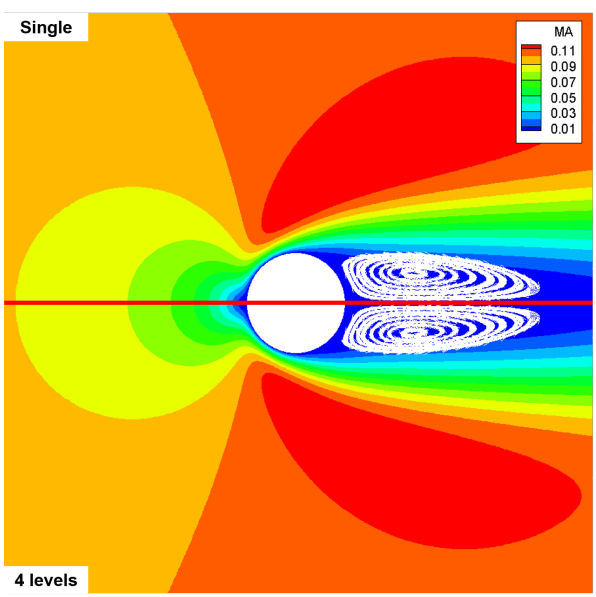
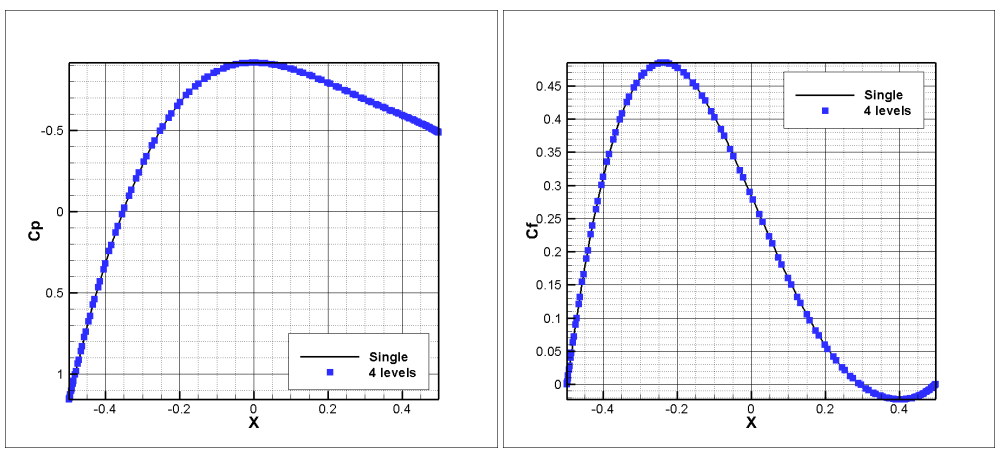


Figure 4.64: Streamline pattern in the downstream (upper : single, nether : 4 levels)



(a) Surface pressure coefficient

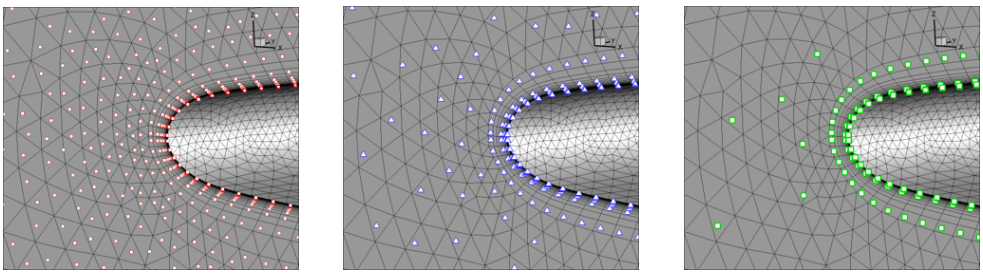
(b) Skin friction coefficient

Figure 4.65: surface coefficient comparison for Re 40 cylinder flow

4.5 Turbulent flows

4.5.1 Transonic flows around ONERA M6 (Re 11.75e6)

The results in this section are provided to investigate convergence properties of the multicloud method with directional coarsening. The transonic flows around ONERA M6(Reynolds number 11.75E6) was numerically solved by turbulent solver. S-A one equation model was used to estimate the viscosity. Furthermore, S-A equation was also included in the multicloud process. However, S-A equation was applied to three-level V cycle, whereas five basic transport equations (mass, momentum, energy) was applied to four-level V cycle. Furthermore, only one iteration was conducted for S-A equations on the coarse levels while five transport equations was performed by multiple iterations on the coarse levels. The flow grid used in this case is shown in Figure 4.66 with its coarsening result. Directional coarsening was applied since the maximum aspect ratio is approximately 500. As seen from Figure 4.66a-4.66c, points are coarsened along the marching direction of prism in order to alleviate high stretch of the grids. Furthermore, meshlss discretization work perfectly on the region where prisms encounter tetrahedral grids. Figure 4.67 and 4.68 show ρ and $\tilde{\nu}$ convergence histories. Significant acceleration effect is shown Both ρ and $\tilde{\nu}$ residuals. Although the results are slightly slower compared to the inviscid results, they are still significantly effective.



(a) level 2 (b) level 3 (c) level 4

Figure 4.66: Illustration of coarsening results for the highly stretched grid for ONERA M6

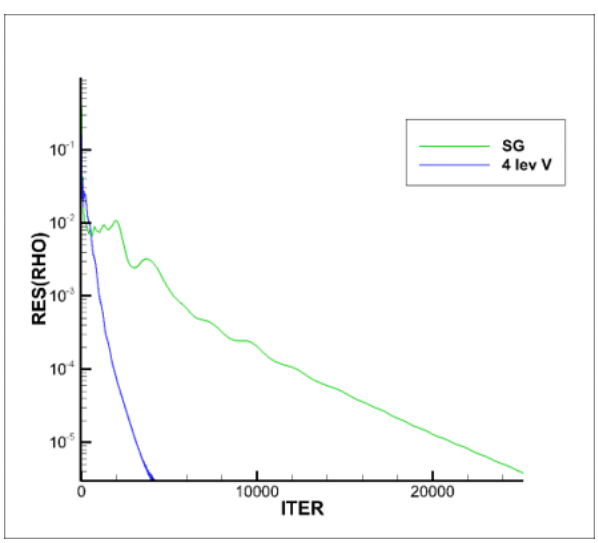


Figure 4.67: Density residual history of ONERA M6 (Re 11.75e6)

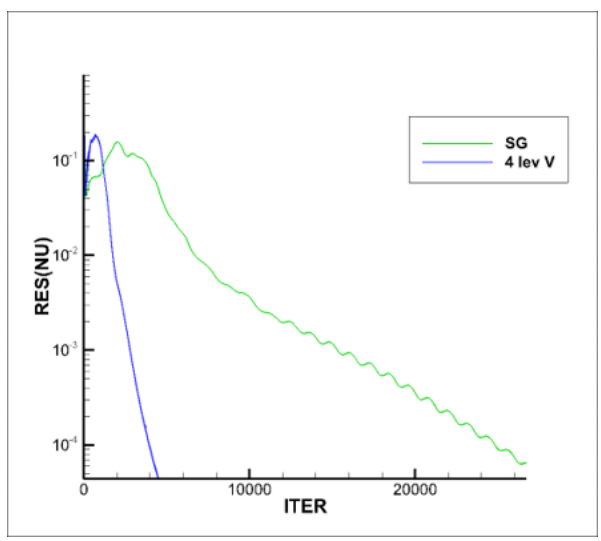
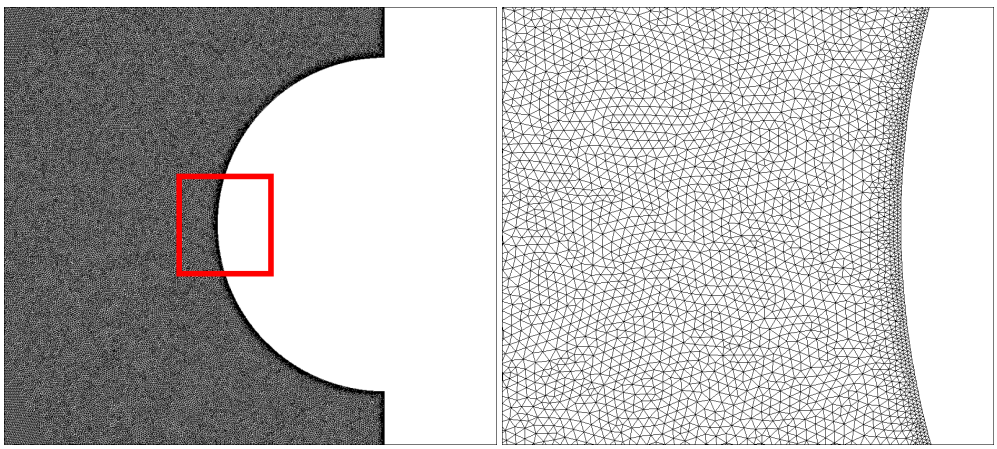


Figure 4.68: $\tilde{\nu}$ residual history of ONERA M6 (Re 11.75e6)



(a) (b)

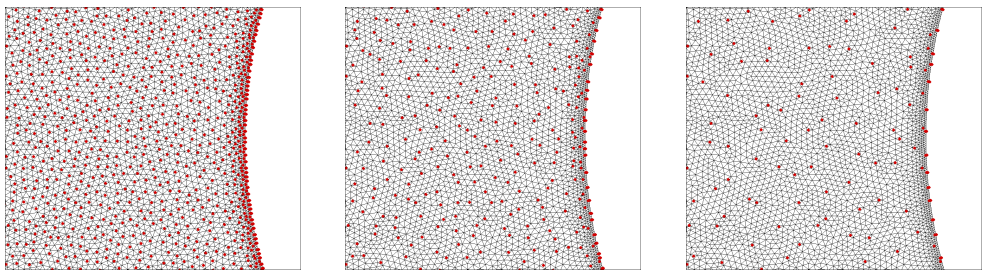
Figure 4.69: Unstructured triangle grids for cylinder

4.6 hypersonic flows

4.6.1 Mach 8 flow around the cylinder

In order to validate the performance of damped restriction and prolongation. hypersonic flows over cylinder was tested. the cylinder geometry is most widely used geometry for validation of schemes dealing with hypersonic flows. Figure 4.69 illustrates two dimensional unstructured grids that consist of triangles. Only unstructured triangle grids are tested in this study since quadrature grids is same as structured grids which were demonstrated in the early studies. Cell finite volume method is used for fine level discretization such that CNA coarsening is applied as shown in Figure 4.70. Inviscid analysis with Mach number 8 is tested for this case. Numerical fluxes are computed by AUSMPW+ and LU-SGS time integration with CFL 1. Both damped restriction and prolongation are applied to the multicloud process.

Figure 4.71 and 4.72 illustrate convergence histories. As seen in Figure 4.71,



(a) Level 2 point cloud (b) Level 3 point cloud (c) Level 4 point cloud

Figure 4.70: Coarse level point cloud generated by CNA coarsening

four-level multicloud method takes 11533 iterations to achieve the order of -8 which is 4 times faster than the single method (46600 iterations). In terms of wall time, the single method takes 0.040 seconds for an iteration whereas 0.068 seconds for the four-level V cycle multicloud method. Thus the speed-up with respect to walltime is the factor of 2.31. Problems that include strong shocks generally exhibit degraded acceleration. Furthermore the factor of 2.31 is a similar level of acceleration compared to the structured multigrid methods for hypersonic flows. Furthermore, the multicloud method demonstrated superior acceleration compared to the structured multigrid method proposed by Kim[6] even though unstructured multigrid methods often shows less acceleration than structured multigrid methods. As seen in Figure 4.73, solution discrepancy does not exhibit since L2 norm of the difference of rho is the order of 1E-10. Consequently, it is evident that the multicloud method for hypersonic flows delivers a satisfactory acceleration effect, regardless of the grid types.

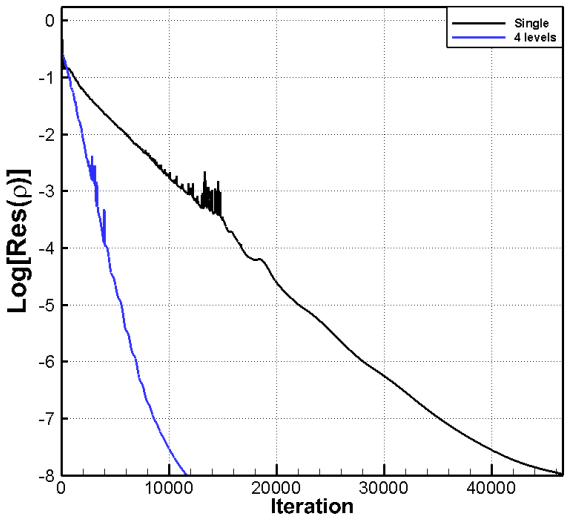


Figure 4.71: Density residual history for Mach 8 cylinder in terms of iteration

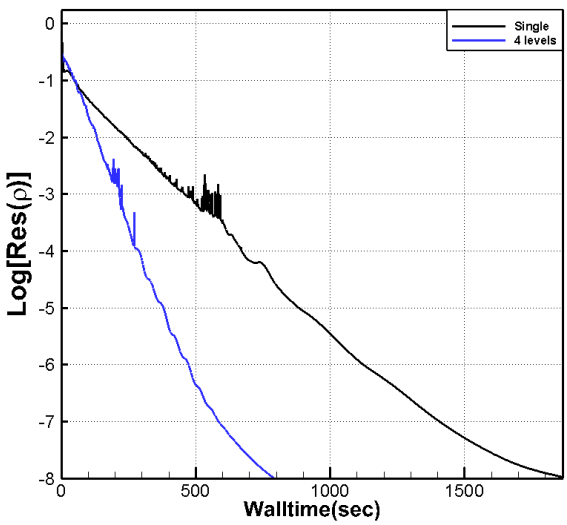
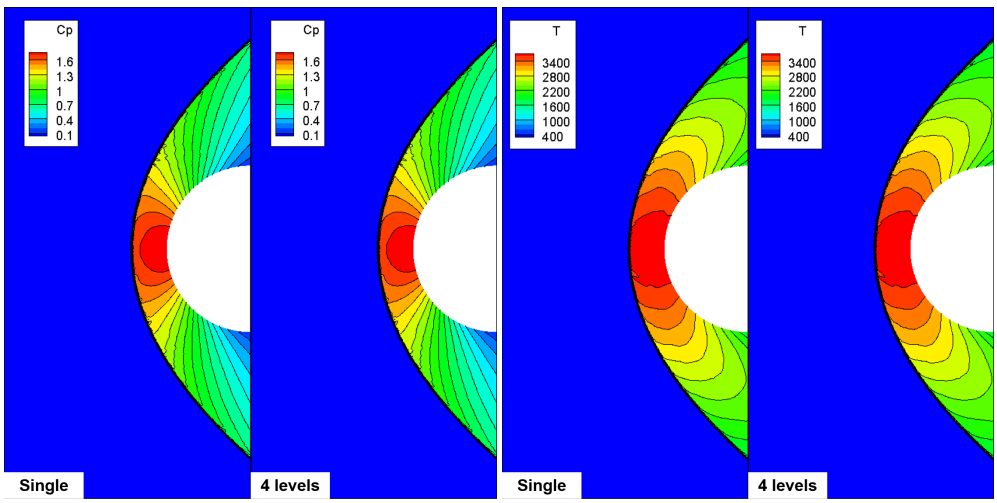


Figure 4.72: Density residual history for Mach 8 cylinder in terms of walltime



(a) Pressure contour (b) Temperature contour

Figure 4.73: Contour comparisons between the single and four-level grid

4.6.2 Mach 8 flows around the sphere

The mach 8 flows around the sphere was tested to verify the performance of the damping functions in three-dimensional space. Figure 4.74 illustrates the symmetric flow grid for the sphere. Only the quarter of the sphere is considered since supersonic inflow and outflow condition is only valid in this case. CFV spatial discretization was applied to the grid in Figure 4.74. AUSMPW+ and LU-SGS were applied with CFL condition 0.2. The four-level V cycle multicloud method with CNA coarsening was used. Figure 4.76 exhibits the convergence history with respect to walltime. The single-method takes 9375 seconds until convergence whereas that of the four-level multicloud is 3835, which is the factor of 2.44. The speed-up factor is not degraded compared to the two-dimension case and even more effective than that of the 2D structured multigrid method of Kim[6]. Pressure and temperature contours exhibit no solution discrepancy for this case(See Figure 4.77). The L2norm of the difference of

rho is approximately 1E-12.

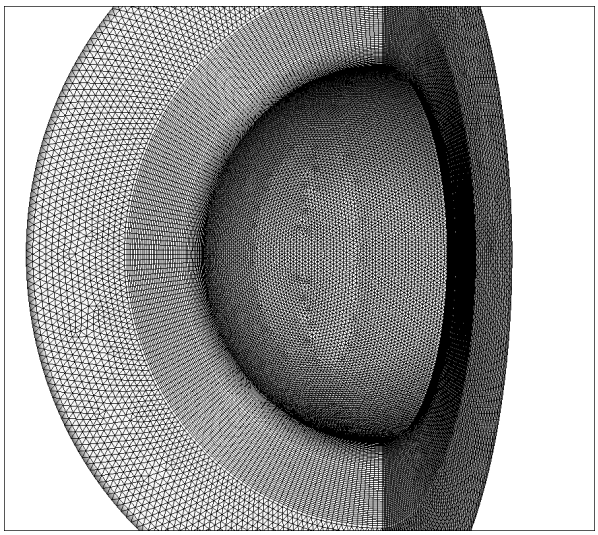


Figure 4.74: Symmetric flow grid for the sphere

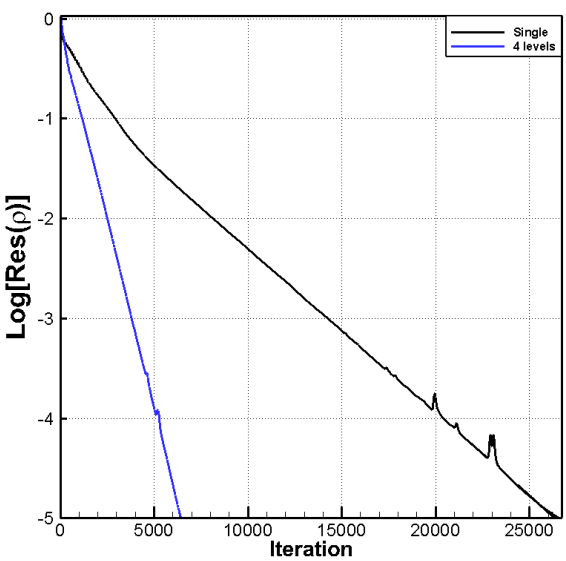


Figure 4.75: Density residual history for Mach 8 sphere in terms of iteration

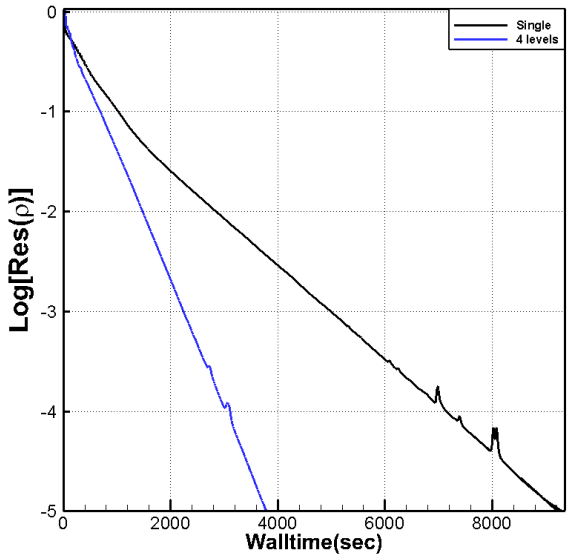
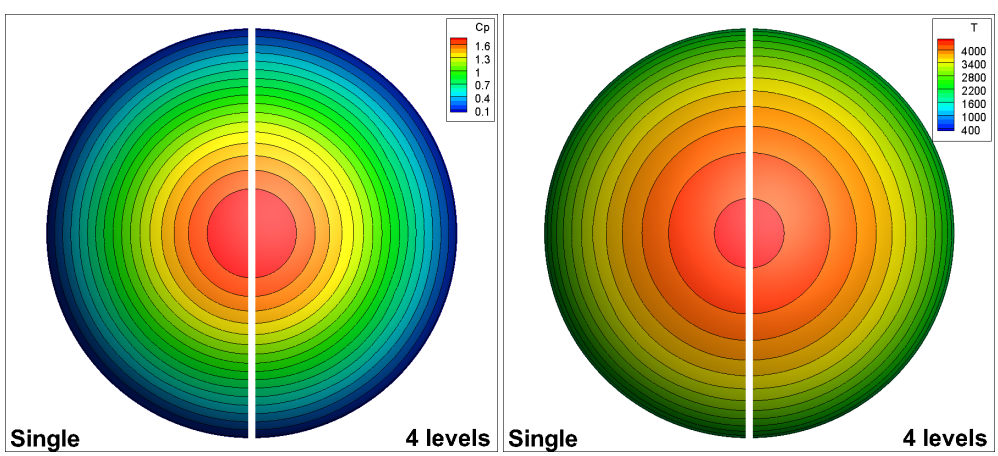


Figure 4.76: Density residual history for Mach 8 sphere in terms of walltime



(a) Pressure

(b) Temperature

Figure 4.77: Surface result comparisons for the Mach 8 sphere (left : single, right : four-level multicloud)

4.6.3 HTV-2

In order to demonstrate the performance of developed damping methods for a three-dimensional practical problem, The DARPA/AF Falcon Hypersonic Technology Vehicle-2 (HTV-2)[60] is selected. The Mach number conditions of HTV-2 ranged from Mach 6 to Mach 16, which are accompanied by strong shock. The hypersonic flow with Mach 16 and angle of attack $10°$[61] was tested, which is the maximum mach number among the test conditions. The prism and tetrahederal hybrid grids are used for the finest level. CNA coarsening was used until 4 level for four-level V cycle. CFV method was used with AUSMPW+ and LU-SGS. CFL condition was 0.5. Figure 4.79 and 4.80 illustrate convergence histories in term of iteration and wall time. The single grid method takes 7600 iterations to converges whereas the multicloud method takes only 2200 iterations. The speed-up in terms of iteration is the factor of 3. But in terms of walltime, the time spent per iteration is 0.55 seconds for the single grid method and 0.96 seconds for the multicloud method. Consequently, the speed-up in terms of walltime is the factor of 2 approximately. The speed-up is less effective than the previous results since strong shock intensively damps the restriction and prolongation. Nonetheless, The results indicate that the acceleration seems to be not only satisfactory but also robust for extremely high speed flows even for the three-dimensional practical problem. Figure 4.81 compares the results between the single-grid and multicloud methods at $x = 1.5$. As seen from the figure, both Mach number and pressure contours exhibit excellent agreement even for extremely strong shock.

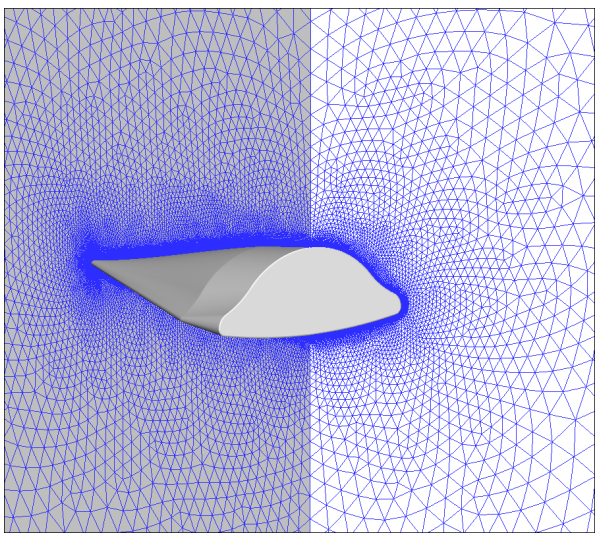


Figure 4.78: Symmetric flow grid for HTV-2

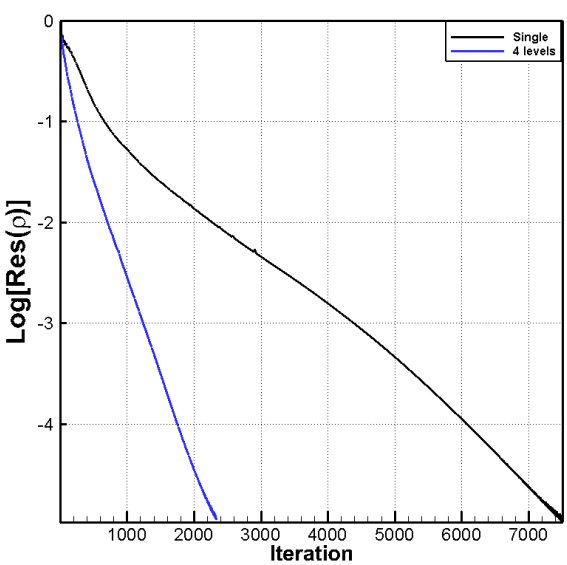


Figure 4.79: Density residual history for HTV-2 in terms of iteration

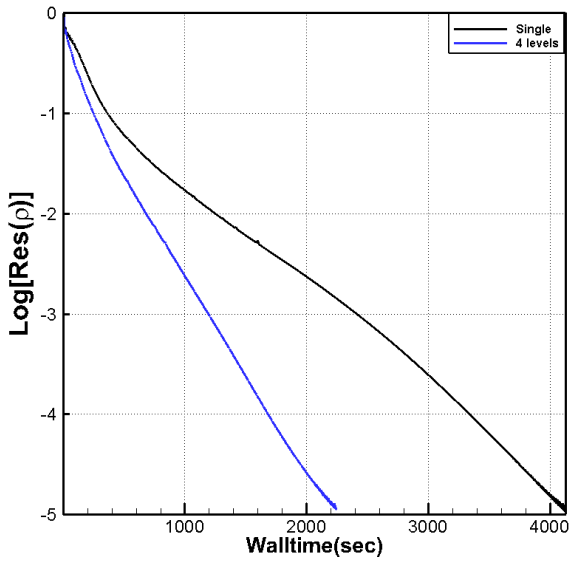
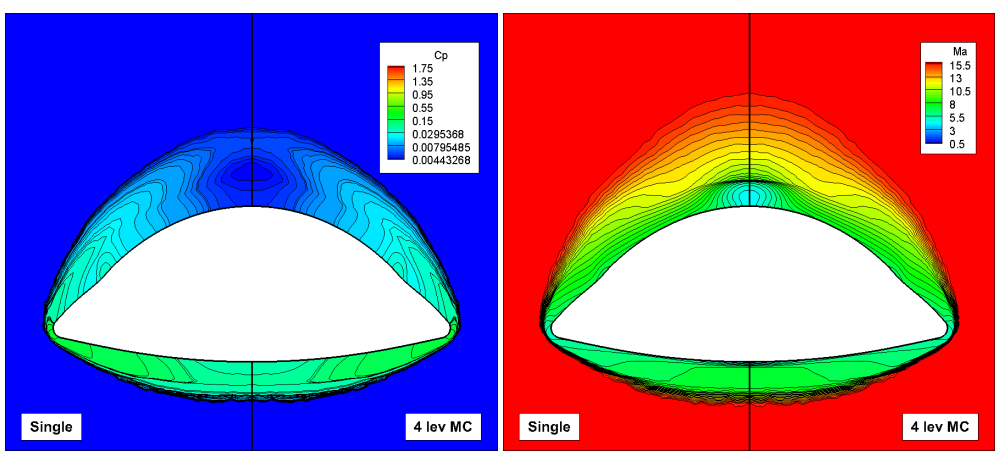


Figure 4.80: Density residual history for HTV-2 in terms of walltime



(a) Pressure (b) Mach number

Figure 4.81: Result comparisons for HTV-2

Chapter 5

Concluding remarks

5.1 Conclusions

The goal of the thesis is to develop a improved meshless geometric multigrid that is called as multicloud method. The improvements were achieved in a variety ways through the thesis.

The main achievement is attained by a new cell coarsening strategy, which is named as common node approach(CNA) coarsening. The new cell coarsening strategy provides extremely coarsened computational domains than the ideal coarsening rates. It was obvious that more coarsened domain grants the enhanced acceleration effect but it was uncertain that the meshless discretization can work robustly on the extremely coarsened domain. It is proved that the conventional least squares(LSM) meshless discretization exhibit instability on the CNA coarsening. However, meshless discretizaion by least squares method with geometric conservation law(GC-LSM) does not lose it robustness on CNA coarsening. Consequently, by combining GC-LSM and CNA coarsening, the extremely dramatic convergence acceleration effect

was obtained since GC-LSM grants the robust and accurate numerical estimation of fluxes even on the CNA coarsened domain.

Furthermore, lower-upper symmetric Gauss-Seidel (LU-SGS) method was applied to meshless methods regardless of its conservative properties. Considering the previous multicloud methods was used based on explicit time integration method, implementation of LU-SGS is significant progress for the multicloud method since it is evident that implicit time integration method guarantees much higher CFL number than that of explicit methods.

The blend of CNA coarsening, LU-SGS and GC-LSM have proven its performance for both inviscid and viscous flows. Results obtained from highly complicated configurations, such as the DLR-F6 wing body nacelle pylon model, clearly highlighted the significant effectiveness of the developed method.

Moreover, meshless damping methods were developed to stably accelerate convergence problems with strong shock. It was proved that the meshless damping methods is significantly effective through results of hypersonic flows regardless of dimension.

Finally, the results clearly demonstrate that the improved multicloud method cosnsitently outperforms the original multicloud method through a variety of proposed methodologies, regardless of the dimension, geometries, and flow conditions involved.

5.2 Future works

Even though the developed method provides successful meshless convergence accelerator, following studies are required for more applications. Most of all, the mutlcloud

application to other turbulence model is needed such as k-w sst[62]. The implementation may be more difficult than S-A one equation since two equations are added for k-w sst. Then, the multicloud application to dynamic mesh for unsteady problems is also highly recommended since meshless discretization can be useful tool for dynamic mesh. Furthermore, the adopation to overset grid can be a good application for the multicloud method.

Bibliography

[1] R. P. Fedorenko, "The speed of convergence of one iterative process," *USSR Computational Mathematics and Mathematical Physics*, vol. 4, no. 3, pp. 227–235, 1964.

[2] A. Brandt, "Multi-level adaptive solutions to boundary-value problems," *Mathematics of computation*, vol. 31, no. 138, pp. 333–390, 1977.

[3] R.-H. Ni, "A multiple-grid scheme for solving the euler equations," *AIAA journal*, vol. 20, no. 11, pp. 1565–1571, 1982.

[4] A. Jameson, "Multigrid algorithms for compressible flow calculations," in *Multigrid Methods II*. Springer, 1986, pp. 166–201.

[5] F. Liu and A. Jameson, "Multigrid navier-stokes calculations for three-dimensional cascades," *AIAA journal*, vol. 31, no. 10, pp. 1785–1791, 1993.

[6] S.-s. Kim, C. Kim, and O.-H. Rho, "Multigrid algorithm for computing hypersonic, chemically reacting flows," *Journal of spacecraft and rockets*, vol. 38, no. 6, pp. 865–874, 2001.

[7] Z. Zhu, P. Alavilli, C. Lacor, C. Hirsch, Z. Zhu, P. Alavilli, C. Lacor, and C. Hirsch, "Efficiency and robustness of multigrid methods for hypersonic flows," in *35th Aerospace Sciences Meeting and Exhibit*, 1997, p. 342.

[8] K. Stüben, "A review of algebraic multigrid," *Numerical Analysis: Historical Developments in the 20th Century*, pp. 331–359, 2001.

[9] J. Ruge and K. Stüben, *Efficient solution of finite difference and finite element equations by algebraic multigrid AMG*. Gesellschaft f. Mathematik u. Datenverarbeitung, 1984.

[10] U. Trottenberg, C. W. Oosterlee, and A. Schuller, *Multigrid*. Elsevier, 2000.

[11] E. Perez, "Finite element and multigrid solution of the two dimensional euler equations on a non structured mesh," Ph.D. dissertation, INRIA, 1985.

[12] D. J. Mavriplis, "Three-dimensional unstructured multigrid for the euler equations," *AIAA journal*, vol. 30, no. 7, pp. 1753–1761, 1992.

[13] D. Mavriplis, "Multigrid solution of the two-dimensional euler equations on unstructured triangular meshes," *AIAA journal*, vol. 26, no. 7, pp. 824–831, 1988.

[14] H. Guillard, "Node-nested multi-grid method with delaunay coarsening," Ph.D. dissertation, INRIA, 1993.

[15] M. F. Adams, "Parallel multigrid solvers for 3d unstructured finite element problems in large deformation elasticity and plasticity," *International Journal for Numerical Methods in Engineering*, vol. 48, no. 8, pp. 1241–1262, 2000.

[16] T. F. Chan and B. F. Smith, "Domain decomposition and multigrid algorithms for elliptic problems on unstructured meshes," *Contemporary Mathematics*, vol. 180, pp. 175–175, 1994.

[17] P. Crumpton and M. Giles, "Implicit time-accurate solutions on unstructured dynamic grids," *International journal for numerical methods in fluids*, vol. 25, no. 11, pp. 1285–1300, 1997.

[18] L. Botti, A. Colombo, and F. Bassi, "h-multigrid agglomeration based solution strategies for discontinuous galerkin discretizations of incompressible flow problems," *Journal of Computational Physics*, vol. 347, pp. 382–415, 2017.

[19] J. E. Jones and P. S. Vassilevski, "Amge based on element agglomeration," *SIAM Journal on Scientific Computing*, vol. 23, no. 1, pp. 109–133, 2001.

[20] V. Venkatakrishnan and D. J. Mavriplis, "Agglomeration multigrid for the three-dimensional euler equations," *AIAA journal*, vol. 33, no. 4, pp. 633–640, 1995.

[21] S. Langer, "Agglomeration multigrid methods with implicit runge–kutta smoothers applied to aerodynamic simulations on unstructured grids," *Journal of Computational Physics*, vol. 277, pp. 72–100, 2014.

[22] H. Nishikawa, B. Diskin, J. Thomas, and D. Hammond, "Recent advances in agglomerated multigrid," in *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2013, p. 863.

[23] S. Langer, A. Schwöppe, and N. Kroll, "Investigation and comparison of implicit smoothers applied in agglomeration multigrid," *AIAA Journal*, vol. 53, no. 8, pp. 2080–2096, 2015.

[24] G. N. Lygidakis, S. S. Sarakinos, and I. K. Nikolos, "Comparison of different agglomeration multigrid schemes for compressible and incompressible flow simulations," *Advances in Engineering Software*, vol. 101, pp. 77–97, 2016.

[25] S. Dargaville, A. G. Buchan, R. P. Smedley-Stevenson, P. N. Smith, and C. C. Pain, "A comparison of element agglomeration algorithms for unstructured geometric multigrid," *Journal of Computational and Applied Mathematics*, vol. 390, p. 113379, 2021.

[26] A. Katz and A. Jameson, "Multicloud: Multigrid convergence with a meshless operator," *Journal of Computational Physics*, vol. 228, no. 14, pp. 5237–5250, 2009.

[27] A. J. Katz, *Meshless methods for computational fluid dynamics*. Stanford University, 2009.

[28] J. J. Monaghan and R. A. Gingold, "Shock simulation by the particle method sph," *Journal of computational physics*, vol. 52, no. 2, pp. 374–389, 1983.

[29] R. Franke, "Scattered data interpolation: tests of some methods," *Mathematics of computation*, vol. 38, no. 157, pp. 181–200, 1982.

[30] E. J. Kansa, "Multiquadrics—a scattered data approximation scheme with applications to computational fluid-dynamics—i surface approximations and partial derivative estimates," *Computers & Mathematics with applications*, vol. 19, no. 8-9, pp. 127–145, 1990.

[31] D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data," in *Proceedings of the 1968 23rd ACM national conference*, 1968, pp. 517–524.

[32] D. Mavriplis, "Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes," in *16th AIAA computational fluid dynamics conference*, 2003, p. 3986.

[33] A. Ghosh and S. Deshpande, "Least squares kinetic upwind method for inviscid compressible flows," in *12th Computational Fluid Dynamics Conference*, 1995, p. 1735.

[34] D. Sridar and N. Balakrishnan, "An upwind finite difference scheme for meshless solvers," *Journal of Computational Physics*, vol. 189, no. 1, pp. 1–29, 2003.

[35] J. Y. Huh, J. S. Rhee, K. H. Kim, and S. Y. Jung, "New least squares method with geometric conservation law (gc-lsm) for compressible flow computation in meshless method," *Computers & Fluids*, vol. 172, pp. 122–146, 2018.

[36] R. Zamolo, E. Nobile, and B. Šarler, "Novel multilevel techniques for convergence acceleration in the solution of systems of equations arising from rbf-fd meshless discretizations," *Journal of Computational Physics*, vol. 392, pp. 311–334, 2019.

[37] A. Radhakrishnan, M. Xu, S. Shahane, and S. P. Vanka, "A non-nested multilevel method for meshless solution of the poisson equation in heat transfer and fluid flow," *arXiv preprint arXiv:2104.13758*, 2021.

[38] N. B. Barik and T. Sekhar, "Mesh-free multilevel iterative algorithm for navier–stokes equations," *Numerical Heat Transfer, Part B: Fundamentals*, vol. 79, no. 3, pp. 150–164, 2020.

[39] S. T. Ha and H. G. Choi, "A meshless geometric multigrid method based on a node-coarsening algorithm for the linear finite element discretization," *Computers & Mathematics with Applications*, vol. 96, pp. 31–43, 2021.

[40] M. Parsani, K. Van den Abeele, C. Lacor, and E. Turkel, "Implicit lu-sgs algorithm for high-order methods on unstructured grid with p-multigrid strategy for solving the steady navier–stokes equations," *Journal of Computational Physics*, vol. 229, no. 3, pp. 828–850, 2010.

[41] K. R. Laflin, S. M. Klausmeyer, T. Zickuhr, J. C. Vassberg, R. A. Wahls, J. H. Morrison, O. P. Brodersen, M. E. Rakowitz, E. N. Tinoco, and J.-L. Godard, "Data summary from second aiaa computational fluid dynamics drag prediction workshop," *Journal of Aircraft*, vol. 42, no. 5, pp. 1165–1178, 2005.

[42] C. Geuzaine and J.-F. Remacle, "Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities," *International journal for numerical methods in engineering*, vol. 79, no. 11, pp. 1309–1331, 2009.

[43] X. Su, S. Yamamoto, and K. Nakahashi, "Analysis of a meshless solver for high reynolds number flow," *International Journal for Numerical Methods in Fluids*, vol. 72, no. 5, pp. 505–527, 2013.

[44] R. Radespiel, "Progress with multigrid schemes for hypersonic flow problems," *Journal of Computational Physics*, vol. 116, no. 1, pp. 103–122, 1995.

[45] P. Gerlinger, H. Möbus, and D. Brüggemann, "An implicit multigrid method for turbulent combustion," *Journal of Computational Physics*, vol. 167, no. 2, pp. 247–276, 2001.

[46] P. Spalart and S. Allmaras, "A one-equation turbulence model for aerodynamic flows," in *30th aerospace sciences meeting and exhibit*, 1992, p. 439.

[47] S. R. Allmaras and F. T. Johnson, "Modifications and clarifications for the implementation of the spalart-allmaras turbulence model," in *Seventh international conference on computational fluid dynamics (ICCFD7)*, vol. 1902. Big Island, HI, 2012.

[48] K. H. Kim, C. Kim, and O.-H. Rho, "Methods for the accurate computations of hypersonic flows: I. ausmpw+ scheme," *Journal of computational physics*, vol. 174, no. 1, pp. 38–80, 2001.

[49] F. Kemm, "A comparative study of tvd-limiters—well-known limiters and an introduction of new ones," *International Journal for Numerical Methods in Fluids*, vol. 67, no. 4, pp. 404–440, 2011.

[50] M. Darwish and F. Moukalled, "Tvd schemes for unstructured grids," *International Journal of heat and mass transfer*, vol. 46, no. 4, pp. 599–611, 2003.

[51] A. Jameson, "Analysis and design of numerical schemes for gas dynamics, 1: artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence," *International Journal of Computational Fluid Dynamics*, vol. 4, no. 3-4, pp. 171–218, 1995.

[52] C. Liu and C. Hu, "An immersed boundary solver for inviscid compressible flows," *International Journal for Numerical Methods in Fluids*, vol. 85, no. 11, pp. 619–640, 2017.

[53] V. Schmitt, "Pressure distributions on the onera m6-wing at transonic mach numbers, experimental data base for computer program assessment," *AGARD AR-138*, 1979.

[54] J. T. Batina, "Accuracy of an unstructured-grid upwind-euler algorithm for the onera m6 wing," *Journal of aircraft*, vol. 28, no. 6, pp. 397–402, 1991.

[55] C. L. Rumsey, S. M. Rivers, and J. H. Morrison, "Study of cfd variation on transport configurations from the second drag-prediction workshop," *Computers & fluids*, vol. 34, no. 7, pp. 785–816, 2005.

[56] H. McMorris and Y. Kallinderis, "Octree-advancing front method for generation of unstructured surface and volume meshes," *AIAA journal*, vol. 35, no. 6, pp. 976–984, 1997.

[57] S. H. Kim, *Development of Point Generation Technique Using Octrees for Numerical Simulation of Steady/Unsteady Flows*. Seoul National University, 2018.

[58] R. C. Swanson and S. Langer, "Comparison of naca 0012 laminar flow solutions: structured and unstructured grid methods," Tech. Rep., 2016.

[59] M. Schäfer, S. Turek, F. Durst, E. Krause, and R. Rannacher, *Benchmark computations of laminar flow around a cylinder*. Springer, 1996.

[60] S. Walker, J. Sherk, D. Shell, R. Schena, J. Bergmann, and J. Gladbach, "The darpa/af falcon program: The hypersonic technology vehicle# 2 (htv-2) flight demonstration phase," in *15th AIAA International Space Planes and Hypersonic Systems and Technologies Conference*, 2008, p. 2539.

[61] J. Shi, L. Zhang, B. Jiang, and B. Ai, "Aerodynamic force and heating optimization of htv-2 typed vehicle," in *21st AIAA International Space Planes and Hypersonics Technologies Conference*, 2017, p. 2374.

[62] F. R. Menter, "Two-equation eddy-viscosity turbulence models for engineering applications," *AIAA journal*, vol. 32, no. 8, pp. 1598–1605, 1994.

초 록

본 논문은 무격자 차분법을 이용하여 격자 비종속적인 수렴 가속 기법에 개발에 대하여 설명하고 있다. 격자 비종속적 수렴 가속 기법은 multicloud 기법이라 불리우며, 저해상도에서 무격자 차분을 이용하여 계산을 수행하여 고주파수의 에러를 감쇄시키고 그 결과로 고해상도 격자의 유동값을 보정하여 수렴을 가속시키는 방법이다. Multicloud 기법은 저해상도에서 무격자 차분을 사용함으로써, 기존의 비정렬 격자 기반의 멀티그리드 기법보다 손쉽게 저해상도의 computational domain을 만들 수 있고, 저해상도에서 다른 기법보다 강건하게 차분을 수행할 수 있는 장점이 있다. 이 장점들로 인해 고해상도에서 셀 중심, 노드 중심 그리고 무격자 기법과 같은 어떠한 방법의 차분법에도 적용될 수 있는 기법이다.

본 연구에서는 새로운 무격자 기반의 cell coarsening 기법을 개발하여 사용하여, 기존의 정렬 및 비정렬 멀티그리드 기법 대비 더 높은 coarsening rates의 저해상도 질점계의 자동적인 생성에 성공하였다. 하지만 높은 coarsening rates는 에러 감쇄 효과를 증폭시키지만 저해상도에서 강건한 계산이 동반될 때에 효과를 볼 수 있다. 이를 위해, 기하학적 보존을 만족하는 최소제곱법을 이용한 무격자 공간 차분 기법을 도입하여 다른 무격자 기법으로 성공하지 못한 수렴 가속 수준에 도달하는데 성공하였다.

뿐만 아니라, 기존의 multicloud 기법에서는 제안되지 않았던, 내재적인 LU-SGS 시간 적분법을 multicloud에 적용하는데 성공하여 기존에 사용되던 외재적

multicloud에 비해서 높은 수준의 수렴 가속효과를 강건하게 얻을 수 있게 되었다.

마지막으로는 무격자 기반의 multicloud 감쇄 함수를 개발하여, 극초음속 유동에서의 multicloud 기법의 효용성을 보여주었을 뿐만 아니라 새로운 cell coarsening 기법과 결합하여 기존의 정렬격자 멀티그리드 기법에 비하여 더 높은 수렴 가속 효과를 보여주었다.

결론적으로, 본 연구를 통해 개선된 multicloud 기법은 기존의 기법에서 보여주지 못했던, 셀 중심 기법에서의 성능 극대화에 성공하였고, 복잡한 삼차원 형상을 포함한 점성, 비점성 그리고 극초음속 유동에서의 효과를 증명하였다.

주요어: 무격자 기법, 수렴 가속, 비정렬 격자, 내재적 시간 적분, 멀티그리드

학번: 2016-30190