AFRL-AFOSR-UK-TR-2012-0027

Development of High-Order Method for Multi-Physics Problems Governed by Hyperbolic Equations

Dr. John A. Ekaterinarius

Foundation for Research and Techno
N. Plastira 100, Vassilika Vouton
Heraklion, Greece 70013

EOARD Grant 11-3070

Report Date: August 2012

Final Report from 20 July 2011 to 19 July 2012

Air Force Research Laboratory
Air Force Office of Scientific Research
European Office of Aerospace Research and Development
Unit 4515 Box 14, APO AE 09421

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From – To)*
1 August 2012	Final Report	20 July 2011 – 19 July 2012

4. TITLE AND SUBTITLE

Development of High-Order Method for Multi-Physics Problems Governed by Hyperbolic Equations

5a. CONTRACT NUMBER

FA8655-11-1-3070

5b. GRANT NUMBER

Grant 11-3070

5c. PROGRAM ELEMENT NUMBER

61102F

6. AUTHOR(S)

Dr. John A. Ekaterinarius

5d. PROJECT NUMBER

5d. TASK NUMBER

5e. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Foundation for Research and Techno
N. Plastira 100, Vassilika Vouton
Heraklion, Greece 70013

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

EOARD
Unit 4515 BOX 14
APO AE 09421

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/AFOSR/RSW (EOARD)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

AFRL-AFOSR-UK-TR-2012-0027

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited. (approval given by local Public Affairs Office)

13. SUPPLEMENTARY NOTES

14. ABSTRACT

In this section we present the discontinuous Galerkin (DG) discretization of the three dimensional Euler and Nervier-Stokes equations for hybrid-type meshes. Without loss of generality the general finite element discretization framework is presented for hexahedral type meshes since all computations of the DG method are performed at the computational domain on the standard cubic element and transferred back to the physical domain elements (tetrahedras, prisms, pyramids, or hexahedras) using collapsed coordinate transformations. This approach greatly facilitates implementation of hybrid meshes where neighboring element communication is performed through the numerical flux defined on the element faces. The numerical solution has been validated for flow over a cylinder and for flow over a wing with Joukowsky airfoil section.

15. SUBJECT TERMS

EOARD, Aerodynamics, CFD

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18, NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Gregg Abate
UNCLAS	UNCLAS	UNCLAS	SAR	41	19b. TELEPHONE NUMBER *(Include area code)* +44 (0)1895 616021

Progress Report for the EOARD Project

Development of High–Order Methods for Multi–Physics Problems Governed by Hyperbolic Equations

by

John A. Ekaterinaris

FORTH / IACM

Award No. FA 8655–11-1-3070

August 1, 2012

1. DG discetization of the 3D Euler equations on hybrid meshes

In this section we present the discontinuous Galerkin (DG) discretization of the three dimensional Euler and Nervier-Stokes equations for hybrid-type meshes. Without loss of generality the general finite element discretization framework is presented for hexahedral type meshes since all computations of the DG method are performed at the computational domain on the standard cubic element and transferred back to the physical domain elements (tetrahedras, prisms, pyramids, or hexahedras) using collapsed coordinate transformations. This approach greatly facilitates implementation of hybrid meshes where neighboring element communication is performed through the numerical flux defined on the element faces. The numerical solution has been validated for flow over a cylinder and for flow over a wing with Joukowsky airfoil section.

1.1 INTRODUCTION

Recent developments of high order numerical methods for unstructured meshes appeared in the past few years offer significant advantages for the simulation of complex flows and turbulence in non trivial geometries of interest to practical applications. The discontinuous Galerkin (DG) [1 – 2], the spectral volume (SV) [3], and the spectral difference [4 – 5] methods have shown promise for high resolution computation of complex flows because they have a compact stencil, and retain the design order of accuracy even for meshes of moderate quality that often result from grid generation over complex configurations. The potential application of these methods for high resolution simulations of practical flow problems can be further enhanced with the use of mixed–type meshes and solution adaptive schemes. Mixed–type meshes facilitate the resolution of near wall regions, which can be represented with regular structured-type meshes (quadrilateral in two dimensions or tetrahedral/prismatic meshes in three dimensions), and allow extension to the far field with sparser triangular or tetrahedral type meshes. Furthermore, when the near wall flow is discretized with a structured-like hexahedral mesh, application of implicit time marching methods is more straightforward, because methods such as the LU-SGS scheme [6], can be used as preconditioners of the large linear system of equations that results for implicit time marching of the DG method.

Solution adaptive refinement strategies of *h*–, *p*–, or *h/p*–type can reduce the computing time for high resolution simulations of complex flows. However, application of *h*–type adaptive refinement strategies, even if the DG method allows use of hanging nodes, is not straightforward for complex flows, because it requires re-meshing and becomes quite difficult for time dependent flows with moving flow features, as it is necessary to remove cells from regions where shear layers or vortices have passed. In contrast, application of *p*– type adaptive strategies is straightforward for DG methods when hierarchical base functions are used as expansion polynomials. In addition, use of *p*– type adaptively is straightforward for time dependent flows, such as vortical flows for

example, because the order of accuracy can be selectively increased in regions of high vorticity gradients. Application of p-type adaptivity with the DG method will be demonstrated in a subsequent section. To further enhance accuracy curved boundaries or the simple procedure of Ref. [7] may be implemented.

In this work the DG finite element method is applied using the *h/p* finite element framework of Ref. [8]. Modal, hierarchical bases are constructed in the computational space over a unit square/cube using tensor products and transformed to the physical space of a quadrilateral/ hexahedra or triangle/tetrahedra using a collapsed coordinate system. The modal bases offer an advantage for mixed element implementation compared with nodal bases used in Ref. [9 – 10]. Numerical evaluations of line and volume integrals, required in the implementation of the DG method, are carried out in the computational domain. Use of the resulting methodology for mixed-type unstructured meshes is straightforward. Furthermore adaptivity of the solution based on computed flow features is possible. In the computational examples, simple *p*– type adaptive schemes based on the gradient of the computed density field are demonstrated. In the region of high gradients the order of the polynomial expansion is raised to a preset desired level and then progressively drops to the lowest order of accuracy that is used for the computation of the free stream where no gradients exist.

The main problem with the application of the DG method for high resolution computation of compressible flow with discontinuities in mixed type meshes is lack of a unified limiting approach. Furthermore, limiting approaches for DG discretizations presented in the literature [11 – 12] do not have a straight forward extension for three dimensional flows. Our new unified limiting approach developed recently for two dimensional flows is suitable for mixed type meshes and application of *p*-type refinement, it can be extended to three dimensions, and it smears discontinuities over a small range of cells because it allows more accurate detection of sharp variations of the numerical solution. Application of limiters is not necessary for the computation of dynamic stall. However, for the sake of generality our new limiting approach, which is applicable because all calculations in the DG formulation are performed in the computational domain for the Euler and NS equations.

1.2 GOVERNING EQUATIONS

The motion of a compressible fluid without any viscous and thermal conduction effects, is described by the system of Euler equations:

$$\partial_t \mathbf{q} + \partial_x \mathbf{f}(\mathbf{q}) + \partial_y \mathbf{g}(\mathbf{q}) + \partial_z \mathbf{h}(\mathbf{q}) = \partial_x \mathbf{f}^v(\mathbf{q}, \nabla\mathbf{q}) + \partial_y \mathbf{g}^v(\mathbf{q}, \nabla\mathbf{q}) + \partial_z \mathbf{h}^v(\mathbf{q}, \nabla\mathbf{q}), \qquad (1)$$

The conservative variable vector is $\mathbf{q} = \{\rho,\ \rho\mathbf{v}, E\}^T$ and the components of the inviscid flux vector are $\mathbf{f}(\mathbf{q}) = \{\rho u,\ \rho u^2 + p,\ \rho u\upsilon,\ \rho uw,\ (E+p)u\}^T$, where, p is the pressure, E is the total energy, and $\mathbf{v} = (u, \upsilon, w)$ are the velocity components. Discretization of the Euler and NS, written in short as conservation law $\mathbf{q}_t + \nabla \cdot \mathbf{F} = 0$, with the DG method is shown next.

1.2.1 DG IMPLEMENTATION

A tessellation of an arbitrary domain Ω, of the flow field, into elements Ω_e is considered $\Omega = \cup_e \Omega_e$, and defines the computational mesh. The tessellation may be composed of either straight sided or curvilinear general shaped elements.

The physical field is approximated using a polynomial expansion $\tilde{\mathbf{q}}_e(\mathbf{x},t)$ over each element, resulting in a discontinuous approximation across the inter element boundaries. Substitution of the $\tilde{\mathbf{q}}_e(\mathbf{x},t)$ in Eq. (1) results in an error of approximation, or residual, of the flow field over the elemental region.

The discontinuous Galerkin method minimizes the approximation error in the computational domain in the weighted residual sense and permits to write the residual for every element as:

$$\int_{\Omega_e} W_e \frac{\partial \tilde{\mathbf{q}}_e}{\partial t} d\Omega_e - \int_{\Omega_e} \nabla W_e \cdot F(\tilde{\mathbf{q}}_e) d\Omega_e + \oint_{\partial\Omega_e} W_e F(\tilde{\mathbf{q}}_e) \cdot \mathbf{n}_e dS = 0 \quad (3)$$

where W_e denotes the weight function, F the flux vector and $\mathbf{n}_e$ the outward normal vector at the element boundary, and $F(\tilde{\mathbf{q}}_e) \cdot \mathbf{n}_e$ is replaced by the numerical flux $\hat{H}(\tilde{\mathbf{q}}_e)$. Any consistent numerical flux is suitable, such as the Roe flux, the HLL flux, or the Osher flux. The most computationally efficient flux is the local Lax – Friedrichs (LF) flux, leading to the following definition of the numerical flux:

$$\hat{H}(\tilde{\mathbf{q}}_e) = \frac{1}{2}\left[F(\tilde{\mathbf{q}}_e^-) + F(\tilde{\mathbf{q}}_e^+) \right] \cdot \mathbf{n}_e - \frac{1}{2}|k|\left(\tilde{\mathbf{q}}_e^+ - \tilde{\mathbf{q}}_e^- \right) \quad (4)$$

where $\tilde{\mathbf{q}}_e^-$ and $\tilde{\mathbf{q}}_e^+$, denotes the solution at the interior and the exterior of the element respectively, and k the spectral radius of the flux Jacobian for the system of the Euler equations. Alternatively, the Roe flux is obtained when the spectral radius k is replaced by $R|\Lambda|L$, where R and L are the right and left eigenvector matrices and Λ is the eigenvalue matrix of the flux Jacobian.

The local approximation of the solution is constructed by a linear combination of elemental basis functions $b_k^e(\mathbf{x})$:

$$\tilde{\mathbf{q}}_e(\mathbf{x},t) = \sum_k \hat{\mathbf{q}}_e^k(t)\ b_k^e(\mathbf{x}), \quad (4)$$

where the coefficient vector $\hat{q}_e^k(t)$ denotes the degrees of freedom (DOF) of the numerical solution in every element to be advanced in time. Substituting Eq. (4) in Eq. (3) obtain:

$$\int_{\Omega_e} W_e \, b_k^e(\mathbf{x}) \frac{\partial \hat{\mathbf{q}}_e^k(t)}{\partial t} d\Omega_e - \int_{\Omega_e} \nabla W_e \cdot F(\tilde{\mathbf{q}}_e) d\Omega_e + \oint_{\partial\Omega_e} W_e \, \widehat{H} dS = 0 \qquad (5)$$

As a Galerkin procedure implies the weighting function W_e is the same with the elemental expansion function $b_k^e(\mathbf{x})$, resulting in a system of $\{(d+2)\times DOF\}\times\{(d+2)\times DOF\}$ equations for the system of Euler equations, where d is the dimension of the problem.

Gauss numerical integration is employed for the computation of the volume and the line integral appearing in Eq. (5). The mixed type element meshes do not pose any problem to the computation of the volume integral over quadrilaterals or triangles, as the computation is practically the same by storing at every quadrature point the value of the basis functions in the computational space (both for quadrilaterals and triangles) and the value of the Jacobian in every element. Moreover, by looping over the edges it is feasible to compute the line integral for every element no matter of the type of its neighbouring elements. Time marching of the solution is performed with an explicit or implicit Runge-Kutta method.

1.3 DG discretization of the 3D Nervier-Stokes equations

The compressible Navier-Stokes equation in conservation law form are:

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}_{\text{inv}}(\mathbf{U}) + \nabla \cdot \mathbf{F}_{\text{vis}}(\mathbf{U}, \nabla \mathbf{U}) = 0$$

where for example

$$\mathbf{F}_{\text{vis, x}}(\mathbf{U}, \nabla \mathbf{U}) = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} - \kappa T_{,x} \end{bmatrix}, \quad \tau_{xx} = \frac{4}{3}\mu\frac{\partial u}{\partial x} - \frac{2}{3}\mu\left(\frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right)$$

where the viscous terms were written as function of the state variable $\mathbf{U}$ and the gradient $\nabla\mathbf{U}$ of the state variables $\nabla \cdot \mathbf{F}_{\text{vis}}(\mathbf{U}, \nabla \mathbf{U})$ as required for the DG implementation.

The DG method is applied for weak form of the governing equations using the LDG approach for the system of equations

$$\nabla \mathbf{U} = G(\mathbf{q}_1,\ \mathbf{q}_2,\ \mathbf{q}_3,\ \mathbf{q}_4)$$
$$\nabla u = \mathbf{q}_1,\ \nabla v = \mathbf{q}_2, \nabla w = \mathbf{q}_3,\ \nabla T = \mathbf{q}_4$$
$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}_{\text{inv}}(\mathbf{U}) + \nabla \cdot \mathbf{F}_{\text{vis}}(\mathbf{U}, \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4) = 0$$

The weak for is

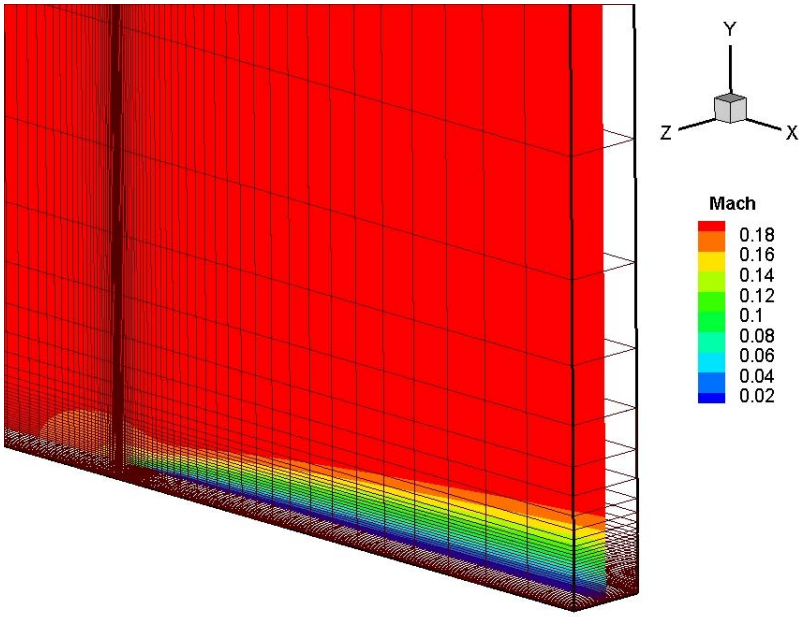
$$\int_\Omega v \frac{\partial \mathbf{U}}{\partial t} d\Omega - \int_\Omega \nabla v \cdot \mathbf{F}_{\text{inv}}(\mathbf{U}) d\Omega + \oint_{\partial\Omega} v \mathbf{F}_{\text{inv}}(\mathbf{U}) \cdot \mathbf{n} ds$$
$$\int_\Omega \nabla v \cdot \mathbf{F}_{\text{vis}}(\mathbf{U}, \nabla \mathbf{U}) d\Omega + \oint_{\partial\Omega} v \mathbf{F}_{\text{vis}}(\mathbf{U}, \nabla \mathbf{U}) \cdot \mathbf{n} ds = 0$$

and each of the gradients is computed using DG discretization. For example the weak form for a generic gradient term evaluation is:

$$\int_\Omega v \mathbf{q}_h d\Omega - \oint_{\partial\Omega} v \hat{u}\ ds + \int_\Omega \nabla v u d\Omega = 0$$

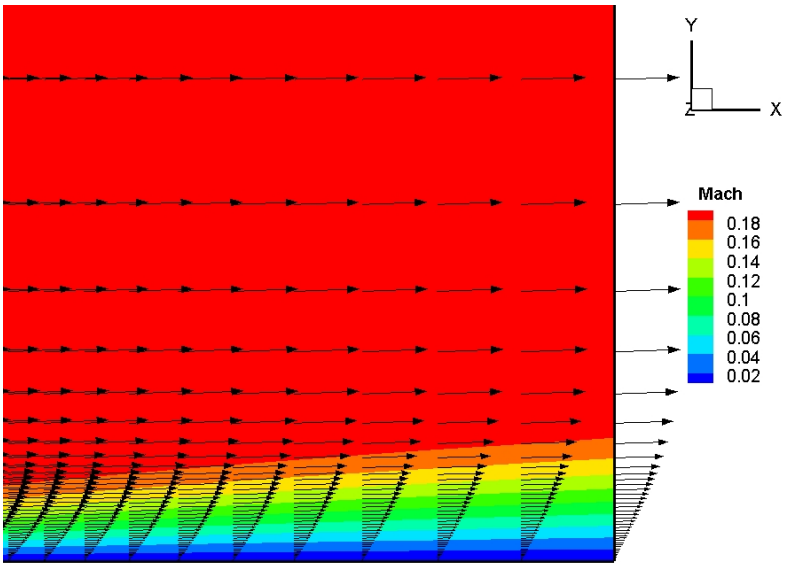
where each of the gradients is computed with the LDG approach. Preliminary results from viscous flow computations are shown bellow.

The computation of the low speed (incompressible) boundary layer flow with zero pressure gradient was considered as validation case. A three dimensional hexahedral element mesh was used for this computation.

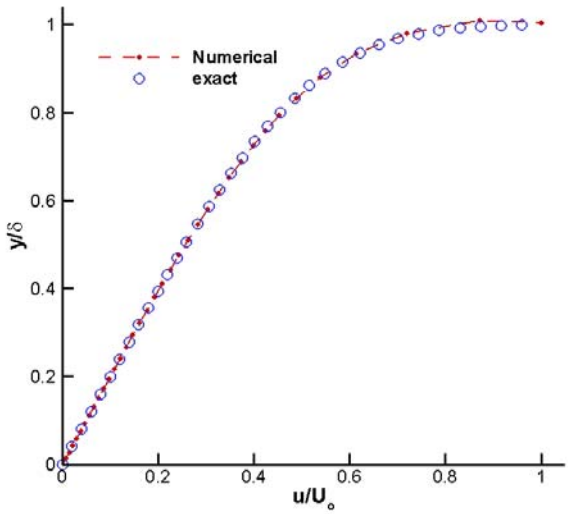


Computed Mach number distribution for low speed boundary layer

The computed velocity profiles are shown for a two dimensional slice at Re = 500 are shown bellow



Computed velocity profiles at Re = 500



Comparison of the P1 computed solution with the exact Blasius solution for the zero pressure gradient laminar boundary layer.

1.4 RESULTS

Additional results from the DG method for the Euler equations are presented for subsonic and supersonic flows using hexahedral and tetrahedral type meshes. The next step is careful validation of the computed results and verification that the design order-of-

convergence is achieved, for example P3 polynomial expansions yield a 4th order accurate solution. Towards this end the code will be parallelized using domain decomposition techniques and MPI so that fine mesh solutions become possible in reasonable computational time. The parallelization approach, which is the second task of the time table, will be presented in the next report.

The flow over a Joukowsky airfoil is computed. The airfoil shape in the $z = x + iy$ plane is obtained through the map $z = \zeta + 1/\zeta$ of the circle in the $\zeta = \xi + i\eta$ plane with center at $(a_x,\, a_y) = (-0.08, 0.08)$ and radius $R = \sqrt{(1-a_x)^2 + a_y^2}$. The computed flow with P1 polynomial bases (second order accuracy) for an infinite in the spanwise direction wing with Joukowsky airfoil section at incidence $\alpha = 0$ deg. and for $M_\infty = 0.5$ is shown in Fig. 1.1. A hexahedral type mesh was used and one layer of cells in the spanwise direction was sufficient and high speed was selected to obtain faster convergence to a steady state. The computed surface pressure coefficient distribution is compared with the analytic result in Fig. 1.2. There are discrepancies only at the leading edge where compressibility effects are more significant. The flow was computed with a tetrahedral type mesh. For this computation two layers of tetrahedras are needed and the mesh is shown in Fig. 1.3. The computed flow field at incidence $\alpha = 0$ deg. and for $M_\infty = 0.3$ is shown in Fig. 1.4. Other type of elements, such as triangular prisms and pyramids can be used for the computations. However, currently the option of using hybrid type meshes is implemented in the code and results for hybrid meshes will be shown in the next report. Hybrid meshes for wing flows could be hexahedra/prismatic (where the wind surface is discretized with a quadrilateral mesh) or prismatic/tetrahedral (where the wind surface is discretized with a triangular mesh). For the hexahedra/prismatic the far field is represented with triangular prisms aligned along the spanwise direction and for the prismatic/tetrahedral mesh the far field is represented with tetrahedras.

The computed flow field at supersonic speed $M_\infty = 1.0$ and incidence $\alpha = 0$ deg. is shown in Fig. 1.5. This computation was obtained for a piecewise constant P0 expansion basis (1st order accurate solution) because slope limiters have not incorporated as yet to allow 2nd order accurate numerical solutions with discontinuities. Recently, a new limiting approach was developed and thoroughly validated for mixed-type of meshes. Our new limiting approach is straightforward to extend into three dimensions, in contrast to other approach presented so far in the literature, and will be implemented in the 3D code. Results from the 3D application of the limiters will be presented in the next report.

The DG method for the three dimensional Euler and NS equations was implemented in a generalized framework that allows implementation of mixed-type meshes. Furthermore, since all calculations are performed in the canonical cubic element of the computational space (where hierarchical, tensor product expansion bases can be obtained) application of a *p*-adaptive procedure is possible. Preliminary results have been presented for simple wing flows. Thorough validation will be performed once the code is parallelized and fine mesh computations can be obtained at a reasonable computing time.

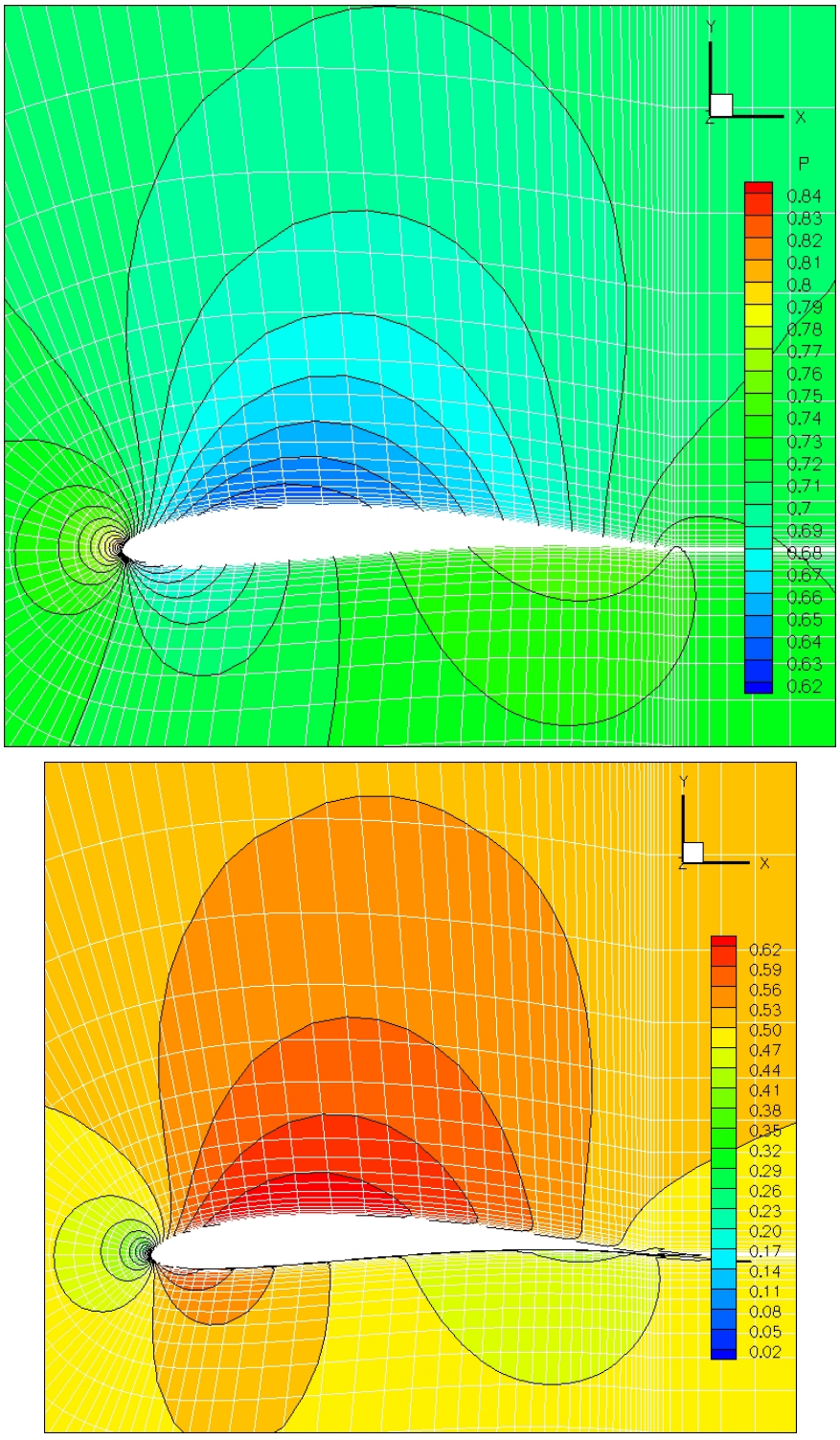


Fig. 1.1 Computed pressure and velocity magnitude fields for subsonic flow over a wing with Joukowsky airfoil sections using hexahedra mesh; $\alpha = 0$ deg., $M_\infty = 0.5$

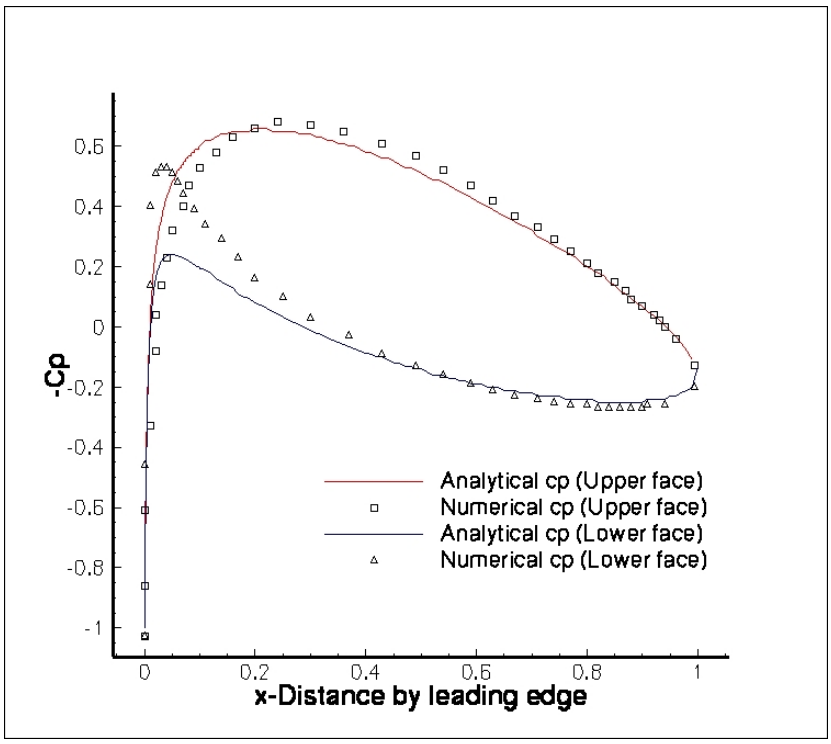


Fig. 1.2 Comparison of the computed ($\alpha = 0$ deg., $M_\infty = 0.5$) surface pressure coefficient with the analytic solution ($\alpha = 0$ deg., incompressible).

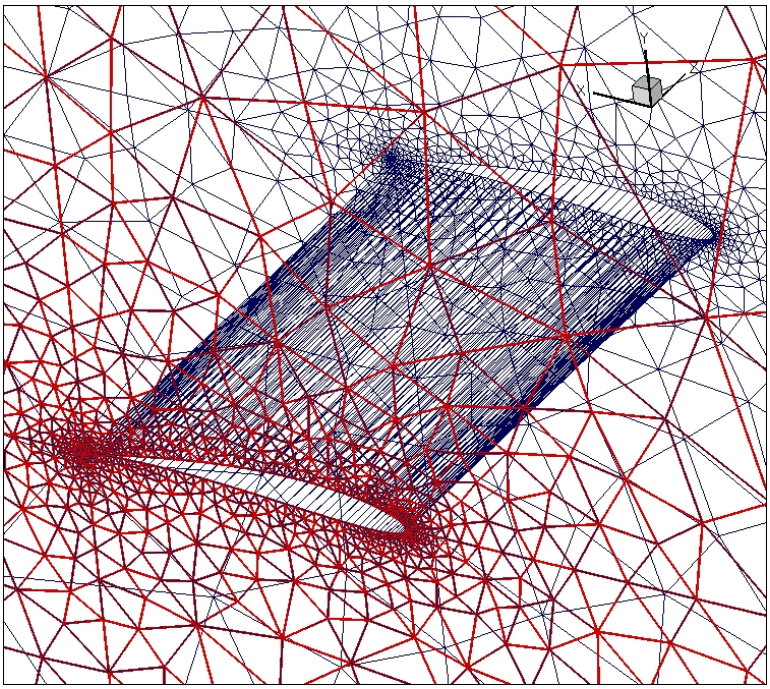


Fig. 1.3 Tetrahedral mesh for the wing.

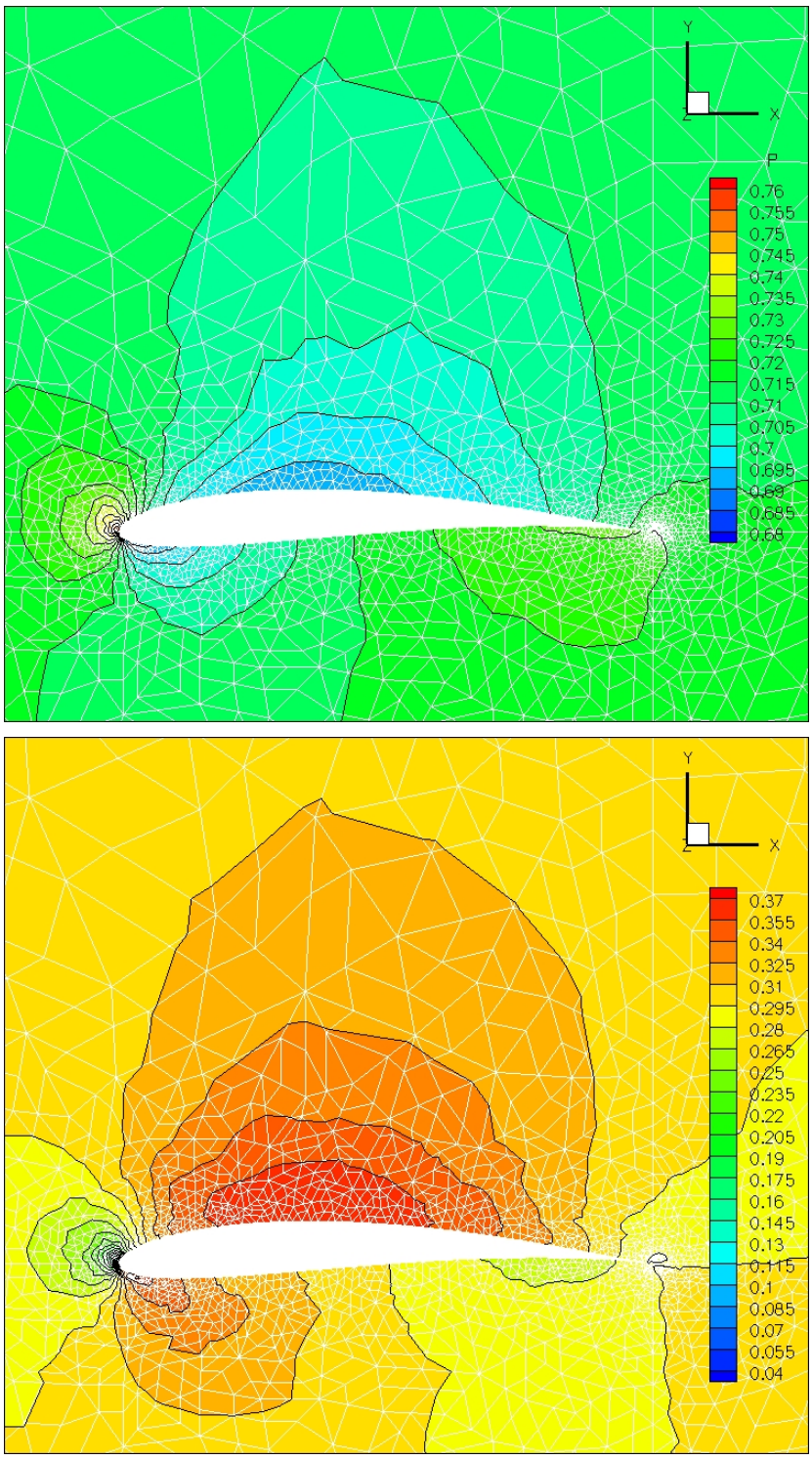


Fig. 1.4 Computed pressure and velocity magnitude for subsonic flow over a wing with Joukowsky airfoil sections using tetrahedra mesh; $\alpha = 0$ deg., $M_\infty = 0.3$

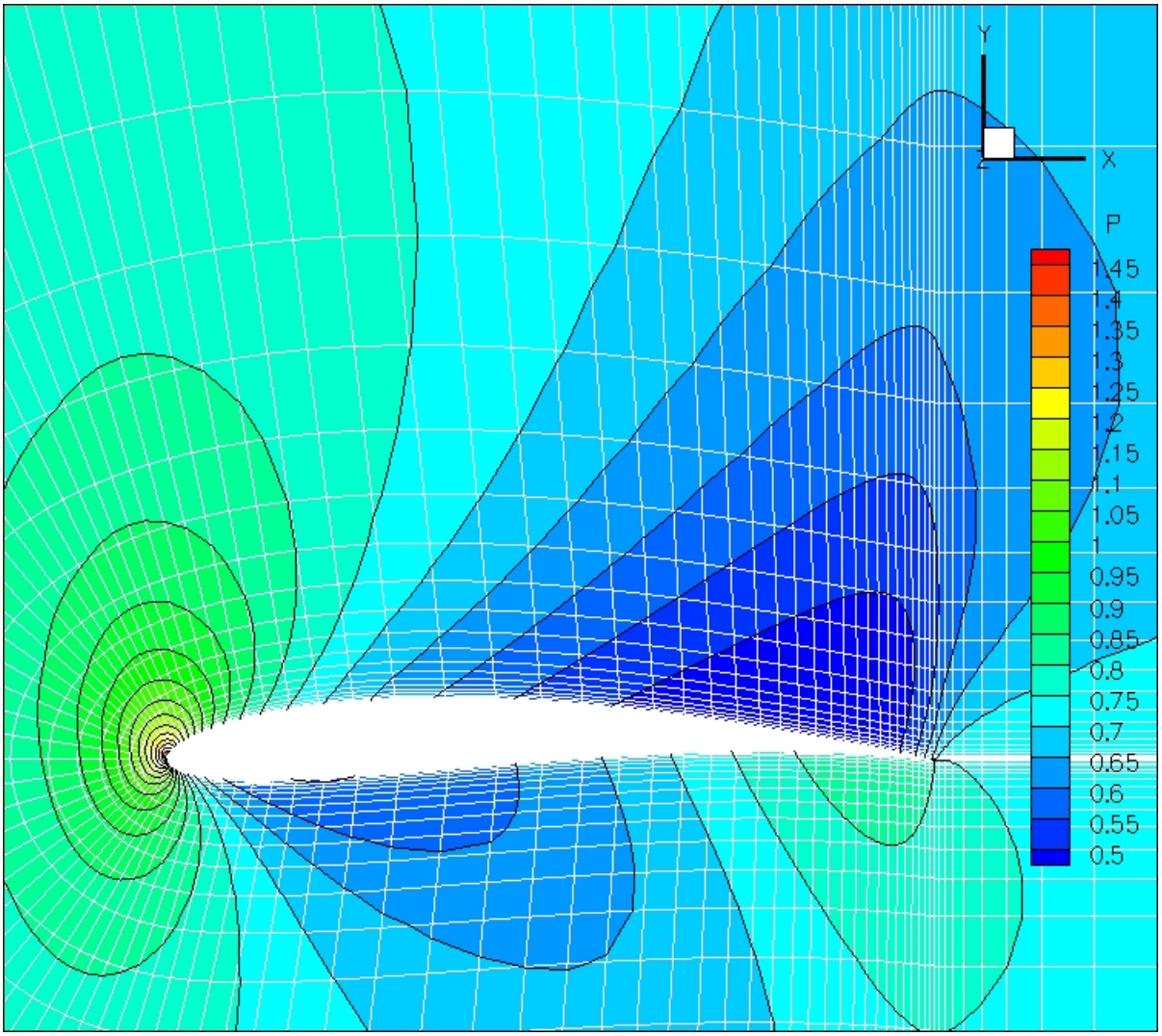


Fig. 1.5 Computed pressure field for supersonic flow over a wing with Joukowsky airfoil sections using hexahedra mesh; $\alpha = 0$ deg., $M_\infty = 1.0$

2. Parallelization using domain decomposition

In this section, we present a parallelization strategy of DG discretizations that relies on the implementation of the method in the transformed computational domain where all operations are performed for the standard square (in 2D) and cubic element (in 3D). In this manner, parallelization through domain decomposition, application of *p*–adaptive strategies, and use of mixed type meshes becomes straightforward as it will be discussed in more detail in the implementation section. One of the main disadvantages of the DG method is the severe stability limitation especially for high order expansions ($CFL \sim 1/p^2$ where p is the polynomial order). Parallelization achieved through domain decomposition and processing in multi-core clusters greatly reduced processing time. However, implementation of implicit time marching schemes is still required and it will initiate in the next period, first for linear 3D problems such as the Maxwell equations and the for the non-linear Euler and Navier-Stokes equations. In this report examples for parallel computations of subsonic and supersonic flows both in two- and three-dimensional domains are presented.

2.1 DG Implementation in the Transformed domain

The implementation of the DG method for arbitrary shaped elements in the domain of interest Ω is carried out in the computational domain for the standard square element configuration in two dimensions and for the standard cubic element in three dimension, which is denoted by Ω_{st}. A collapsed coordinate transformation, shown in Fig.2.1, is used to transform arbitrary triangles of physical space into the standard element Ω_{st}. The transformation between quadrilaterals or triangles in the physical domain to the standard square element is depicted in Fig. 2.2. Similar transformations, shown in Fig. 2.3, are used to map three-dimensional hexahedral, prismatic, and pyramid elements to the standard cubic element. Clearly, use of the collapsed coordinate transformation allows treating triangles, quadrilaterals, tetrahedral etc. in a unified way by building tensor product bases for the standard element configuration and transforming them back to the physical domain. Furthermore, the evaluation of the volume and surface integrals is greatly facilitated.

In the present work, hierarchical basis functions are constructed over the standard element configuration in the domain, (ξ,η,ζ) $[-1,1]\times[-1,1]\times[-1,1]$, where arbitrary hexahedra and tetrahedral elements in the physical space (x,y,z) transform to the standard cubic element. The basis are formed by the tensor product of Legendre polynomials. These hierarchical basis functions have been chosen in order to facilitate application of *p*-adaptive schemes of arbitrary order on mixed-type meshes. Gauss-Legendre numerical integration is used for the evaluation of the integrals in the computational domain. The Lax-Friedrichs numerical flux in the surface integral is also computed out in the computational domain.

2.2 Results

Sample results for parallel implementation of the DG method for the Euler equations are presented for subsonic and supersonic flows using hexahedral type meshes. The next step is careful optimization of the parallelization strategy in order to achieve high scalability in many-core architectures. However, two additional essential ingredients are missing from our scheme. Namely, an implicit time marching scheme, and the viscous terms, which are missing and it is not straightforward to add in the DG context. Therefore, full optimization of the parallelization strategy is postponed until the viscous terms and the implicit scheme are implemented, and it will be carried out once these modules have been added.

Supersonic flow at *M=3* over a cylinder is computed using a hexahedral mesh with arbitrary shape elements. A three-dimensional limiting procedure, which to the best of our knowledge has been not applied as yet for three dimensional DG discretizations, is used. This approach will be shown in more detail in a subsequent report. The shock capturing capability, shown in Fig 2.4, for decomposed domain parts appears very good. Computed Mach and pressure distributions on a plane are shown in Fig. 2.5. The limited elements are shown in Fig. 2.6. A more stringent shock capturing standard case is shown in Fig. 2.7 where a, *M=10,* right moving shock is reflected from a wall. The flow was computed in many domains and the shocks cross again the domain boundaries with no distortion. Furthermore limiting is confined to very narrow regions around the moving, complex shock structures.

Next, subsonic flow over a wing with Joukowsky airfoil section is computed. The airfoil shape in the $z = x + iy$ plane is obtained through the map $z = \zeta + 1/\zeta$ of the circle in the $\zeta = \xi + i\eta$ plane having centre at $(a_x, a_y) = (-0.08, 0.08)$ and radius $R = \sqrt{(1-a_x)^2 + a_y^2}$. The domain decomposition into seven parts for parallel computation is shown in Fig. 2.8. The computed flow with P1 polynomial bases (second order accuracy) for the wing with Joukowsky airfoil section at incidence $\alpha = 0$ deg. and for $M_\infty = 0.25$ is shown in Fig. 2.9.

The DG method for the three dimensional Euler equations was implemented in a generalized framework that allows implementation of mixed-type meshes and parallelized for many-core architectures using domain decomposition. The scaling of the parallelized numerical scheme is approximately linear at least for small number of processors we tested so far. The current parallelization strategy will be used as building block for parallelization of time-accurate implicit schemes in may-core architectures.

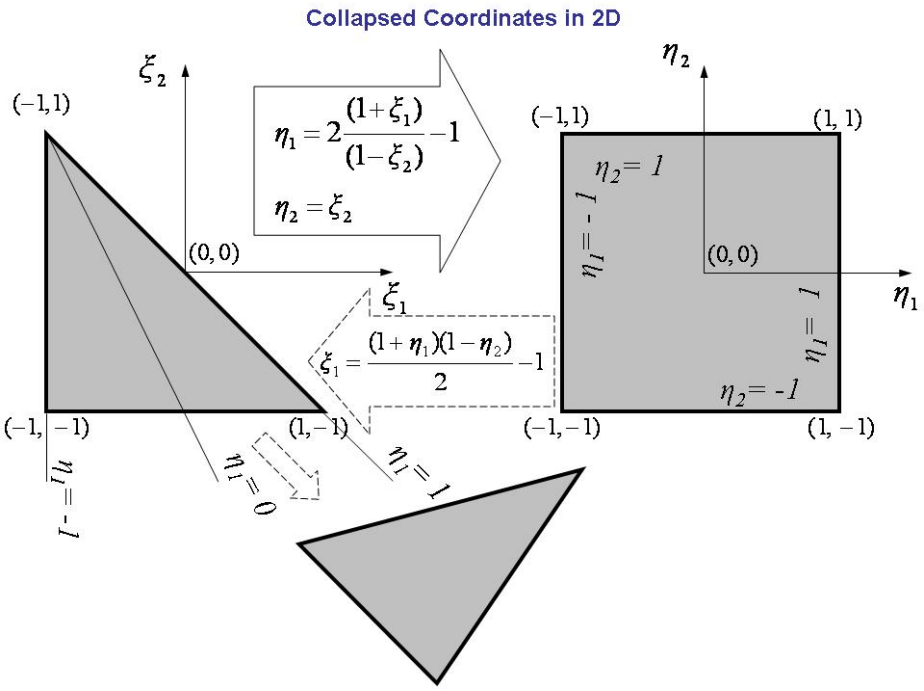


Fig. 2.1 Collapsed coordinate transformation to map arbitrary triangles of the physical domain to the standard square in the computational domain.

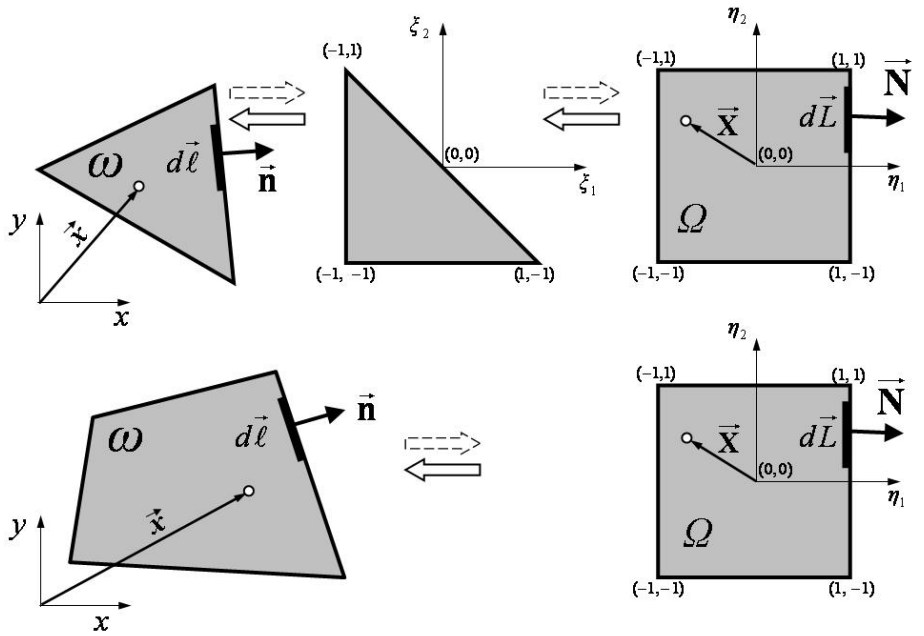


Fig. 2.2 Unified treatment of arbitrary quadrilateral and triangular elements of the physical domain by transforming to the standard square where all the numerical implementation is carried out.

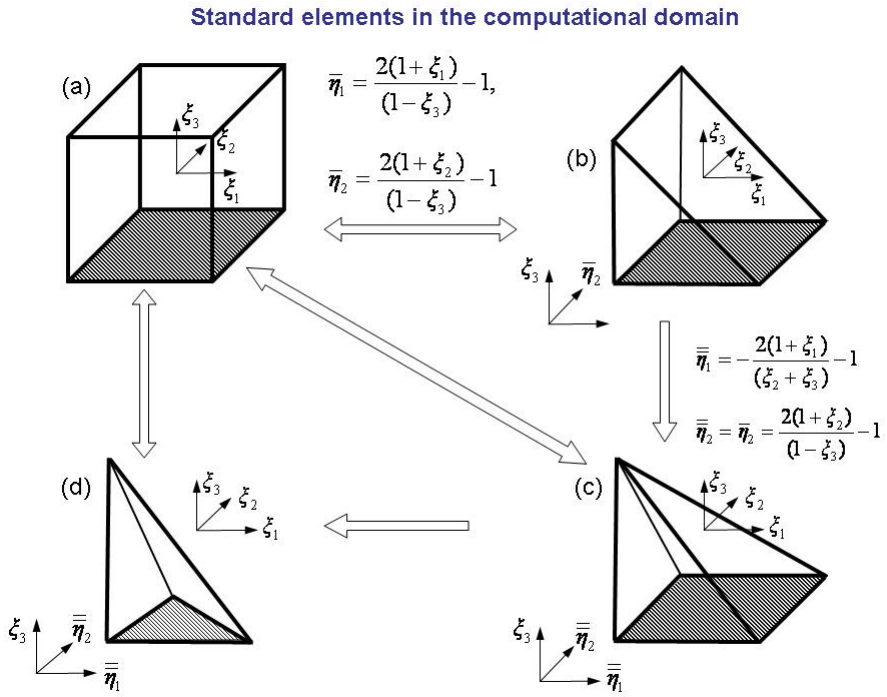


Fig. 2.3 Transformation of tetrahedral, prismatic and pyramid elements to the standard square through sequential application of collapsed coordinate transformations.

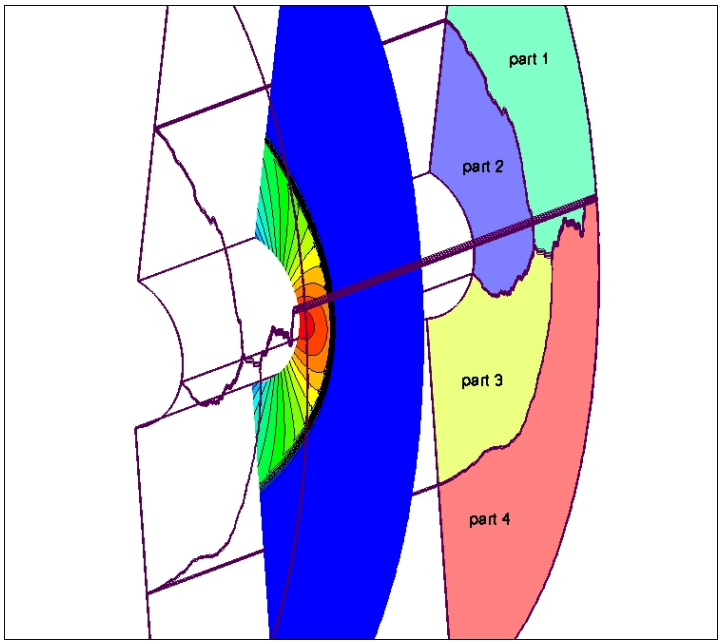


Fig. 2.4 Computed pressure distribution for supersonic flow at $M_\infty = 3.0$ over a 3D cylinder using hexahedra mesh and division of the global mesh into four parts for parallel implementation.

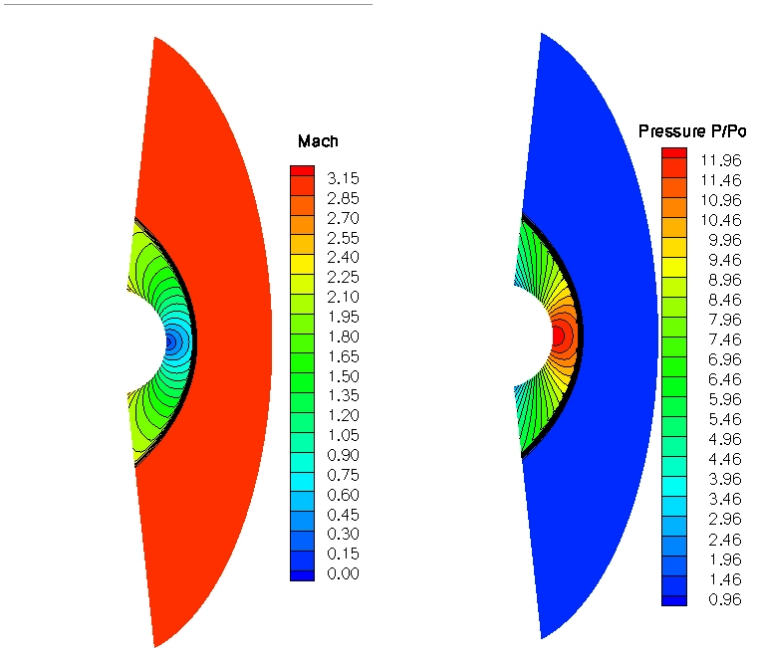


Fig. 2.5 Computed speed and pressure distribution for supersonic flow at $M_\infty = 3.0$ over a 3D cylinder

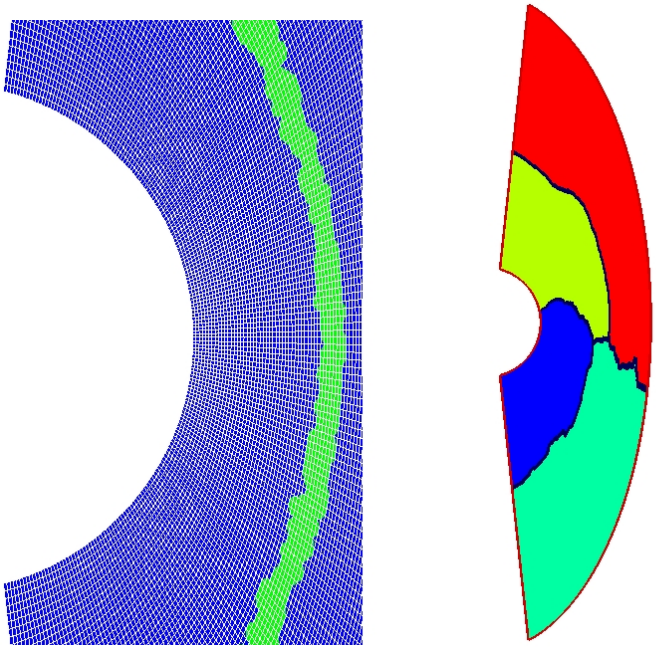


Fig. 2.6 Elements where limiting was applied (marked in green) and domain decomposition for the computation of supersonic flow over a cylinder at $M_\infty = 3.0$.

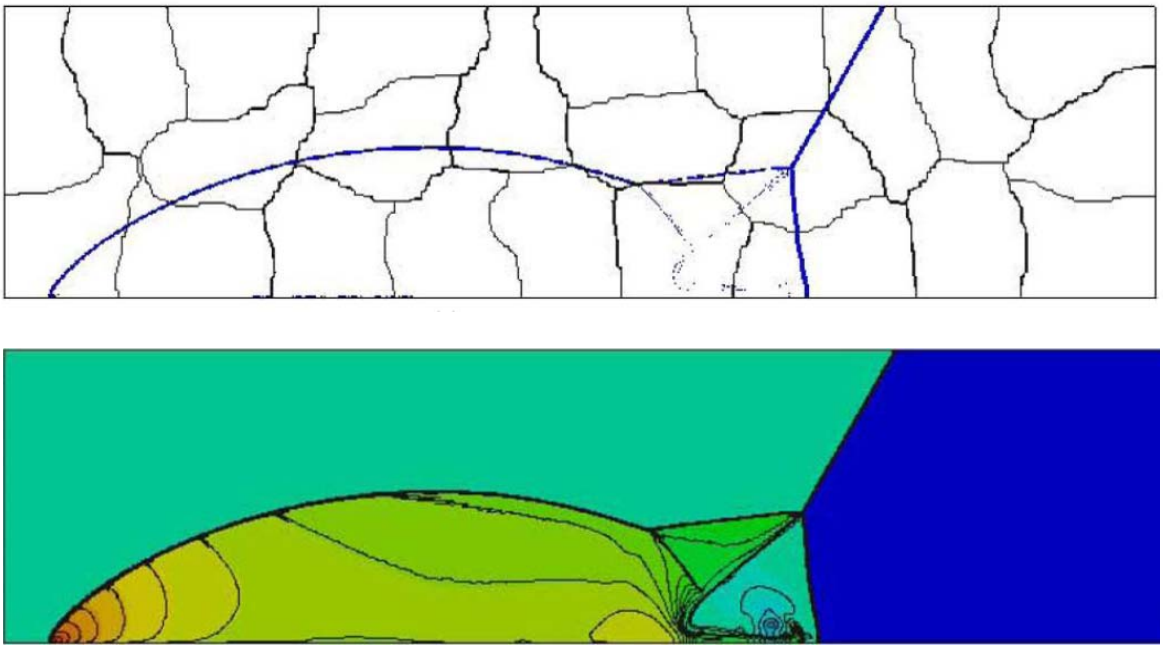


Fig. 2.7 Computed density contours for the Mach 10 reflection problem using a rectangular mesh with spacing $\Delta x = \Delta y = 1/240$ and a P^1 polynomial approximation.

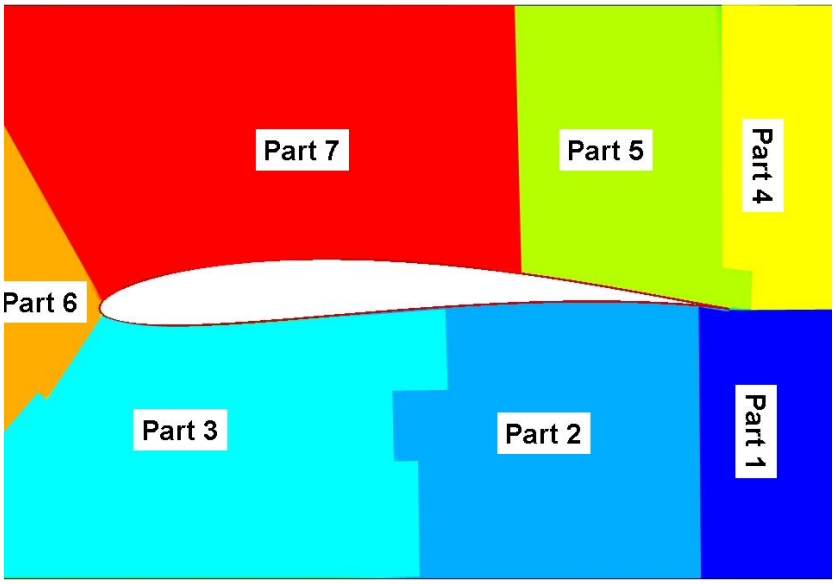


Fig. 2.8 Domain partition for the computation of subsonic flow at $M_\infty = 0.25$ over a 3D wing with Joukowsky airfoil section.

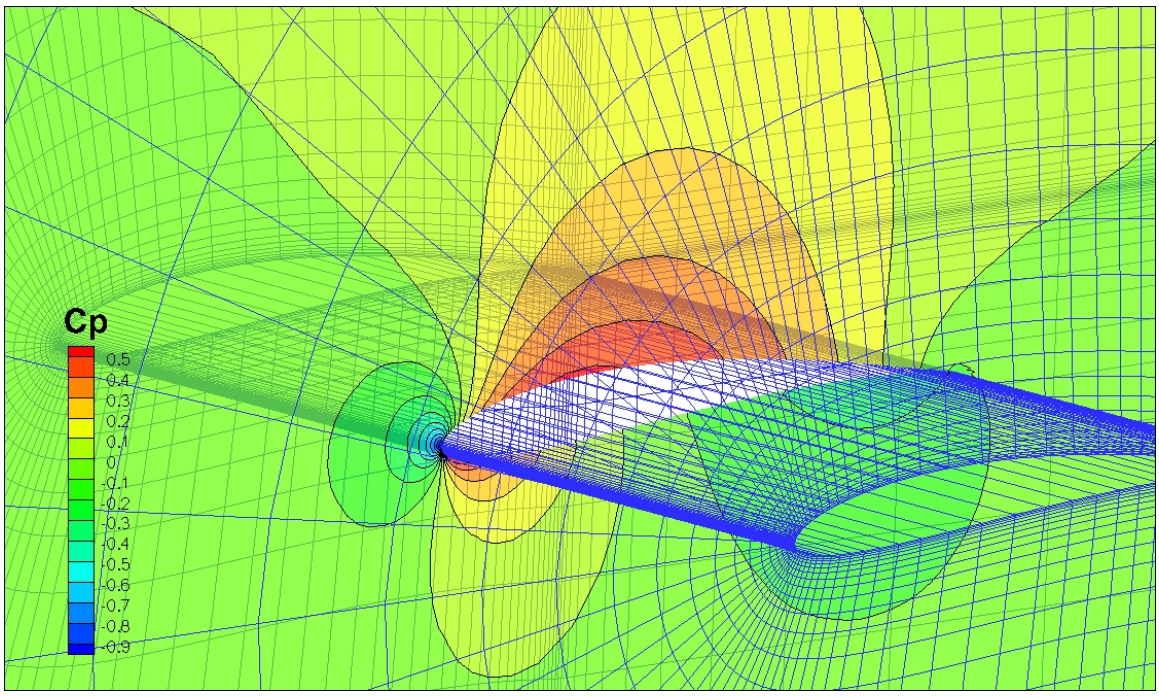


Fig. 2.9 Computed pressure coefficient distribution for subsonic flow at $M_\infty = 0.25$ over a 3D wing with Joukowsky airfoil section at incidence $\alpha = 0$ deg. using hexahedral mesh and division of the global mesh into seven parts for parallel implementation.

3. Implicit time marching scheme and implementation

Space discretization of the Euler equations with the DG method yields a system of ordinary differential equations (ODE's) for the degrees of freedom to be advanced in time. This system of ODEs has the following form:

$$[\mathbf{M}]\frac{d\mathbf{U}}{dt} + R(\mathbf{U}) = 0 \tag{3.1}$$

where $\mathbf{U}$ represents the state variables for each element (more precisely the coefficients of the polynomial expansion for each state variable to be advanced in time), $[\mathbf{M}]$ is the mass matrix, and $R(\mathbf{U})$ is the residual of the DG space discretization.

The development of implicit time integration methods on unstructured meshes has been the subject of of recent investigations Ref. 13-17. Implementation of implicit time integration is demonstrated for simplicity and without loss of generality for the backward Euler time discretization, since each stage of implicit Runge-Kutta methods is a backward Euler step. The backward Euler time discretization of the ODE system of Eq. (3.1) yields

$$[\mathbf{M}]\frac{\mathbf{U}^{n+1} - \mathbf{U}^{n}}{\Delta t} + R(\mathbf{U}^{n+1}) = 0 \tag{3.2}$$

Eq. (3.2) represents a nonlinear system of the form

$$F(\mathbf{U}^{n+1}) = 0 \tag{3.3}$$

This system is solved using a Newton-type method, which requires linearization of $F(\mathbf{U}^{n+1})$. Approximate linearization is performed as follows:

$$\mathbf{F}'(\mathbf{U}) = \frac{\mathbf{M}}{\Delta t} + \frac{\partial R(\mathbf{U})}{\partial \mathbf{U}} \tag{3.4}$$

where the term $\partial R(\mathbf{U})/\partial \mathbf{U}$ is the Jacobian of the residual and includes derivatives of the physical fluxes and the numerical fluxes. This term must be calculated accurately in order to obtain accurate linearization. However, analytical evaluation of the residual Jacobian is time consuming, difficult, and some times even impossible,

because the upwind terms included in the numerical fluxes may be non-differentiable functions, as is the Roe flux. For that reason, analytic evaluation of the system's Jacobian is performed by neglecting some terms, especially derivatives of the upwind part. Although, this appears to be an acceptable linearization from a practical point of view, its success is limited, especially for flows with discontinuities. An attractive alternative is to numerically evaluate the elements of the Jacobian matrix, regardless of the type of the numerical flux. Although, this might seem an expensive operation, its cost is comparable to that of the analytic computation of the Jacobian. Furthermore, when viscous terms are included all differentiations in the evaluations of viscous Jacobian must be performed in terms of the expansion polynomial bases and the number of operations is very large.

The general form of a k-stage approximate Newton's method for solving Eq. (3.3) is:

$$\mathbf{U}_{k+1} = \mathbf{U}_k - \left[\mathbf{F}'(\mathbf{U}_k)\right]^{-1} \mathbf{F}(\mathbf{U}_k), \quad k = 0,1,2,..... \tag{3.5}$$

with $\mathbf{U}_0$ being an initial approximation to the solution and $\mathbf{F}'(\mathbf{U}_k)$ is the Jacobian, which must be non singular at every Newton iteration. In practice, the Newton iteration in Eq. (3.5) is implemented in the following two steps:

- Approximately solve $\mathbf{F}'(\mathbf{U}_k)\Delta\mathbf{U}_k = \mathbf{F}(\mathbf{U}_k)$
- Update the iteration $\mathbf{U}_{k+1} = \mathbf{U}_k + \Delta\mathbf{U}_k$

where the iterations are terminated based on a required drop in the norm of the nonlinear residual. At every Newton iteration, a linear system needs to be solved, and the solution is usually obtained by an iterative method. Despite of the sparse nature of the matrix size of the system is very large especially for high order expansions and the computational cost and storage requirements become very large. Direct methods for the solution of this system are not suitable for nonlinear problems such as the Euler equations but they may become an attractive alternative for linear problems such as the Maxwell equations.

It should be noted here that, for parallel implementation through domain decomposition, during the Newton iteration updated values of $\Delta\mathbf{U}_k$ at the boundaries are not available. Therefore an additional outer iteration loop must be established in order to ensure that the solution is fully updated between the domains at the end of each time step. Achieving updated values (within certain tolerance) between the adjacent domains that are involved in the numerical flux which is part of the residual is very crucial element for time dependent problems. For steady-state problems, the tolerance between the consecutive approximations of the boundary values could be relaxed. However, for time accurate computations this tolerance should be down to machine accuracy. As a result, the number of the required iterations becomes larger and the computing cost increases. The technique used to ensure full update between the sub-domains will be demonstrated in more detail later.

3.1 Krylov subspace methods

Krylov subspace methods are approaches for solving large linear systems either by direct or iterative methods. They are generalized projection methods for solving a linear system:

$$[\mathbf{A}]\mathbf{x} = \mathbf{b}$$

using the Krylov subspace, Kj : Kj = span (r_0, Ar_0,A^2r_0, . . . ,$A^{j-1}r_0$), where r_0 is the residual, $r_0 = b - Ax_0$, and x_0 is an initial guess of the solution.

The widely used GMRES (Generalized Minimal Residual) method belongs to the class of Krylov subspace methods. For GMRES, the Krylov subspace is formed by an Arnoldi based method, which is an orthogonalization procedure that generates orthonormal bases of the Krylov subspace. The main advantage of the GMRES method is that it does not require the explicit formulation of matrix **A**, it rather requires the matrix-vector products used to form the Krylov subspace. This application of the GMRES method is called matrix-free, and results in significant savings in storage that are crucial for three dimensional applications of high order DG discretizations. In the context of Newton's method, this feature leads to the so called Jacobian free Newton Krylov (JFNK) methods, where the Jacobian matrix needs to be accessed only in the form of matrix-vector products. That is:

$$\mathbf{F}'(\mathbf{U})\,\Delta\mathbf{U} \approx \frac{\mathbf{F}(\mathbf{U}+h\Delta\mathbf{U}) - \mathbf{F}(\mathbf{U})}{h}$$

where the differencing parameter *h* is computed as follows:

$$h = \begin{cases} e_{rel}\dfrac{\mathbf{U}^T\Delta\mathbf{U}}{\|\Delta\mathbf{U}\|_2^2}, & |\mathbf{U}'\Delta\mathbf{U}| > u_{\min}\,\|\Delta\mathbf{U}\|_1 \\ e_{rel}u_{\min}\,sign(\mathbf{U}^T\Delta\mathbf{U})\dfrac{}{\|\Delta\mathbf{U}\|_2^2} & \text{otherwise} \end{cases} \qquad (1.6)$$

3.2 GMRES Preconditioning

Close approximation of the Jacobian matrix of the system is required in order to ensure accelerated convergence of the iterative method for the linear system. For the JFNK method in particular, a very good preconditioning is required. For that reason, in the present work a numerical approximation of the complete Jacobian matrix is performed, where in order to speed up the computation of its elements, the sparsity of the

matrix is used in advance. The matrix elements are computed by the following first order accurate approximation:

$$\mathbf{F}_{ij}^{'}(\mathbf{U}) \approx \frac{\mathbf{F}_i(\mathbf{U}+he_j)-\mathbf{F}_i(\mathbf{U})}{h} \tag{3.7}$$

where index *i* refers to the global number of the DOF at the element and index *j* refers to the global number of the DOF of the neighbouring element.

The full procedure for each time step *n* along with the strategy used for the update of the information for multi-domain updates that are required when domain decomposition is used for parallel processing is shown in the schematic flow chart of Fig. 3.1.

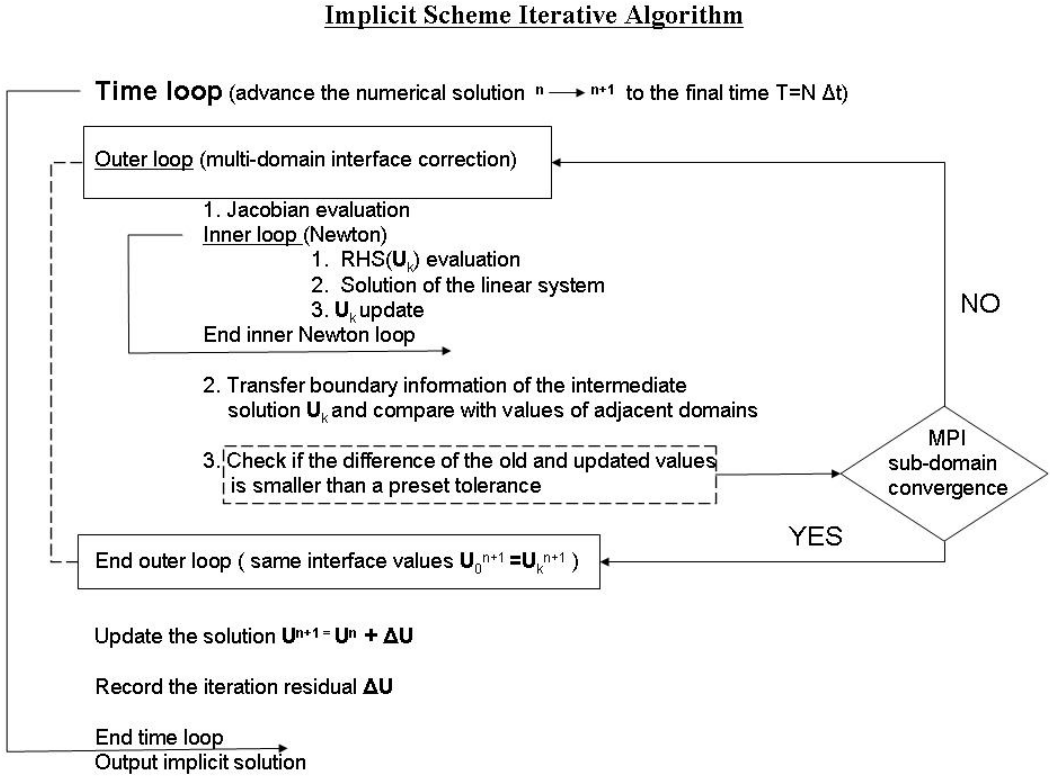


Figure 3.1. Iteration loop implementation to ensure that when domain decomposition is applied values from the adjacent sub-domains involved in the residual $R(\mathbf{U}^{n+1})$ have been fully updated and converged to the same time step $n+1$.

3.2 Results

Results with implicit time marching for steady-state and time-accurate solutions are presented for model problems of two dimensional flows solved with three dimensional meshes. All solutions with implicit time marching were obtained with parallel implementation. The first class of results are steady-state solutions for inviscid low speed flow (*M=0.2*) over a cylinder and an airfoil. Then implicit time marching is applied for supersonic (*M=2.0*) flow over a cylinder. Finally, a time-accurate solution for the convection of an isentropic vortex is shown.

The computed flow field and the comparison with the exact result for inviscid flow over a cylinder are shown in Figs. 3.2 and 3.3. The convergence rate of the computed solution with $\Delta t = 0.08$, CFL~500 is shown in Fig. 3.4. The convergence was based on the L_2 norm of the residual computed as:

$$L_2(\Delta\rho) = \frac{1}{N_{el}} \sqrt{\sum_{j=1}^{N_{el}} \left| \rho_j^{n+1} - \rho_j^n \right|^2}$$

Convergence to machine accuracy was achieved in approximately 1000 iterations.

The computed flow field with $\Delta t = 0.08$, CFL~500 and the comparison with the exact result for inviscid low speed (M=0.2) flow over a Joukowski airfoil are shown in Figs. 3.5 and 3.6. The convergence rate of the computed solution is shown in Fig. 3.7 similarly to the cylinder case converge to machine accuracy was achieved.

Finally, the numerical solution of a time accurate problem was computed using implicit time marching. The convection of an inviscid isentropic vortex is a good test problem for evaluating the validity of the implicit time marching approach because it is an exact solution of the compressible Euler equations. Fort his problem the mean density ρ_∞, velocities u_∞, v_∞, pressure p_∞, and temperature T_∞ have the free stream values: $(\rho_\infty, u_\infty, v_\infty, p_\infty, T_\infty) = (1, 0.5, 0, 1/\gamma, 1)$ that are perturbed by the isentropic vortex $(\delta u, \delta v, \delta T)$

$$\delta u = -\frac{\Gamma}{2\pi}(y - y_0)e^{A(1-r^2)},\ \delta v = +\frac{\Gamma}{2\pi}(x - x_0)e^{A(1-r^2)},\ \delta \mathrm{T} = -\frac{\Gamma^2(\gamma-1)}{16\mathrm{A}\gamma\pi^2}e^{2A(1-r^2)}$$

where $r = \sqrt{(x-x_0)^2 + (y-y_0)^2}$ and $\Gamma = 4.0$, $A = 1$ are parameters determining the strength of the vortex. For invisicd flow, this isontropic vortex convects with speed u_∞ without change of form and the accuracy of time advancement can be evaluated. Numerical solutions were obtained with very large values of time steps (CFL~10^3) and the convergence rate of the numerical solution is shown in Fig. 3.8. It can be seen that the design order of accuracy of the backward Euler method is almost exactly achieved. Higher order accurate implicit time marching methods such as the Crank-Nicolson, the implicit RK2, and the implicit RK3 method have been

implemented. However, accuracy test for these methods are very time consuming because either small element size or high order of accuracy must be used to ensure that space discretization error are much smaller than the time discretization induced error.

Implicit time marching schemes for the Euler equations were developed and applied for steady-state and time accurate computations in multi domains. Large CFL numbers were achieved and time accuracy was maintained. Implicit time marching of viscous flows and further demonstration of time accuracy will be carried out next.

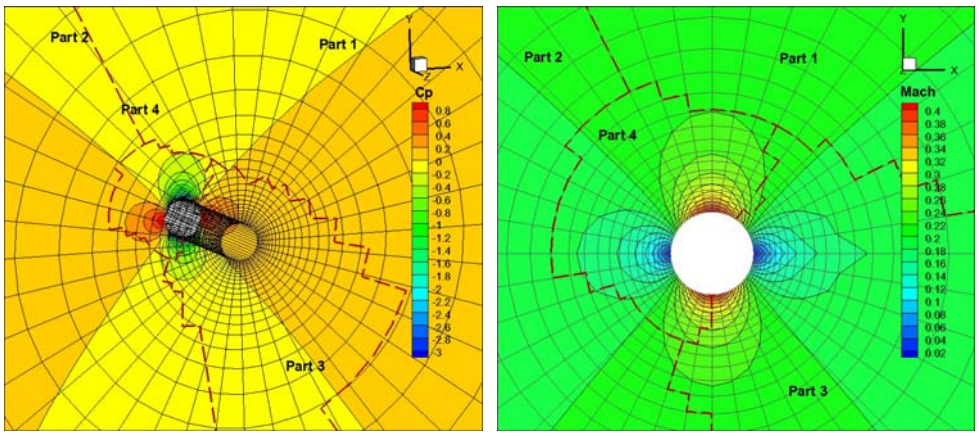


Figure 3.2. Computed flow field for invisicd *M=0.2* low speed flow over a cylinder including the sub-domain partition for MPI processing.

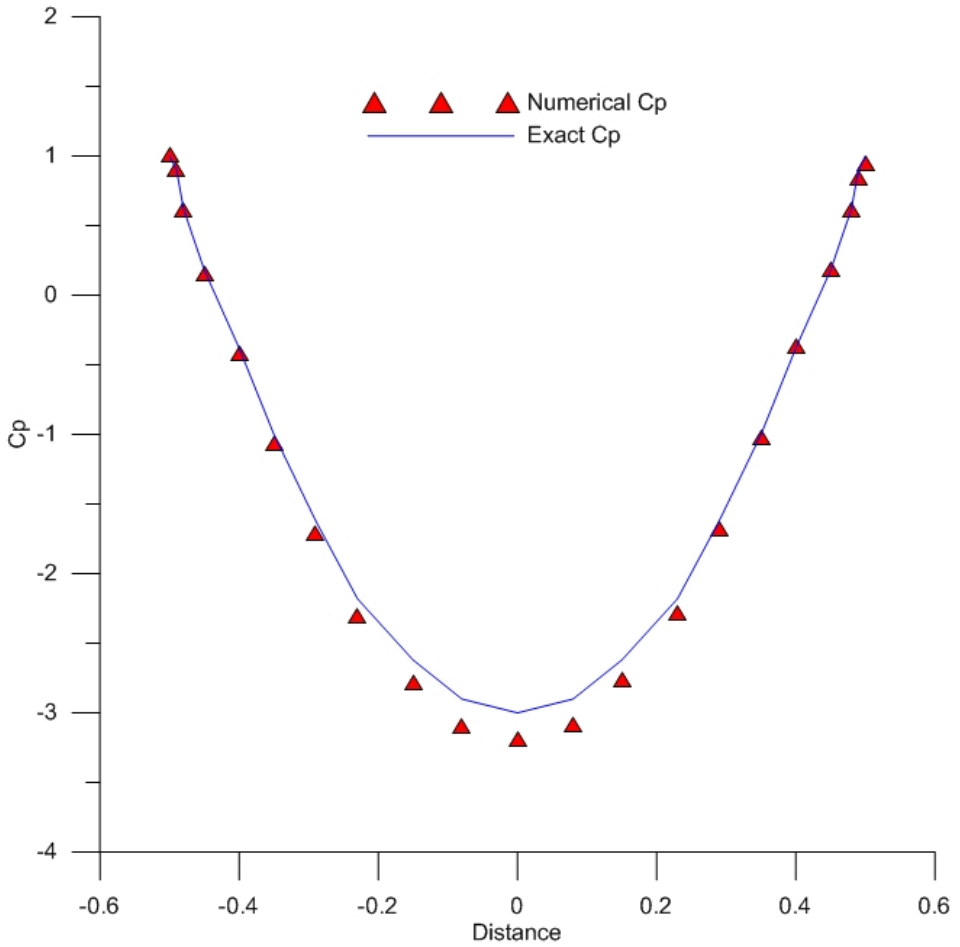


Figure 3.3. Comparison of the surface pressure computed for invisicd, *M=0.2,* low speed flow with the potential flow exact result for the cylinder.

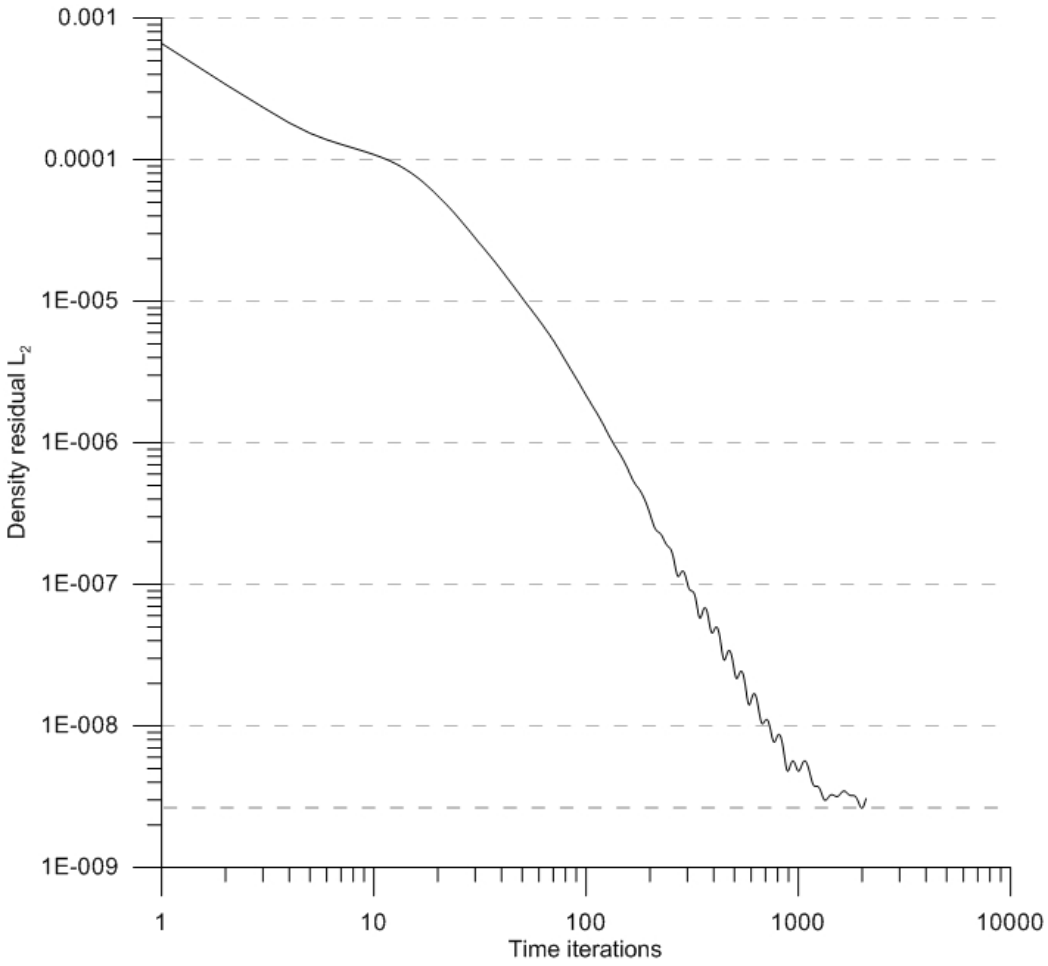


Figure 3.4. Convergence rate based on the L_2 norm of the density residual for the computation for invisicd, *M=0.2,* low speed flow over a cylinder.

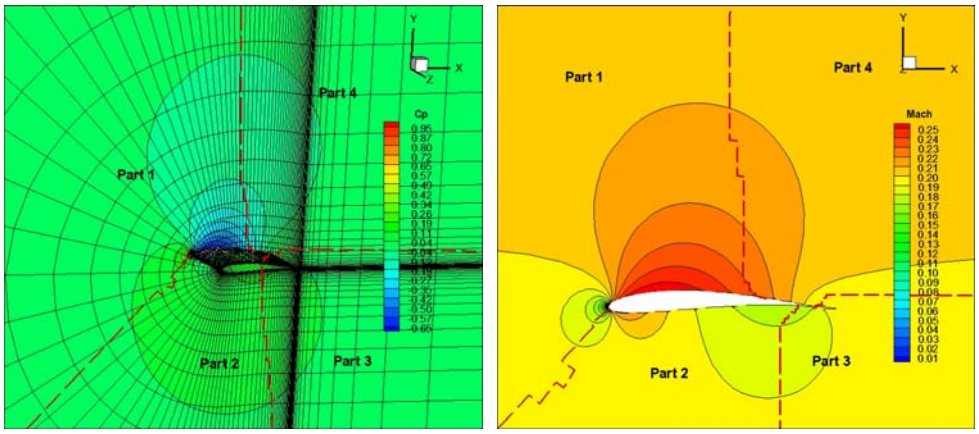


Figure 3.5. Computed flow field for invisicd, *M=0.2,* low speed flow over the Joukowski airfoil including the sub-domain partition for MPI processing.

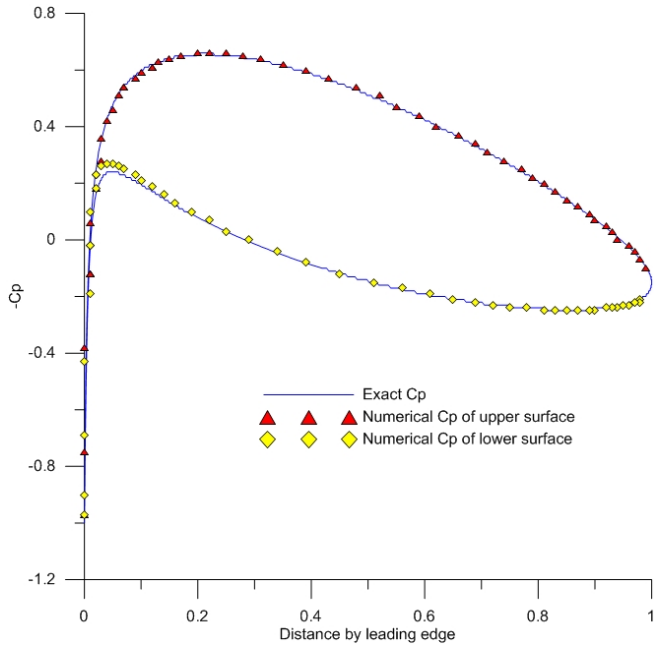


Figure 3.6. Comparison of the surface pressure computed for invisicd, *M=0.2,* low speed flow with the potential flow exact result for the Joukowski airfoil.

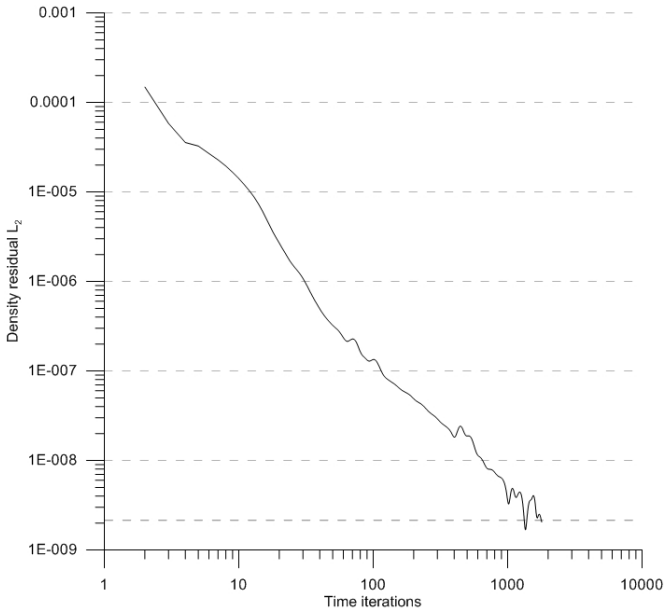


Figure 3.7. Convergence rate based on the L_2 norm of the density residual for the computation for invisicd, *M=0.2,* low speed flow over the Joucowski airfoil.

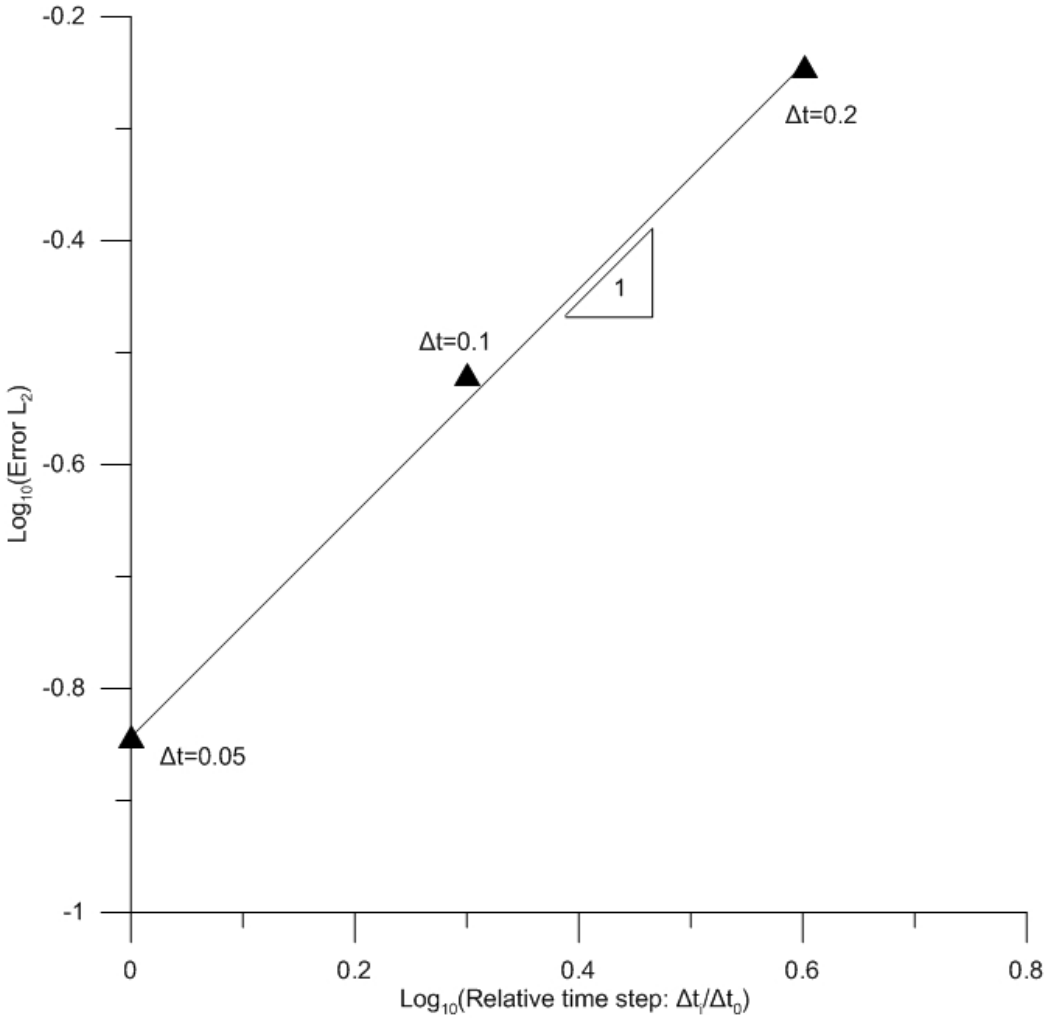


Figure 3.8. Time convergence of backward Euler implicit time integration scheme obtain for the convection of an isentropic vortex between four sub-domains and parallel implementation of the implicit algorithm.

4. NS solutions and turbulence model

The once equation turbulence model of Spalart and Almaras (SA) was implemented in the DG finite element context. For the implementation of this model in the DG context several new features, such as treatment of a nonlinear diffusion term, appear. Therefore in order to verify the discretization strategy of the SA turbulence model, the algebraic turbulence model of Baldwin and Lomax (BL), which does not pose problems in terms of discretization but is rather not very well suited to use with unstructured meshes, has been also implemented. Comparisons of the computed eddy viscosity profiles for a zero pressure gradient, flat plate, turbulent boundary layer demonstrated that both the SA and BL turbulence models yield essentially the same velocity and eddy viscosity distribution. Furthermore, in order to overcome numerical problems caused by negative eddy viscosities computed by the SA model at the edge of the boundary layer positivity preserving limiters were used for the computed eddy viscosity.

Strategies for p-adaptive refinement of three dimensional flows were further examined and an attempt was made to incorporate p-type multigrid in the implicit solver to further enhance the efficiency and accuracy of implicit time marching with large time steps.

4.1 Background

The one equation Spalart-Almaras (SA) turbulence model [18-21] in conservative form suitable for compressible flow numerical solutions is

$$\underbrace{\frac{\partial \rho\tilde{\nu}}{\partial t}}_{I}+\underbrace{\frac{\partial \rho\tilde{\nu}u_j}{\partial x_j}}_{II}=\underbrace{\frac{1}{\sigma}\left[\frac{\partial}{\partial x_j}\left((\mu+\rho\tilde{\nu})\frac{\partial\tilde{\nu}}{\partial x_j}\right)+\rho c_{b2}\frac{\partial\tilde{\nu}}{\partial x_j}\frac{\partial\tilde{\nu}}{\partial x_j}\right]}_{III}+\underbrace{c_{b1}\tilde{S}\tilde{\nu}}_{IV}-\underbrace{(c_{w1}f_w)\frac{1}{\rho}\left(\frac{\rho\tilde{\nu}}{D}\right)}_{V} \qquad (4.1)$$

In this equation $\tilde{\nu}$ is the working variable that is related to eddy viscosity μ_t as:

$$\mu_t=\rho\nu_t=\rho\tilde{\nu}f_{\nu 1} \qquad (4.2)$$

Terms *I* and *II* express convention of the density scaled working variable $\rho\tilde{\nu}$, term *III* expresses diffusion, the production is part *IV,* and the destruction is part *V*.

The term $\tilde{S}$ of the production (part *IV* of the right hand site) is given by

$$\tilde{S}=|\Omega|+\frac{\tilde{\nu}}{\kappa^2 D^2}f_{\nu 2}, \qquad (4.3)$$

where $|\Omega|$ is the magnitude of the vorticity vector and *D* is the distance to the nearest wall. The need of the wall distance *D* is a fundamental difference compared to $k-\omega$ and $k-\varepsilon$ two equation turbulence models which could manage without information about

distance from the nearest wall. To overcome this difficulty the distance from the nearest wall was computed and stored. The definition of the model is completed by the auxiliary relations

$$
\begin{aligned}
& f_w = g\left[\frac{1+c_{w3}^6}{g^6+c_{w3}^6}\right], \qquad g = r + c_{w2}(r^6 - r), \quad r = \min\left[\frac{\tilde{v}}{\tilde{S}\kappa^2 D^2},\ 10\right], \\
& f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}}, \quad \chi = \frac{\tilde{v}}{v} \\
& f_{v2} = 1 - \frac{\chi}{1+\chi f_{v1}}
\end{aligned}
\tag{4.4}
$$

The closure constants of the model are

$$
\begin{aligned}
& c_{b1} = 0.1355, && c_{b2} = 0.622, && \sigma = 2/3, && \kappa = 0.41 \\
& c_{w1} = c_{b1}/\kappa + (1+c_{b2})/\sigma, && c_{w2} = 0.3, && c_{w3} = 2.0, && c_{v1} = 7.1
\end{aligned}
\tag{4.5}
$$

The nondimensional for of the SA one equation turbulence model is

$$
\underbrace{\frac{\partial \rho\tilde{v}}{\partial t}}_{I} + \underbrace{\frac{\partial \rho\tilde{v}u_j}{\partial x_j}}_{II} = \underbrace{\frac{1}{\mathrm{Re}}\frac{1}{\sigma}\left[\frac{\partial}{\partial x_j}\left((\mu+\rho\tilde{v})\frac{\partial\tilde{v}}{\partial x_j}\right) + \rho c_{b2}\frac{\partial\tilde{v}}{\partial x_j}\frac{\partial\tilde{v}}{\partial x_j}\right]}_{III} + \underbrace{c_{b1}\tilde{S}\tilde{v}}_{IV} - \underbrace{(c_{w1}f_w)\frac{1}{\rho}\left(\frac{\rho\tilde{v}}{D}\right)}_{V} \tag{4.6}
$$

where now

$$
\tilde{S} = |\Omega| + \frac{1}{\mathrm{Re}}\frac{\tilde{v}}{\kappa^2 D^2} f_{v2}, \quad r = \min\left[\frac{\tilde{v}}{\mathrm{Re}\,\tilde{S}\kappa^2 D^2},\ 10\right] \tag{4.7}
$$

The one equation turbulence model of Eq. (1) is discretized in the DG context decoupled from the flow equations e.g. the density ρ and the convective velocity u_j are taken from the flow solution at the current time step. The local DG (LDG) method is used and the gradient $\mathbf{N} = \partial\tilde{v}/\partial x_j$ appearing in the diffusive part of the model is evaluated first. Then after integration by parts the nonlinear diffusion term $\tilde{v}\partial\tilde{v}/\partial x_j$ can be evaluated.

Numerical implementation of the SA model for complex time dependent flows often produces negative values of $\tilde{v}$ which subsequently yield negative eddy viscosity, $\mu_t = \rho\tilde{v}f_{v1}$ since f_{v1} has the same sign with the working variable $\tilde{v}$. Therefore the following positivity limiters that ensure positive $\tilde{v}$ value were used. Positivity is enforced by first computing the minimum value of the working variable $\tilde{v}_{\min}$ looping over the quadrature points on the edges. Then the following
parameter is defined:

$$\theta_1 = \min\left\{ \frac{c_0^{\tilde{\nu}} - \varepsilon}{c_0^{\tilde{\nu}} - \tilde{\nu}}, 1 \right\}, \quad \varepsilon = \min(10^{-3}, c_0^{\tilde{\nu}}) \tag{4.8}$$

and the coefficient for the working variable expansion are modified as follows:

$$\hat{c}_0^{\tilde{\nu}} = \theta_1 c_0^{\tilde{\nu}} \tag{4.9}$$

Proper discretization of the SA model was verified by comparing results with the eddy viscosity obtained from the algebraic Balwin-Lomax (BL) turbylence model for a simple flow case the zero pressure gradient flat plate turbulent boundary layer. The eddy viscosity for the BL model is obtained for a direction normal to the wall. Therefore rays normal to the wall were drawn where needed as shown in Fig. 4.1 and the eddy viscosity was computed.

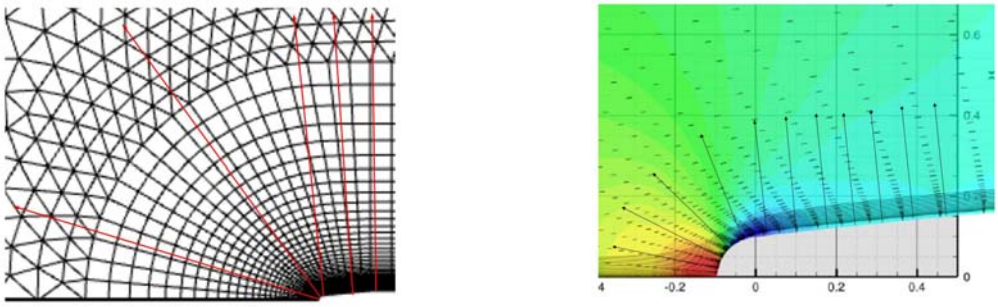


Figure 4.1. Normal to the wall rays for the computation of the eddy viscosity with the algebraic BL model, (left) computational mesh (right) pressure and velocity.

It is necessary that the SA turbulence model is integrated in time using and implicit method otherwise the explicit time step limitations for the SA turbulence model are more severe that for the main flow solver. It must be noted that the SA turbulence model is run decoupled from the solution of the flow field and the velocities and the density in Eq. 1 are considered known. The implicit time marching scheme used for the time advancement of the SA is shown in Fig. 4.2.

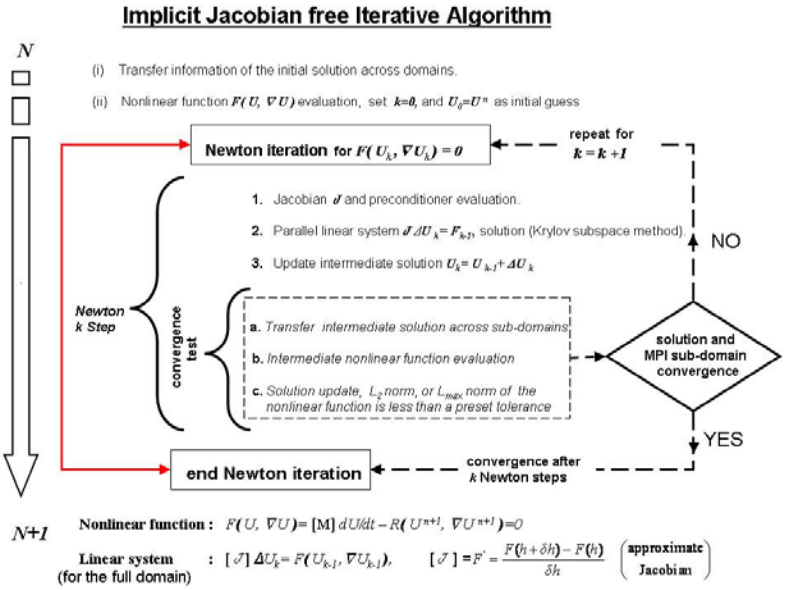


Figure 4.2. Implicit algorithm for the SA model.

The PETC library that is used for the implementation of the Jacobian free algorithm gives the options of using many implicit Runge-Kutta (RK) methods for time advancement. In addition, it incorporates the so-called implicit/explicit RK methods which are used for the SA model since the source terms IV and V in Eqs. (4.1) and (4.6) are not linearized but they are kept in the right hand side.

The gradient tem $\vec{\mathbf{N}} = \partial\tilde{v}/\partial x_j$ is evaluated first and the local DG (LDG) method is used to treat diffusion terms of the model. As a result the following system of equations is solved.

$$\vec{\mathbf{N}} = \partial\tilde{v}/\partial x_j$$

$$\frac{\partial\rho\tilde{v}}{\partial t} + \frac{\partial\rho\tilde{v}u_j}{\partial x_j} - \frac{1}{\mathrm{Re}\,\sigma}\left[\frac{\partial}{\partial x_j}\left((\mu+\rho\tilde{v})\vec{\mathbf{N}}\right) + \rho c_{b2}\,|\,\vec{\mathbf{N}}\,|^2\right] = c_{b1}\tilde{S}\tilde{v} - (c_{w1}f_w)\frac{1}{\rho}\left(\frac{\rho\tilde{v}}{D}\right) \qquad (4.10)$$

As an alternative the hybridized DG method is currently considered for the discretization of the turbulence model because it provides certain advantages compared with the LDG method in particular after post processing of the numerical solution it can provide k+2 order of accuracy for k order polynomial approximation.

4.2 Turbulence model implementation

Computed velocity profiles for a flat plate boundary layer at Re= 2×10^6 are shown in Figs. 4.3-4.6. A hybrid (hexahedral/prismatic) mesh over a flat plate of total length 10 units with a rounded leading edge, as shown in Fig. 4.1, was constructed. The computed eddy viscosity field and the velocity at x=5 are shown in Fig. 4.3. Linear plots on the computed axial velocity and eddy viscosity distribution are shown in Figs. 4.4-4.6.

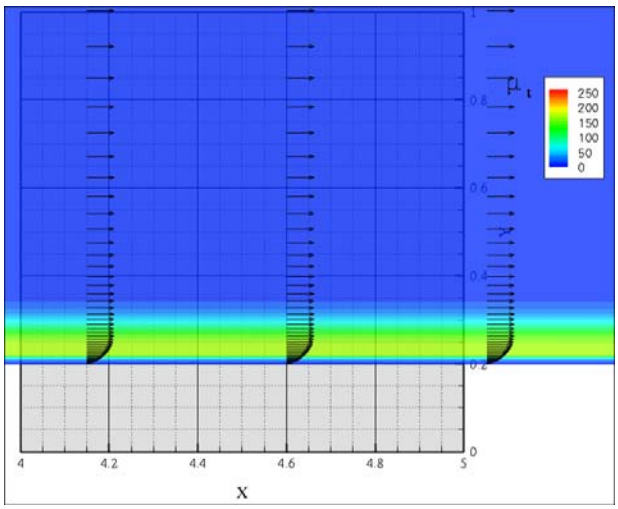


Figure 4.3. Computed velocity vectors and eddy viscosity with the SA turbulence model at x=5, Re= 10^6

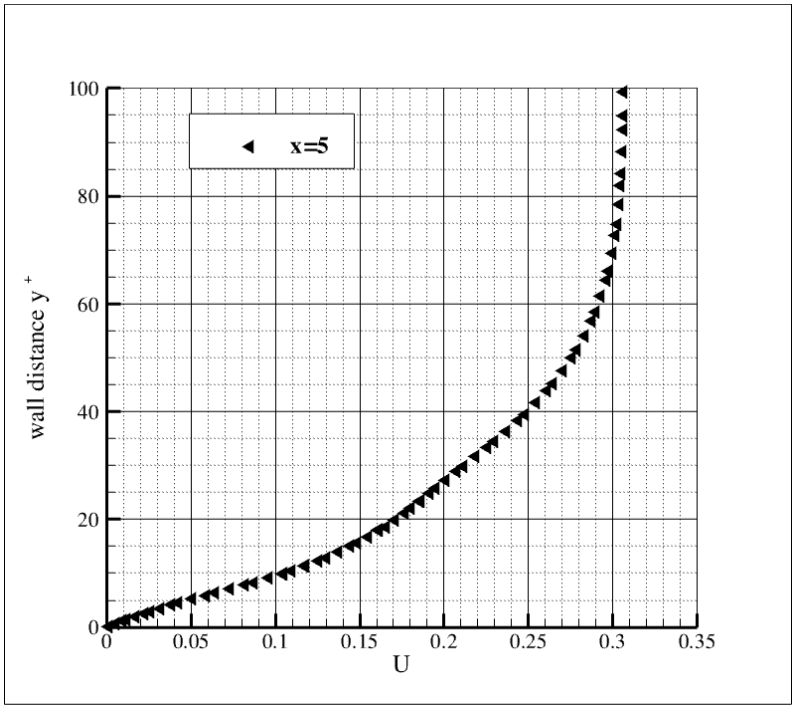


Figure 4.4. Computed u velocity with the SA turbulence model at x=5, Re=10^6

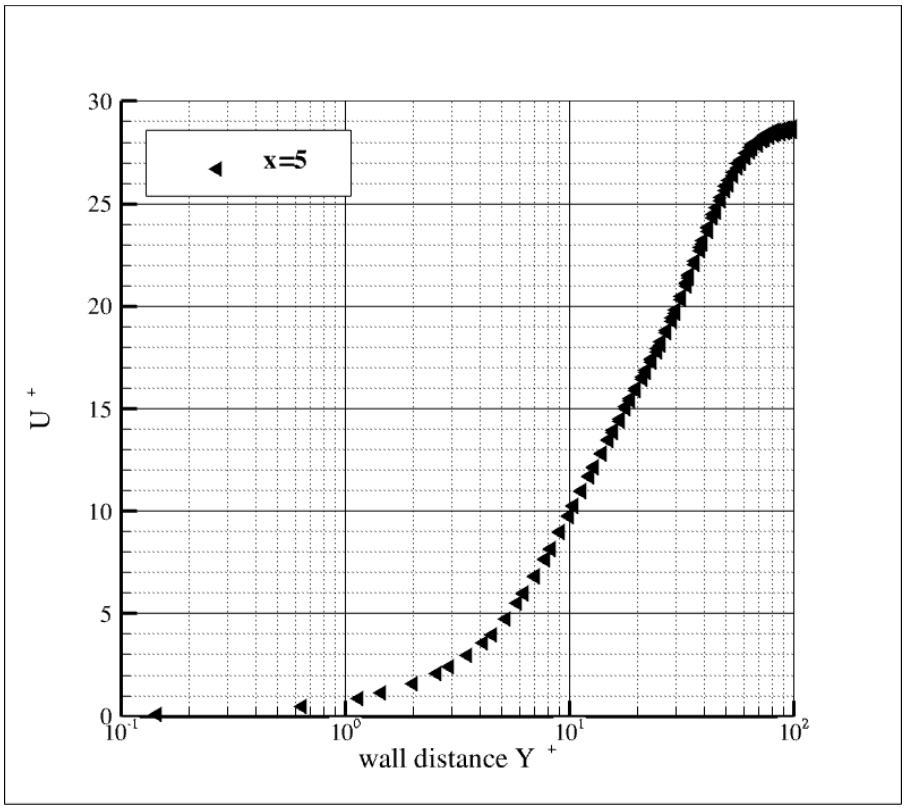


Figure 4.5. Log-log scale y^+, u^+ diagram of the computed u velocity with the SA turbulence model at x=5, Re=10^6

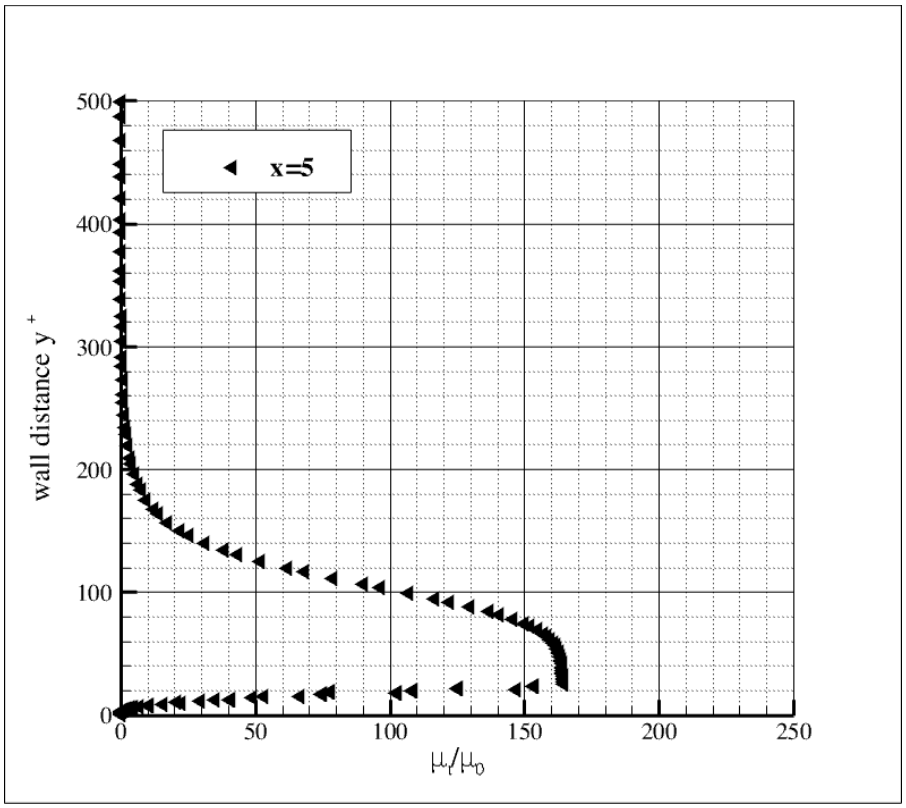
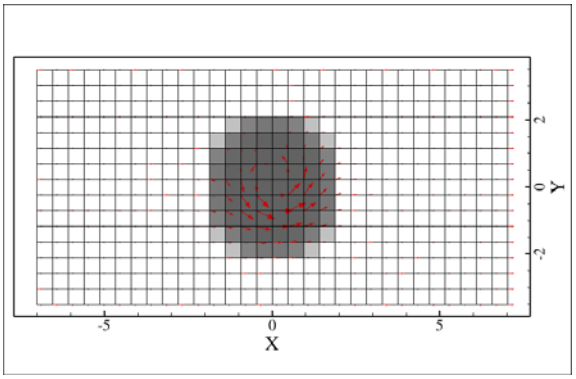


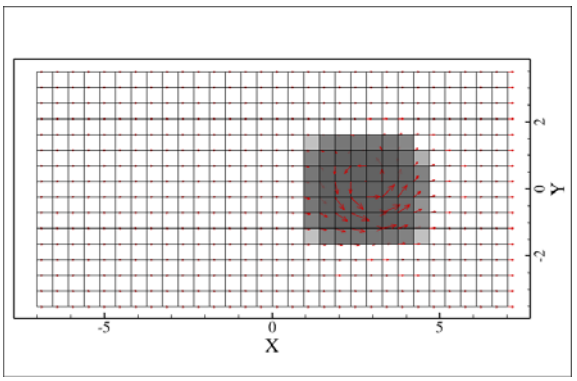
Figure 4.6. Computed eddy viscosity with the SA turbulence model at x=5, Re=10^6

4.3 P-adaptive numerical solutions

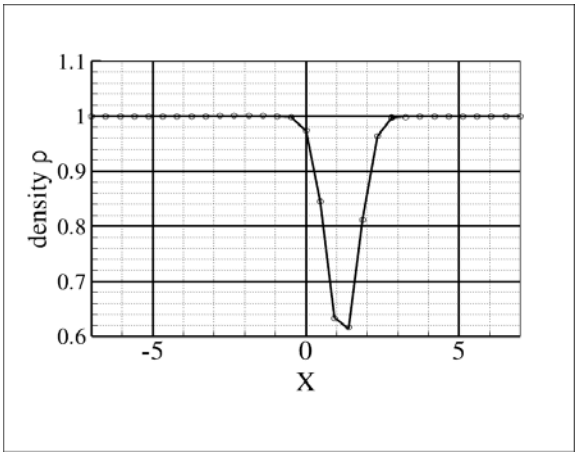
Results for adaptive p-refinement of time dependent solutions are shown bellow. Without loss of generality results of p-adaptive numerical solutions were computed for hexahedral meshes. A snapshot of the computed flow field for a p-adaptive numerical solution for an inviscit, isentropic, convecting voretex, which is an exact solution of the Euler equations, is shown in Fig. 4.7. Adaptivity of the numerical solution was based on the computed vorticity. The irrotational part of the flow field was computed as second order accurate (P1 computation). The accuracy progressively increased in regions of higher vorticity from P2 to P3. Comparisons with the exact result of numerical solutions computed with P3 global accuracy and progressively increase P1 to P5 accuracy are shown in Fig. 4.8. Clearly, increase of the accuracy over P3 only for 16 cells improved the accuracy of the numerical solution while at the same time the overall computational cost of the p-adaptive numerical was diminished.



(a)



(b)



(c)

Figure 4.7. Snapshot of a P1 to P3, p-adaptive numerical solution; white cells P1, light gray cells P2, and dark gray cells P3. (a) convection for 15 unit lengths (b) convection for 16 unit lengths (c) comparison with exact

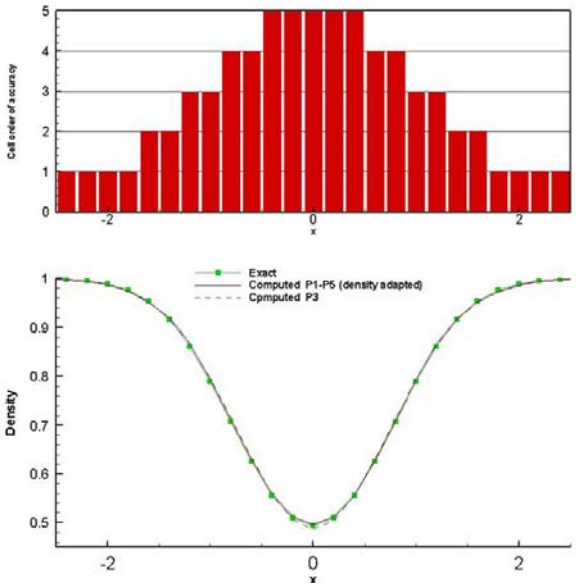


Figure 4.8. Comparison of the computed density with global P3 and adaptive P1-P5 numerical solutions.

Different criteria for p-type adaptive solutions of the laminar flow field over two airfoils at Re = 10^4 (a test case for the 1st international workshop on high order methods) were implemented. The computational mesh and a snapshot of the computed velocity magnitude are shown in Fig. 4.9. Clearly, for this case a criterion base only on vorticity it will result in very high order in the near wall region while for the rest of the flow field no sufficient adaption will be performed. Therefore different criteria for adaptation must be tested.

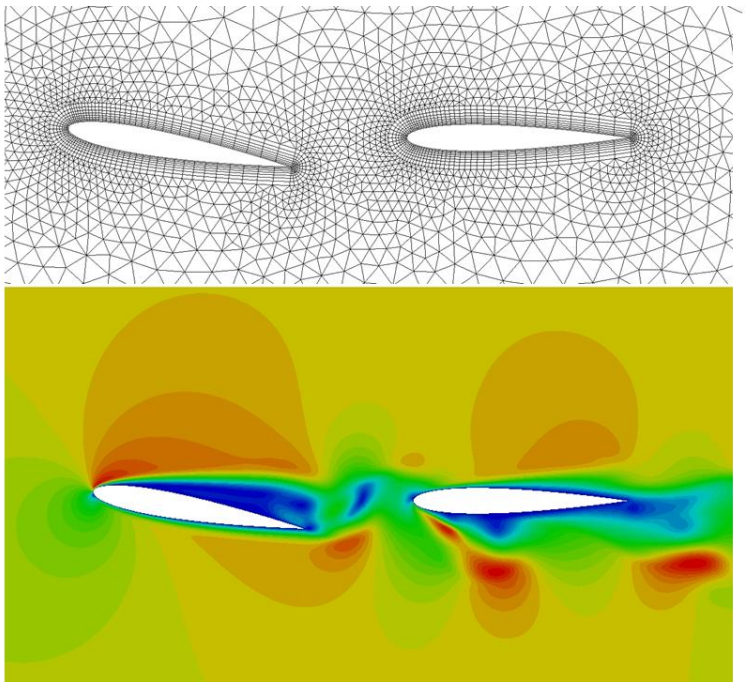


Figure 4.9. Computational mesh and snapshot of the computed velocity with a numerical solution with P4 global accuracy.

References

[1] Ekaterinaris, J. A. "High-order accurate, low numerical diffusion methods for aerodynamics," *Progress in Aerospace Sciences*, Vol. 41, No. 3-4, 2005, pp. 192-300.

[2] Cockburn, B. and Shu, C.W. "The Runge-Kutta discontinuous Galerkin method for conservation laws, multidimensional systems," *Journal of Computational Physics*, Vol. 141, 1998, pp. 199–224.

[3] Liu, Y., Vinocur, M., and Wang, Z.J. "Spectral (finite) volume method for conservation laws on unstructured grids V: Extension to three-dimensional systems," *Journal of Computational Physics*, Vol. 212, No. 2, 2006, pp. 454-472.

[4] Liu, Y., Vinokur, M. and Wang, Z.J. "Spectral difference method for unstructured grids I: Basic formulation," *J. of Comp. Physics*, Vol. 216, No. 2, 2006, pp. 780-801.

[5] Wang, Z. J., "High-order methods for the Euler and Navier–Stokes equations on unstructured grids," *Prog. in Aerospace Sciences*, Vol. 43, No. 1-3, 2007, pp. 1-41.

[6] Lv, X., Zhao, Y., Huang, X.Y., Xia, G.H., and Wang Z.J., "An efficient parallel/unstructured-multigrid preconditioned implicit method for simulating 3D unsteady compressible flows with moving objects," *Journal of Computational Physics*, Vol. 215, No. 2, 2006, pp. 661-690.

[7] L. Krivodonova, M. Berger, "High – order accurate implementation of solid wall boundaries in curved geometries," *Journal of Computational Physics*, Vol. 211, 2006, pp. 492–512.

[8] Karniadakis, G.E. and Sherwin, S. Spectral/hp Element Methods for CFD. Oxford University Press, 2nd edition, 2003.

[9] I. Toulopoulos and J. A. Ekaterinaris, "Discontinuous-galerkin discretizations for viscous flow problems in complex domains," AIAA Paper, 2005-1264.

[10] I. Toulopoulos and J. A. Ekaterinaris, "High-order discontinuous-Galerkin discretizations for computational aeroacoustics in complex domains," *AIAA Journal*, Vol. 44, No. 3, 2006, pp. 502–511.

[11] Cockburn B, Lin S.Y., Shu C.W. "TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems," *Journal of Computational Physics*, Vol. 84, No. 1, 1989, pp. 90–113.

[12] Cockburn B, Hou S, Shu CW. TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case," *Math. of Computation,* Vol. 54, 1990; pp. 545–81.

[13] Nejat, A. and Gooch C.O., "Effect of discretization order on preconditioning and convergence of a high-order unstructured Newton-GMRES solver for the Euler equations," *Journal of Computational Physics*, Vol. 227, 2008, pp. 2366–2386.

[14] Knoll, D.A, Keyes, D.E., "Jacobian-free Newton–Krylov methods: a survey of approaches and applications," *Journal of Computational Physics*, Vol. 193, 2004, pp. 357–397.

[15] Peraire, J, Nguyen, N. C., and Cockburn, B. "A Hybridizable Discontinuous Galerkin Method for the Compressible Euler and Navier-Stokes Equations, AIAA Paper 2010-363.

[16] Persson, P.O.. Peraire, J "Newton-GMRES preconditioning for discontinuous Galerkin discretizations of the Navier–Stokes equations," *SIAM J. Sci. Comput*., Vol. 30, No. 6, 2008, pp. 2709–2733.

[17] Wang, L. and Mavriplis D.J., "Implicit solution of the unsteady Euler equations for high-order accurate discontinuous Galerkin discretizations," *Journal of Computational Physics*, Vol. 225, No. 2, 2007, pp. 1994-2015.

[18] Spalart P.R., Deck S, Shur M.L., Squires K.D., Strelets M.Kh., Travin A.. "A new version of detachededdy simulation, resistant to ambiguous grid densities. *Theor. Comp. Fluid Dyn.* 20, 2006. pp. 181–95

[19] Spalart. P.R., Hedges. L., Shur. M., Travin, A., "Simulation of active flow control on a stalled airfoil," *Flow Turbul. Combustion,* 2003. pp. 71:361–73

[20] Spalart, P.R., Jou W-H, Strelets, M., Allmaras, S.R.. "Comments on the feasibility of LES for wings, and on a hybrid RANS/LES approach," In *Advances in DNS/LES*, ed. C Liu, Z Liu, 1997, pp. 137–47, Columbus, OH: Greyden Press

[21] Spalart, P.R., "Detached-Eddy Simulation," *Annual Review of Fluid Mechanics,* Vol. 41, 2009, pp. 181–202