Spring 2003

Implementation and Testing of Unsteady Reynolds-Averaged Navier-Stokes and Detached Eddy Simulation Using an Implicit Unstructured Multigrid Scheme

Juan A. Palaez
Old Dominion University

IMPLEMENTATION AND TESTING OF UNSTEADY REYNOLDS-AVERAGED NAVIER-STOKES AND DETACHED EDDY SIMULATION USING AN IMPLICIT UNSTRUCTURED MULTIGRID SCHEME

by

Juan A. Peláez
B.S. August 1996, Parks College of Saint Louis University
M.S. May 1998, Parks College of Saint Louis University

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirement for the Degree of

DOCTOR OF PHILOSOPHY

AEROSPACE ENGINEERING

OLD DOMINION UNIVERSITY
May 2003

Approved by:

Osama A. Kandil (Director)

Dimitri J. Mavriplis (Co-Director of Committee)

Colin P. Britcher (Member)

Robert L. Ash (Member)

ABSTRACT

IMPLEMENTATION AND TESTING OF UNSTEADY REYNOLDS-AVERAGED NAVIER-STOKES AND DETACHED EDDY SIMULATION USING AN IMPLICIT UNSTRUCTURED MULTIGRID SCHEME

Juan A. Peláez
Old Dominion University, 2003
Director: Dr. O. A. Kandil

Investigation and development of the Detached Eddy Simulation (DES) technique for the computation of unsteady flows on unstructured grids are presented. The motivation of the research work is driven by the ultimate goal of predicting separated flows of aerodynamic importance, such as massive stall or flows over complex non-streamlined geometries. These cases, in which large regions of massively separated flow are present, represent a challenge for conventional Unsteady Reynolds-Averaged Navier-Stokes (URANS) models, that in many cases, cannot produce solutions accurate enough and/or fast enough for industrial design and applications. A Detached Eddy Simulation model is implemented and its performance compared to the one equation Spalart-Allmaras Reynolds-Averaged Navier-Stokes (RANS) turbulence model. Validation cases using DES and URANS include decaying homogenous turbulence in a periodic domain, flow over a sphere and flow over a wing with a NACA 0012 profile, including massive stall regimes.

Because of the inherent unsteadiness of turbulence, the first step towards computing separated flows is the development of an unsteady solution technique for

unstructured meshes to be able to produce time accurate solutions. An implicit method for the computation of unsteady flows on unstructured grids was implemented based on an existing steady state multigrid unstructured mesh solver. The resulting non-linear system of equations is solved at each time step by using an agglomeration multigrid procedure. The method allows for arbitrarily large time steps and is efficient in terms of computational effort and storage. Validation of the time accurate URANS solver is performed for the well-known case of flow over a cylinder.

Co-Director of Advisory Committee: Dr. Dimitri J. Mavriplis

ACKNOWLEDGMENTS

The author would like to thank Professor Osama Kandil for his support during the last four years that made possible the culmination of this dissertation. Thanks to Dimitri Mavriplis and Manuel Salas for giving me the opportunity to conduct my investigation in a great research environment, such as ICASE, working with state of the art computational tools and resources. Particularly, I would like to express my appreciation to Dr. Mavriplis for all his patience and persistence in instructing me on the complex world of computational fluid dynamics. I also want to extend this gratitude to all the ICASE and ODU personnel that helped me and facilitated my work.

Finally, I would like to thank members of the "Big House" for their hospitality during these last two years.

TABLE OF CONTENTS

Page

LIST OF TABLES VIII

LIST OF FIGURES X

LIST OF SYMBOLS XVIII

LIST OF ABBREVIATIONS XX

INTRODUCTION 1

- HISTORICAL BACKGROUND 1
- DETACHED EDDY SIMULATION 7
- DISSERTATION OUTLINE 9

GOVERNING EQUATIONS 11

- GOVERNING EQUATIONS 11
- NAVIER-STOKES EQUATIONS 15
- REYNOLDS AVERAGED NAVIER-STOKES EQUATIONS (RANS) 19
- SPALART-ALLMARAS ONE-EQUATION MODEL 22

SOLVER DESCRIPTION 25

- SPATIAL DISCRETIZATION 25
- CONVECTIVE AND VISCOUS TERMS 27
- ARTIFICIAL DISSIPATION 32
- STEADY SOLVER 34
- MULTIGRID 38
- PARALLELIZATION 43
- UNSTEADY SOLVER IMPLEMENTATION 47

FLOW AROUND A CIRCULAR CYLINDER 50

- INTRODUCTION 50
- COMPUTATIONAL DOMAIN 51
- RESULTS 56
- CONCLUSIONS 60

DETACHED EDDY SIMULATION 62

DECAYING HOMOGENEOUS TURBULENCE 68

- DESCRIPTION OF THE CASE 68
- COMPUTATIONAL DOMAIN. 70
- INITIAL CONDITION AND BOUNDARY CONDITIONS. 71
- CASES TESTED AND RESULTS 77
- CONCLUSIONS 94

FLOW AROUND A SPHERE 95
INTRODUCTION 95
COMPUTATIONAL DOMAIN 100
STEADY RESULTS 103
UNSTEADY RESULTS 104
CONCLUSIONS 116
FLOW OVER A WING 120
INTRODUCTION 120
TEST DESCRIPTION 120
COMPUTATIONAL DOMAIN 121
RESULTS FOR STALL TESTS 125
RESULTS FOR POST-STALL TESTS (AOA=60 DEGREES) 130
CONCLUSIONS 133
FUTURE WORK AND RECOMMENDATIONS 135
ADAPTIVE ARTIFICIAL DISSIPATION 135
MESH REFINEMENT AND ADAPTIVE MESHING 141
HIGHER ORDER METHODS FOR SPATIAL AND TIME DISCRETIZATION 148
HYBRID RANS/LES METHODS USING DIFFERENT RANS TURBULENCE MODELS 148
CONCLUSIONS 149
REFERENCES 152
VITA 157

LIST OF TABLES

Page

Table 1. Status of computational aerodynamics as predicted by Chapman in 1976. [7] 4

Table 2. Summary of turbulence modeling strategies including required grid resolution and feasibility date for a simulation of flow over a vehicle as presented by Spalart in 1999 [8]. 8

Table 3. Non-dimensionalization parameters 15

Table 4. Multigrid Level Description for grid of Figures 8 through 10. 56

Table 5. Predicted Strouhal Number for Various Grid and Time Step Size 57

Table 6. Comparison of spreading rates of different free shear flows computed using the65

Table 7. Summary of converged/not converged runs of the tests performed 79

Table 8. Multigrid level description of the unstructured mesh used to compute the flow around a sphere. Mesh shown in Figure 37 and 38. 101

Table 9. Computed Steady Drag Coefficient for Flow over Sphere at Re = 250 compared with Experimental and Previous Computational Values 104

Table 10. Summary of results of the Strouhal number and averaged drag coefficient computed for the flow over a sphere at M=0.2 and Re=10^4 for different levels of artificial dissipation. 116

Table 11. Multigrid level description of the unstructured mesh used to compute the flow over a NACA 0012 airfoil. Mesh shown in Figure 58, 59, 60 and 61. 122

Table 12. Comparison of computed lift and drag coefficients obtained for AoA=60 degrees, Mach number = 0.25 and Re = 10^5 with experimental values and previous computational results. 133

Table 13. Comparison of computed lift and drag coefficients obtained for AoA=60 degrees, Mach number = 0.25 and Re = 10^5 with experimental values and previous computational results. 147

LIST OF FIGURES

Page

Figure 1. 3D Control Volume Example 27

Figure 2. Illustration of graph of block-structured jacobian matrix arising from linearization of unstructured mesh discretization 37

Figure 3. Illustration of agglomeration procedure for coarse level construction in a multigrid agglomeration algorithm. 40

Figure 4. Sample agglomerated multigrid levels used for the computation of the flow over a wing between walls 42

Figure 5. Illustration of creation of ghost points at inter-processor boundaries 45

Figure 6. Illustration of edge contraction and creation of weighted graph for mesh partitioning. Contracted line is represented in red. 46

Figure 7. Experimental Strouhal-Reynolds Number correlation for the flow over a circular cylinder. Reproduced from [20]..... 50

Figure 8. Two-dimensional side view of the computational domain for circular cylinder case. Fine grid of 631,225 points. 52

Figure 9. Two-dimensional side view of unstructured grid used for computation of flow over circular cylinder. Number of points=631,225, Wall resolution=$5x10^{-5}$ cylinder diameters. 53

Figure 10. Three-dimensional view of the computational domain for circular cylinder case. Unstructured mesh on side walls of 25,249 points per wall..... 54

Figure 11. Sample convergence rate of the density residual for one time step and 40 sub-iterations obtained on grid of Figures 8 through 10 using four multigrid levels. 55

Figure 12. Computed lift coefficient time history for the flow over a circular cylinder using three different time steps. Mach=0.2, Re=1,200 57

Figure 13. Mach contours at three different time snapshots for flow over a circular cylinder. Mach=0.2, Re=1,200 58

Figure 14. Computed mean pressure distribution over the cylinder surface compared with experimental data. Experimental data extracted from [32].....59

Figure 15. Computed mean skin friction over the cylinder surface. M=0.2, Re=1,200. ...60

Figure 16. Comparison of the length scale used by the Spalart-Allmaras turbulence model and DES for an unstructured mesh used to compute the flow over a NACA 0012 airfoil shown in Figures 58&59. 67

Figure 17. Three-dimensional view of the computational domain for the decaying homogenous turbulence case. 64x64x64 mesh shown.....70

Figure 18. Sample two-dimensional cut of the initial velocity field for the 32x32x32 node mesh.....75

Figure 19. Sample two-dimensional cut of the initial velocity field for the 64x64x64 node mesh.....76

Figure 20. Close up view of the sample two-dimensional cut of the initial velocity field for the 64x64x64 node mesh 76

Figure 21. Comparison of the computed and measured energy spectra decay. Sub-Grid-Scale model disabled. Nominal value of the artificial dissipation scaling factor. Computation performed in fine (64^3) and coarse (32^3) meshes. 80

Figure 22. Comparison of the computed and measured energy spectra decay. Sub-Grid-Scale model activated. Nominal value of the artificial dissipation scaling factor. Computation performed in fine (64^3) and coarse (32^3) meshes. 81

Figure 23. Comparison of the computed and measured energy spectra. Sub-Grid-Scale model disabled. 1/4 of the nominal value of the artificial dissipation scaling factor. Computation performed in fine (64^3) and coarse (32^3) mesh. 82

Figure 24. Comparison of the computed and measured energy spectra decay. Sub-Grid-Scale model activated. 1/4 of the nominal value of the artificial dissipation scaling factor. Computation performed in fine (64^3) and coarse (32^3) mesh. 83

Figure 25. Comparison of the computed and measured energy spectra decay. Sub-Grid-Scale model activated. 1/5 of the nominal value of the artificial dissipation scaling factor. Computation performed in fine (64^3) and coarse (32^3) mesh. 84

Figure 26. Comparison of the computed and measured energy spectra. (1/6.66) of the nominal value of the artificial dissipation scaling factor. Computation performed in coarse (32^3) mesh. 85

Figure 27. Comparison of the computed and measured energy spectra decay for different values of the C_{DES} constant. Computation performed in coarse (32^3) mesh with (1/4) of the nominal artificial dissipation scaling factor. Sub-Grid-Scale model activated. 87

Figure 28. Comparison of the computed and measured energy spectra decay for different values of the C_{DES} constant. Computation performed in fine (64^3) mesh with (1/4) of the nominal artificial dissipation scaling factor. Sub-Grid-Scale model activated. ... 88

Figure 29. Comparison of the computed and measured energy spectra decay to t=0.87 for different mesh types (Hexahedral & Prism). Computation performed in coarse (32^3)

mesh with an artificial dissipation scaling factor of 5. Sub-Grid-Scale model activated. 90

Figure 30. Comparison of the computed and measured energy spectra decay to t=2.00 for different mesh types (Hexahedral & Prism). Computation performed in coarse (32^3) mesh with an artificial dissipation scaling factor of 5. Sub-Grid-Scale model activated. 91

Figure 31. Comparison of the computed and measured energy spectra decay to t=0.87 for different mesh types (Hexahedral & Prism). Computation performed in coarse (64^3) mesh with an artificial dissipation scaling factor of 5. Sub-Grid-Scale model activated. 92

Figure 32. Comparison of the computed and measured energy spectra decay to t=2.00 for different mesh types (Hexahedral & Prism). Computation performed in coarse (64^3) mesh with an artificial dissipation scaling factor of 5. Sub-Grid-Scale model activated. 93

Figure 33. Experimental measurement of the drag coefficient of a sphere as a function of the Reynolds number. Reproduced from [36] 96

Figure 34. Experimentally measured transition angle as a function of the Reynolds number. Reproduced from [36] 97

Figure 35. Experimental measurements of the pressure on the surface of a sphere for different Reynolds numbers. Reproduced from [36]. 97

Figure 36. Experimental measurements of the skin friction on the surface of a sphere for different Reynolds numbers: –o–, $Re=1.62x10^5$; –x–, $Re=3.18x10^5$; –Δ–, $Re=5.00x10^6$. Reproduced from [36]. 98

Figure 39. Sample convergence rate of the density residual for one time step and 20 sub-iterations obtained on grid of Figure 40 and 41 using four multigrid levels..... 103

Figure 40. Time history of the force coefficients using RANS 105

Figure 41. Comparison of Mach contours at different time snapshots computed using DES (left) and URANS (right)..... 106

Figure 42. Sample pressure coefficient distribution over the surface of the..... 109

Figure 43. Sample time history of the lift and drag coefficients for nominal artificial dissipation. 110

Figure 44. Power spectrum of the streamwise drag coefficient for nominal artificial dissipation. 110

Figure 45. Mean pressure coefficient distribution over the sphere for nominal artificial dissipation. 110

Figure 46. Mean skin friction distribution over the sphere for nominal artificial dissipation. 110

Figure 47. Sample time history of the lift and drag coefficients for a fourth of the artificial dissipation nominal value 112

Figure 48. Power spectrum of the streamwise drag coefficient for a fourth of the artificial dissipation nominal value..... 112

Figure 49. Mean pressure coefficient distribution over the sphere for a fourth of the artificial dissipation nominal value. 112

Figure 50. Mean skin friction distribution over the sphere for a fourth of the artificial dissipation nominal value..... 112

Figure 51. Sample time history of the lift and drag coefficients for a fourth of the artificial dissipation nominal value and laminar flow enforced ahead of 90 degrees azimuthal. 114

Figure 52. Power spectrum of the streamwise drag coefficient for a fourth of the artificial dissipation nominal value and laminar flow enforced ahead of 90 degrees azimuthal. 114

Figure 53. Mean pressure coefficient distribution over the sphere for a fourth of the artificial dissipation nominal value and laminar flow enforced ahead of 90 degrees azimuthal. 114

Figure 54. Mean skin friction distribution over the sphere for a fourth of the artificial dissipation nominal value and laminar flow enforced ahead of 90 degrees azimuthal. 114

Figure 55. Time history of the lift and drag coefficient when transition from artificial dissipation nominal levels to ¼ of nominal levels is forced at t=100. 115

Figure 56. Summary of pressure coefficient distribution results computed for the flow over a sphere at M=0.2 and Re=10^4 for different levels of artificial dissipation. 118

Figure 57. Summary of skin friction distribution results computed for the flow over a sphere at M=0.2 and Re=10^4 for different levels of artificial dissipation. 119

Figure 58. Two-Dimensional side view of the computational domain for the NACA 0012 wing case. Unstructured grid of 1,231,667 points. 123

Figure 59. Two-Dimensioanl side view of unstructured grid used for computation of flow over NACA 0012 wing. Number of points=1,231,667, Wall resolution=1x10^{-5} wing chords. 123

Figure 60. Three-Dimensional view of the computational domain for NACA 0012 wing case. Unstructured mesh of 1,231,667 points. 124

Figure 61. Three-Dimensional detailed view of the wing-wall intersection 124

Figure 62. Sample URANS Drag Coefficient Time History for Flow over NACA 0012 Wing at various Angles of Attack. Mach number=0.25, Re=10^5. 126

Figure 63. Sample DES Drag Coefficient Time History for Flow Over NACA 0012 Wing at various Angles of Attack. Mach number=0.25, Re=10^5 126

Figure 64. Sample Mach contours computed using URANS and DES at 12 and 16 degrees angle of attack. Mach number=0.25, Re=10^5. 127

Figure 65. Comparison of Computed Lift Coefficient versus Angle of Attack for URANS and DES versus Experimental Data at two Different Reynolds Numbers 129

Figure 66. Comparison of Computed Drag Coefficient versus Angle of Attack for URANS and DES versus Experimental Data at two Different Reynolds Numbers 129

Figure 67. Lift and drag coefficient time history for 60 degrees of AoA. Computation performed on unstructured mesh of 1,231,667 nodes. Mach number = 0.25, Re = 10^5. 131

Figure 68. Sample Mach contours of the flow around a NACA 0012 at 60 degrees AoA. Computation performed on unstructured mesh of 2,107,026 nodes. Mach number = 0.25, Re = 10^5 132

Figure 69. Preliminary results for the case of decaying homogenous turbulence in a box for adaptive artificial dissipation. Computation performed in an unstructured mesh of (32x32x32) nodes. 139

Figure 70. Near-field view of original meshes and improved mesh two-dimensional cut 144

Figure 71. Far-field view of original meshes and improved mesh two-dimensional cut. 145

Figure 72. Iteration sequence of the lift and drag coefficient for a steady DES calculation of a NACA 0012 at 60 degrees AoA. Computation performed on unstructured mesh of 2,107,026 nodes. Mach number = 0.25, Re = 10^5 146

Figure 73. Iteration sequence of the residual for a steady DES calculation for a NACA 0012 at 60 degrees AoA. Computation performed on unstructured mesh of 2,107,026 nodes. Mach number = 0.25, Re = 10^5 146

LIST OF SYMBOLS

$\tilde{v}$	Eddy viscosity.
∞	Freestream condition
C_D	Drag coefficient.
C_{DES}	DES model constant.
C_f	Skin friction coefficient.
C_L	Lift coefficient.
C_p	Pressure coefficient.
C_{pb}	Back-pressure coefficient.
D_{ij}	Strain-rate tensor.
E	Total energy.
e	Internal energy per unit mass.
K	Thermal conductivity.
k	Wave number based on the number of grid points.
L_{ref}	Reference length
M	Mach Number.
M_∞	Freestream Mach Number.
p	Static pressure.
p_∞	Freestream static pressure.
q	Heat flux.
Re	Reynolds Number.

Re_∞	Freestream Reynolds Number.
St	Strouhal Number.
t	Time
vis2	Artificial dissipation scaling factor.
Γ	Ratio of specific heats.
$\Delta\tau$	Pseudotime step.
ε_m	Eddy viscosity.
μ	Viscosity.
μ_∞	Freestream viscosity.
υ	Kinematic viscosity.
υ_∞	Freestream kinematic viscosity.
ν_T	Kinematic eddy viscosity.
ρ	Density.
ρ_∞	Freestream density.
σ_{ij}	Stress tensor.
τ_{ij}	Shear stress tensor.

LIST OF ABBREVIATIONS

CFD	Computational Fluid Dynamics
DES	Detached Eddy Simulation
DNS	Direct Numerical Simulation
ICASE	Institute for Computer Applications in Science and Engineering
LaRC	Langley Research Center
LES	Large Eddy Simulation
LGS	Local-Grid-Spacing
NACA	National Advisory Committee for Aeronautics.
NASA	National Aeronautics and Space Administration.
ODU	Old Dominion University
RANS	Reynolds-Averaged Navier-Stokes
SGS	Sub-Grid-Scales
SA	Spalart-Allmaras
URANS	Unsteady Reynolds-Averaged Navier-Stokes
YPG	Young Person's Guide to Detached Eddy Simulation Grids

CHAPTER I

INTRODUCTION

Historical Background

Numerical solutions of complicated mathematical models of physical systems have been possible in the last decades based on the introduction and development of digital computers. Depending on the computational power, determined by the computational speed and storage capacity, different levels of complexity of the mathematical models have been solved.

During the last decade, the use of Computational Fluids Dynamics (CFD) in the aerospace industry has revolutionized the process of aerodynamic design. CFD has become a useful tool used extensively in aerospace applications to determine aerodynamic forces and optimize aerodynamic shapes. However, CFD has some inherent inconveniences that limit its potential in the aerospace industry. CFD has been mainly used during design iterations with the purpose of producing a better final product. The decisions made during the design iterations often involve trades between aerodynamics and added weight and/or cost with an important impact on the design in terms of time and money. If CFD is going to be involved in this crucial design-optimization process, CFD must be accurate enough to support these important trade-off decisions and fast enough to produce accurate solutions in the fast-paced product development environment

of the aerospace industry. These two issues, accuracy and time, are closely related and too much stress on trying to solve one of them can easily deteriorate the other one.

The ultimate goal of computational tools in aerospace is to be able to accurately predict flight performance over a wide range of flight envelopes. Current CFD codes are considered to be reliable for very limited regions of the flight envelope where the flow is attached and steady [1, 2, 3]. Outside this regime, the fidelity of current CFD codes deteriorates and accurate solutions are very expensive in terms of time and computer resources and, in many cases, results are unattainable. Close to the boundaries of the attached and steady flow, Navier-Stokes codes have demonstrated good capabilities of yielding accurate enough solutions but with costly time penalties. The aerospace industry recognizes CFD as a major future design tool that will considerably reduce the cost of the aircraft development cycles, but currently the strength of CFD is not to provide data but to provide understanding and to improve the design.

CFD capabilities have evolved from a technology demonstrator during the 1980's, to being capable of detailed analysis of specialized cases during the 1990's, to finally becoming a design tool in the present decade. This evolution of CFD capabilities and their role in industry has been closely coupled to the research lines explored during the last decades. Table 1 shows the expected progression of CFD capabilities predicted by Chapman in 1976. These predictions, although slightly optimistic, determined fairly well the evolution of CFD during the subsequent decades.

In the late 1970's and early 1980's, potential and Euler codes were able to resolve inviscid flows over full aircraft configurations. Efforts focused on decreasing the processing time of the algorithms by using powerful acceleration techniques for iterative solvers. The bottom line of these convergence acceleration techniques is finding the optimum balance between speed of convergence and cost of iteration. One of the most powerful acceleration techniques is the multigrid method [4]. Multigrid concepts were introduced in the 1960's by the Russian mathematicians Fedorenko and Bachvalov, but the potential of multigrid passed unrecognized until mid 1970's. The need for more efficient steady solvers rapidly increases when going from steady solvers to unsteady implicit solvers, which involve the solution of intermediate pseudo steady-state problems for each time step iteration.

By 1990, Reynolds-Averaged Navier-Stokes (RANS) methods had matured enough to solve complicated flows over complex geometries. Unsteady solvers emerged as the next logical step once considerable progress had been made in the computation of steady flows. The unsteady time scale of the problem determines what method, implicit or explicit, is the most suitable in each case. Explicit methods are used for problems in which the frequencies being considered are very high or, in other words, the time scales are very small and comparable to the grid scale. When dealing with low frequency problems, the use of explicit schemes is too restrictive and implicit methods are the desirable option.

Stage of Approximation	Readiness 2D-3D-3D Airfoils-Wings-Aircraft	Limitations	Pacing Item
Inviscid	1971 1973 1976	No flow separation	Code development
Viscous time averaged	1975 1977 1979	Accuracy of turbulence model	Turbulence modeling
Viscous time dependant	Mid 1980's	Accuracy of Navier-Stokes equations	Development of advanced computers

Table 1. Status of computational aerodynamics as predicted by Chapman in 1976. [7]

Tremendous improvements have been made in the area of solution strategies and the advances made in computer architecture and networking speeds has made possible the solution of advanced approximations of complicated flows. However, the task of generating grids about complex configurations has presented a serious challenge. The need for computing flow solutions around complex geometries opened a new area of research on unstructured mesh techniques [5]. The unstructured solvers introduced flexibility compared with structured mesh solvers for tessellating about complex geometries and for adapting around flow features, such as a shocks and boundary layers. This not only impacted the flow solution accuracy but also the overall solution methodology time since grid generation is a part of the solution process that demands considerable skills and resources. The drawback of unstructured mesh techniques, as compared to structured mesh methods, is the overhead information required to specify the mesh connectivity, which results in increased storage and CPU time requirements.

CFD is expected to become a powerful design tool in the next decades [6], capable of solving extremely challenging flows, such as massively separated flows, wake

interactions, store separation, cavities, separation onset-progression and reattachment, to name a few. To accomplish this objective, advances must be made in the area of turbulence modeling, which remains one of the major unsolved problems of classical physics. In 1996, John Lumley summarized the importance of turbulent flows and our ability to calculate them as follows "Rational design of aircraft, automobiles, nuclear reactors and all sorts of industrial mixing and forming process,... are dependant on an ability to calculate the effects of turbulent transport reliably. Unfortunately, we cannot do that. One hundred years of intense effort have brought us very good qualitative understanding of turbulent flows in nearly all practical respects, but have not brought us the ability to calculate reliably" [6].

Numerical solutions of turbulent flow cases can be achieved using different levels of approximation. The most widespread method is to solve the Reynolds averaged Navier-Stokes equations (RANS). In the RANS equations, the turbulent fluctuations appear in the Reynolds stress term which must be modeled using any of the turbulence models available in the literature. However, a common limitation of these models is their lack of generality, since the model coefficients are usually set using simple well-documented flows. In this sense, current RANS solvers are fairly successful at predicting mostly attached flows, such as a wing in cruise condition, but fail to capture a range of different off-design situations as post stall regimes, high lift configurations and non-streamlined bodies. Generally, in cases in which the RANS approach fails, the flow is characterized by large regions of separation in which a very wide range of scales are present in the flow. While the small scales tend to depend on the viscosity and, therefore,

to be universal, the large ones are affected by the boundary conditions. This is the main cause of the lack of generality of turbulence models, as it is difficult to model the effect of the large scales in the same way for many different types of flows.

Therefore, the failure to develop a universal valid turbulence model has led to new approaches such as Direct Numerical Simulation (DNS) and Large Eddy Simulation (LES). DNS is the most straightforward approach to the problem. It consists in solving the governing equations on a mesh that is fine enough to capture the smallest scales contained in the flow with a scheme designed to minimize the numerical dispersion and dissipation. The drawback is extremely high cost of the DNS computation, which is proportional to at least Re^3, where Re represents the Reynolds number. The use of DNS has grown exponentially based on expanding computer capabilities [7]. However, DNS has been limited to very simple flows and low Reynolds number cases, due to its expense.

The flow limitations of RANS and the difficulty of using DNS for realistic applied engineering problems have generated great interest in the Large Eddy Simulation approach (LES) for computing flows with large amounts of separation. Large Eddy Simulation is a compromise between Direct Numerical Simulation and Reynolds Averaged Navier-Stokes. In LES, the contribution of the large scales is computed exactly and only the smallest scales, sub-grid scales (SGS), in the flow are modeled. Because LES models the smallest scales present in the flow, the smallest cells in the grid can be much larger than the viscous range scales, and much larger time steps can be taken than

in DNS. This implies that solutions can be obtained at higher Reynolds number than using DNS or that the solution can be obtained at less expense for a given Reynolds number. However, a major difficulty of LES is that near solid surfaces, all the eddies are small and the "large" and "small" eddies tend to overlap. Therefore the required grid spacing and time step gradually evolve towards DNS as the solid boundary is approached. Using LES to resolve near wall streaks at industrial Reynolds numbers would be prohibitively expensive as was highlighted by Spalart et al. [8]. This is summarized in Table 2 reproduced from Ref. [8], which is constructed assuming a target flow over an airliner or a car and shows the number of grid points required by each method and the year in which the simulation will be feasible, as a so-called "Grand Challenge" problem. An industrial level solution capability would come later. These feasibility estimates are based on the "rule of thumb" that computer power increases by a factor of 5 every 5 years.

Detached Eddy Simulation

The Detached Eddy Simulation (DES) approach was conceived with the idea of combining the strengths of RANS methods near the solid boundaries and of LES elsewhere. The concept of DES was introduced in the literature by Spalart et al. in 1997 [9]. In that paper, the basis for a hybrid combination of LES and RANS was established. In 1999, the first application of DES was presented for a NACA 0012 airfoil at very high angles of attack [10]. Later applications of DES included flow around a cylinder [11] and flow around a sphere [12], showing very promising results for massively separated flows.

All previously mentioned work on DES was developed based on structured mesh codes. The first application of DES on an unstructured mesh code appears in a doctoral dissertation by J.R.Forsythe in 2000 [13], in which DES was applied to study supersonic flow separation for the cases of shock induced turbulent boundary layer separation and supersonic flow behind a base.

Name	Unsteady technique	Number of grid points	Year
2DURANS	YES	10^5	1980
3DRANS	NO	10^7	1985
3DURANS	YES	10^7	1995
DES	YES	10^8	2000
LES	YES	10^{15}	2045
DNS	YES	10^{16}	2080

Table 2. Summary of turbulence modeling strategies including required grid resolution and feasibility date for a simulation of flow over a vehicle as presented by Spalart in 1999 [8].

Later applications of DES in unstructured solvers include its implementation in the unstructured mesh code Cobalt60 under a U. S. Department of Defense Challenge Project titled "Multidisciplinary Applications of Detached-Eddy Simulations of Separated Flows at High Reynolds Numbers". The final goal of this project is the computation of the flow around a complete aircraft in massively separated flow regimes. Preliminary results of this work have been published in January of 2002 [14, 15, 16, 17, 18] and include vortical flows over delta wings, flow over a full aircraft configuration (F16, F18A/E, C-130, X38), flow over a rounded square and flow over a prolate

spheroid. This research is still in progress and more results are expected to be presented in 2003.

The ultimate goal of this dissertation work was to develop a Large Eddy Simulation capability based on an existing unstructured grid Navier-Stokes solver in order to perform detached eddy simulations combining RANS near the walls and LES in massively separated regions in a non zonal manner. DES is implemented in a second-order accurate parallel-unstructured mesh code and tested on cases previously solved using structured mesh codes to study its feasibility.

Dissertation Outline

The outline of the dissertation is the following. Chapter II presents an overview of the governing equations. Chapter III describes the numerical discretization and solver scheme of the steady multigrid unstructured mesh solver. Because of the inherent unsteadiness associated with the massively separated regions, the first step in developing a large eddy simulation capability involves the extension of the currently existing parallel-unstructured multigrid steady-state Reynolds-averaged Navier-Stokes solver to an unsteady Reynolds-averaged Navier-Stokes flow solver. This is presented in the second part of Chapter III. In Chapter IV, the unsteady Reynolds-averaged Navier-Stokes (URANS) solver is tested using the well-known case of the flow over a circular cylinder. The DES implementation on the parallel-unstructured mesh unsteady Reynolds-averaged Navier-Stokes solver is demonstrated in Chapter V. The DES capabilities of the solver and the artificial dissipation effects are assessed using a case of

decaying isotropic turbulence in a periodic domain which is presented in Chapter VI. In Chapter VII, DES and URANS are compared using the case of flow over a sphere. In addition, DES and URANS are compared again in Chapter VIII using a case of aeronautical interest, such as, the flow over a NACA 0012 airfoil. Finally, in Chapter XIX, some ideas are presented as a basis for future research and in Chapter X the conclusions of the investigation are stated.

CHAPTER II

GOVERNING EQUATIONS

Governing Equations

The governing equations for fluid flow will be derived invoking the physical laws of conservation of mass, momentum and energy. The starting point for any conservative equation will be the Reynolds Transport Theorem that will transform material derivatives of volume integrals into volume integrals of Eulerian derivatives.

Let α be any specific property of a fluid such as density, momentum or specific energy. The Reynolds' Transport Theorem states that the total rate of change of the integral of α(t) over an arbitrary material volume will be equal to the time derivative of α(t) inside a coinciding control volume plus the flux of α(t) across the control surface enclosing the control volume. The mathematical expression of this theorem is shown in equation (1).

$$\frac{D}{Dt}\int_{\tilde{V}(t)}\alpha(t)d\tilde{V} = \int_{V(t)}\frac{\partial\alpha}{\partial t}dV + \int_{S(t)}\alpha(t)\vec{u}\cdot\hat{n}dS \tag{1}$$

Using Gauss' Theorem to convert the surface integral into a volume integral, equation (1) transforms into

$$\frac{D}{Dt}\int_{\tilde{V}(t)}\alpha(t)d\tilde{V} = \int_{V(t)}\left[\frac{\partial\alpha}{\partial t}+\frac{\partial}{\partial x_k}(\alpha u_k)\right]dV \tag{2}$$

Conservation of mass

Consider a specific mass of arbitrary shape and volume. By observing this material volume as it flows, it is seen that its volume and shape may change, but its total mass will remain constant. Mathematically this is equivalent to saying that the Lagrangian derivative of the mass contained in the material volume is equal to zero.

$$\frac{D}{Dt}\int_{\tilde{V}(t)} \rho(t)\, d\tilde{V} = 0 \tag{3}$$

In this case the specific fluid property referred before as α, is the mass density, ρ.

Applying the Reynolds' Transport Theorem to equation (3)

$$\int_{V(t)} \left[\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_k}(\rho u_k)\right] dV = 0 \tag{4}$$

Since the volume is arbitrarily chosen the only way that equation (4) can be satisfied for any shape and volume will be if the integrand is equal to zero.

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_k}(\rho u_k) = 0 \tag{5}$$

This is the conservative form of the continuity equation that will be used to force the conservation of mass in the flow field.

Conservation of momentum

The conservation of momentum equation is based on the application of the Newton's second law of motion to an element of fluid. This states that the rate of change of the linear momentum of a given mass of fluid is proportional to the net external force acting on the mass. The external forces acting on the fluid will be classified as body forces, such as gravitational forces, and surface forces, such as pressure and viscous-shear stresses. The mathematical expression of the above statement is

$$\frac{D}{Dt}\int_{\tilde{V}(t)} \rho u_i \, d\tilde{V} = \int_S Q_i \, dS + \int_V \rho f_i dV \tag{6}$$

The first term on the right hand side represents the surface forces while the second term represents the body forces. Considering that the surface forces are fully represented by the stress tensor and applying the Reynolds Transport Theorem to the expression above, yields

$$\int_V \left[\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_k}(\rho u_i u_k)\right] dV = \int_S \sigma_{ij} n_i \, dS + \int_V \rho f_i dV \tag{7}$$

Applying Gauss Theorem to transform surface integrals into volume integrals

$$\int_V \left[\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_k}(\rho u_i u_k)\right] dV = \int_V \frac{\partial}{\partial x_j}\sigma_{ij} dV + \int_V \rho f_i dV \tag{8}$$

Therefore

$$\frac{\partial}{\partial t}(\rho u_i)+\frac{\partial}{\partial x_k}(\rho u_i u_k)=\frac{\partial}{\partial x_j}\sigma_{ij}+\rho f_i \tag{9}$$

where the stress tensor is taken to be of the form

$$\sigma_{ij}=-p\delta_{ij}+\tau_{ij} \tag{10}$$

Conservation of energy

The conservation of energy is based on the First Law of Thermodynamics, which states that the rate of change of the total energy is equal to the rate at which work is being done plus the rate at which heat is being added.

$$\frac{D}{Dt}\int_{\tilde{V}(t)}(\rho E)d\tilde{V}=\int_S u_i\cdot Q_i\,dS+\int_V u_i\cdot\rho f_i dV-\int_S q_i\cdot n_i\,dS \tag{11}$$

where E represents the total energy per unit mass and q the conductive heat flux leaving the control volume. The total energy per unit mass is given by

$$E=\left(e+\frac{1}{2}u_i\cdot u_i\right)\equiv Total\ \ Energy \tag{12}$$

As in the previous case, equation (11) can be transformed into

$$\int_V\frac{\partial}{\partial t}(\rho E)+\frac{\partial}{\partial x_i}(\rho E u_i)dV=\int_S u_i\cdot\sigma_{ki}\cdot n_k\,dS+\int_V u_i\cdot\rho f_i dV-\int_S q_i\cdot n_i\,dS \tag{13}$$

$$\int_V \frac{\partial}{\partial t}(\rho E) + \frac{\partial}{\partial x_i}(\rho E u_i) dV = \int_V \frac{\partial}{\partial x_k}(u_i \cdot \sigma_{ki}) dV + \int_V u_i \cdot \rho f_i dV - \int_V \frac{\partial}{\partial x_i}(q_i) dV \quad (14)$$

Therefore

$$\frac{\partial}{\partial t}(\rho E) + \frac{\partial}{\partial x_i}(\rho E u_i) = \frac{\partial}{\partial x_k}(u_i \cdot \sigma_{ki}) + u_i \cdot \rho f_i - \frac{\partial}{\partial x_i}(q_i) \quad (15)$$

where the stress tensor is given by equation (16)

$$\sigma_{ij} = -p\delta_{ij} + \tau_{ij} \quad (16)$$

Navier-Stokes equations

The governing equations described above are non-dimensionalised using the following reference parameters (Table 3) to obtain the dimensionless governing equations (the ∞ subscript designates freestream conditions).

Length	L_{ref} (determined in the input file)
Velocity	$\sqrt{p_\infty / \rho_\infty}$
Time	$\frac{L_{ref}}{\sqrt{p_\infty / \rho_\infty}}$
Density	ρ_∞
Viscosity	μ_∞

Table 3. Non-dimensionalization parameters.

Starting from equation (9), and dividing and multiplying by the corresponding reference parameters, the right-hand-side (RHS) and the left-hand-side (LHS) of equation (9) are multiplied by the following parameters:

$$\frac{\rho_\infty \cdot \sqrt{p_\infty/\rho_\infty}}{\frac{L_{REF}}{\sqrt{p_\infty/\rho_\infty}}} \cdot LHS = \frac{p_\infty}{L_{REF}} \cdot LHS = \frac{1}{L_{REF}} \cdot \mu_\infty \cdot \sqrt{p_\infty/\rho_\infty} \cdot \frac{1}{L_{REF}} \cdot RHS = \frac{\mu_\infty}{L_{REF}^2} \cdot \sqrt{p_\infty/\rho_\infty} \cdot RHS \tag{17}$$

Taking all the parameters to the right-hand-side

$$LHS = \left\{ \frac{\mu_\infty}{L_{REF}^2} \cdot \sqrt{p_\infty/\rho_\infty} \right\} \cdot \left\{ \frac{L_{REF}}{p_\infty} \right\} \cdot RHS = \frac{\mu_\infty}{\rho_\infty \cdot L_{REF} \cdot \sqrt{p_\infty/\rho_\infty}} \cdot RHS = \frac{\sqrt{\gamma} \cdot M_\infty}{\mathrm{Re}_\infty} \cdot RHS \tag{18}$$

Therefore, the conservative form of the dimensionless, unsteady, compressible Navier-Stokes equations in matrix form is given by

$$\frac{\partial w}{\partial t} + \frac{\partial f_c}{\partial x} + \frac{\partial g_c}{\partial y} + \frac{\partial h_c}{\partial z} = \frac{\sqrt{\gamma} \cdot M_\infty}{\mathrm{Re}_\infty} \cdot \left[\frac{\partial f_v}{\partial x} + \frac{\partial g_v}{\partial y} + \frac{\partial h_v}{\partial z} \right] \tag{19}$$

where w is the solution vector and f_c, g_c and h_c are the Cartesian components of the convective fluxes

$$w=\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ \rho E\end{bmatrix} \quad f_c=\begin{bmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ \rho uw\\ \rho uE+up\end{bmatrix} \quad g_c=\begin{bmatrix}\rho v\\ \rho vu\\ \rho v^2+p\\ \rho vw\\ \rho vE+vp\end{bmatrix} \quad h_c=\begin{bmatrix}\rho w\\ \rho wu\\ \rho wv\\ \rho w^2+p\\ \rho wE+wp\end{bmatrix} \tag{20}$$

In equation (19), M_∞ is the freestream Mach Number, Re_∞ is the Reynolds Number based on a characteristic length, ρ is the fluid density, u, v and w are the Cartesian velocity components in the x, y and z direction respectively, E is the total energy, and p is the pressure which can be calculated from the equation of state of a perfect gas

$$p=(\gamma-1)\cdot\rho\cdot\left[E-\frac{\left(u^2+v^2+w^2\right)}{2}\right] \tag{21}$$

The viscous fluxes f_v, g_v and h_v are given by

$$f_v=\begin{bmatrix}0\\ \tau_{xx}\\ \tau_{xy}\\ \tau_{xz}\\ u\cdot\tau_{xx}+v\cdot\tau_{xy}+w\cdot\tau_{xz}-q_x\end{bmatrix} \tag{22}$$

$$g_v=\begin{bmatrix}0\\ \tau_{yx}\\ \tau_{yy}\\ \tau_{yz}\\ u\cdot\tau_{yx}+v\cdot\tau_{yy}+w\cdot\tau_{yz}-q_y\end{bmatrix} \tag{23}$$

$$h_v = \begin{bmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ u \cdot \tau_{zx} + v \cdot \tau_{zy} + w \cdot \tau_{zz} - q_z \end{bmatrix} \tag{24}$$

where τ represents the shear stress tensor, and q the heat flux vector, which are given by the constitutive equation for a Newtonian fluid under the Stokes hypothesis and Fourier's Law for heat conduction, respectively

$$\begin{aligned}
\tau_{xx} &= 2\mu \frac{\partial u}{\partial x} - \frac{2}{3}\mu\left(\vec{\nabla} \cdot \vec{V}\right) \\
\tau_{yy} &= 2\mu \frac{\partial v}{\partial y} - \frac{2}{3}\mu\left(\vec{\nabla} \cdot \vec{V}\right) \\
\tau_{zz} &= 2\mu \frac{\partial w}{\partial z} - \frac{2}{3}\mu\left(\vec{\nabla} \cdot \vec{V}\right) \\
\tau_{xy} &= \tau_{yx} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \\
\tau_{xz} &= \tau_{zx} = \mu\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) \\
\tau_{yz} &= \tau_{zy} = \mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right) \\
q_x &= -k\frac{\partial T}{\partial x} = -\frac{\gamma}{\gamma - 1} \cdot \frac{\mu}{\mathrm{Pr}} \cdot \frac{\partial\left(\frac{p}{\rho}\right)}{\partial x} \\
q_y &= -k\frac{\partial T}{\partial y} = -\frac{\gamma}{\gamma - 1} \cdot \frac{\mu}{\mathrm{Pr}} \cdot \frac{\partial\left(\frac{p}{\rho}\right)}{\partial y} \\
q_z &= -k\frac{\partial T}{\partial z} = -\frac{\gamma}{\gamma - 1} \cdot \frac{\mu}{\mathrm{Pr}} \cdot \frac{\partial\left(\frac{p}{\rho}\right)}{\partial z}
\end{aligned} \tag{25}$$

γ is the ratio of specific heats of the fluid and Pr is the Prandtl number. The coefficient of viscosity is determined from Sutherland's law

$$\mu = C1 \cdot T^{0.72} \tag{26}$$

where C1 is a constant.

Reynolds Averaged Navier-Stokes Equations (RANS)

Reynolds time averaging is used with the Navier-Stokes equations to account for stationary turbulence, where a stationary turbulence is defined as turbulent flow that does not vary with time on the average. For such flow, we define each flow variable as the sum of a mean and a fluctuating part such that

$$F(x) = \bar{f}(x) + f'(x,t) \tag{27}$$

In equation (27) the first term on the right hand-side is the time-averaged value, or mean value, defined by

$$\bar{f}(x) = \lim_{Time \to \infty} \int_{t}^{t+Time} f(x,t)\, dt \tag{28}$$

The time average of the mean value is the same as the time-averaged value,

$$\bar{\bar{f}}(x) = \lim_{Time \to \infty} \int_{t}^{t+Time} \bar{f}(x)\, dt = \bar{f}(x) \tag{29}$$

The time average of the fluctuating part is zero by definition. Equation (30) illustrates the rules of time averaging.

$$\left.\begin{aligned} A &= \bar{a} + a' \\ B &= \bar{b} + b' \end{aligned}\right\} \Rightarrow \left\{\begin{aligned} \overline{A} &= \overline{\bar{a}} + \overline{a'} = \bar{a} \\ \overline{B} &= \overline{\bar{b}} + \overline{b'} = \bar{b} \\ \overline{A \cdot B} &= \overline{(\bar{a} + a') \cdot (\bar{b} + b')} = \overline{\bar{a} \cdot \bar{b}} + \overline{a' \cdot b'} \end{aligned}\right. \tag{30}$$

Notice how the time average of the product of the fluctuating parts in the last expression of equation (30) is not zero. This will have important consequences when substituting in the Navier-Stokes equations as will be shown in the paragraph below.

Substituting the flow variables in the Navier-Stokes equations as the mean value and the perturbation, as defined in equation (27), and averaging in time, the following expression is obtained for the momentum conservation equation

$$\frac{\partial}{\partial t}\left(\rho \bar{u}_i\right) + \frac{\partial}{\partial x_k}\left(\rho \bar{u}_i \bar{u}_k + \rho \overline{u'_j u'_i}\right) = \frac{\partial}{\partial x_j}\sigma_{ij} + \rho f_i \tag{31}$$

where $\bar{u}$ represents the mean value of the velocity, u' the perturbation velocity of zero average in time and the correlation $[\overline{u'_j u'_i}]$ the time-averaged rate of momentum transfer due to the turbulence. Note that equation (31) is obtained by introducing density-weighted velocities defined in equation (32) as suggested by Favre [19].

$$\bar{u}_i = \frac{1}{\rho} \lim_{Time \to \infty} \frac{1}{T} \int_{t}^{t+Time} \rho \cdot u_i \, dt \tag{32}$$

The $\overline{u_j' u_i'}$ correlation term in equation (31) is the real difficulty of the RANS equations since in order to compute all mean flow properties of the flow, a model of this term required for closure. This quantity is known as the Reynolds-stress tensor and, as can be observed, it consists of six new unknowns that must be modeled. Without going any further, it should be pointed out that the function of the turbulence model is to prescribe the unknown correlation terms based on known quantities to make the whole system solvable. But it must be stressed that the averaging process is merely a mathematical process in nature that does not introduce additional physics information to the problem. As expressed by Wilcox [24], "in essence Reynolds averaging is a brutal simplification that loses much of the information contained in the Navier-Stokes equation".

Boussinesq Approximation

By manipulating the conservation of momentum equation (31) from the conservative form to the non-conservative form, the following expression is obtained

$$\frac{\partial}{\partial t}(\rho u_i) + \rho u_k \frac{\partial}{\partial x_k}(u_i) = -\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\mu \frac{\partial u_i}{\partial x_j} - \rho \overline{u_i u_j}\right) + \rho f_i \qquad (33)$$

The two terms in the parenthesis in the right-hand side of equation (33) represent the stresses produced by the viscosity effects and the stresses produced by the turbulence effects. In 1877, Boussinesq ended up assuming that turbulence stresses act like the viscous stresses in the sense that they are directly proportional to the velocity gradient.

This is known as the Boussinesq approximation, which introduces the concept of eddy viscosity (ε_m) in the stress tensor expression as

$$\sigma_{ij} = -p\delta_{ij} + (\mu + \rho\varepsilon_m)\overline{D_{ij}} \tag{34}$$

where

$$\rho\varepsilon_m \overline{D_{ij}} = -\rho\overline{u_i u_j} = \rho\varepsilon_m \left(\frac{\partial \overline{U_i}}{\partial x_j} + \frac{\partial \overline{U_j}}{\partial x_i} \right) \tag{35}$$

The Boussinesq approximation assumes that the principal axes of the Reynolds-stress tensor are coincident with the principal axes of the mean strain-rate tensor at all points in the flow. The coefficient of proportionality is the eddy viscosity. This approximation reduces the number of unknowns from six to one and, although it provides accurate predictions for many flows, it also shows important deficiencies in flows with sudden changes in the mean strain rate, such as separated flows. The majority of RANS turbulence models are based on the Boussinesq approximation and attempt to model the eddy viscosity with an algebraic equation or one or two partial differential equations in the flow field.

Spalart-Allmaras one-equation model

The one-equation turbulence model of Spalart-Allmaras presents a transport equation for the turbulent viscosity assembled using empiricism and arguments of

dimensional analysis. The Spalart-Allmaras model solves a single field equation for a variable related to the eddy viscosity through the kinematic eddy viscosity

$$\nu_t = \tilde{\nu} f_{v1} \tag{36}$$

The Spalart-Allmaras model can be expressed in dimensionless form as:

Eddy viscosity equation

$$\frac{\partial \tilde{\nu}}{\partial t} + u_j \frac{\partial \tilde{\nu}}{\partial x_j} = C_{b1}[1 - f_{t2}]\Omega \tilde{\nu} + \frac{M_\infty}{\mathrm{Re}}\left\{C_{b1}[(1-f_{t2})f_{v2}]\frac{1}{\kappa^2} - C_{w1} f_{w1}\right\}\left(\frac{\tilde{\nu}}{d}\right)^2 - \frac{M_\infty}{\mathrm{Re}}\frac{C_{b2}}{\sigma}\tilde{\nu}\frac{\partial^2 \tilde{\nu}}{\partial x_j^2} + \frac{M_\infty}{\mathrm{Re}}\frac{1}{\sigma}\frac{\partial}{\partial x_j}\left[(\nu + (1+C_{b2})\tilde{\nu})\frac{\partial \tilde{\nu}}{\partial x_j}\right] \tag{37}$$

with the auxiliary relations

$$f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3} \quad ; \quad f_{v2} = \frac{1}{\left(1 + \frac{\chi}{C_{v2}}\right)^3}$$

$$\chi = \frac{\tilde{\nu}}{\nu}$$

$$f_{t2} = C_{t3} e^{-C_{t4}\chi^2} \quad ; \quad f_w = g\left[\frac{1 + C_{w3}^6}{g^6 + C_{w3}^6}\right]^{1/6} \tag{38}$$

$$g = r + C_{w2}(r^6 - r) \quad ; \quad r = \frac{\tilde{\nu}}{\tilde{S}\left(\frac{\mathrm{Re}}{M_\infty}\right)\kappa^2 d^2}$$

$$\tilde{S} = f_{v3}\Omega + \frac{\tilde{\nu} f_{v2}}{\left(\frac{\mathrm{Re}}{M_\infty}\right)\kappa^2 d^2} \quad ; \quad f_{v3} = \frac{(1+\chi f_{v1})(1-f_{v2})}{\chi}$$

and the closure coefficients

$$C_{b1}=0.1355 \qquad C_{b2}=0.622 \qquad C_{v1}=7.1 \qquad C_{v2}=5.0 \qquad \sigma=\frac{2}{3}$$
$$C_{w2}=0.3 \qquad C_{w3}=2.0 \qquad C_{t3}=1.2 \qquad C_{t4}=0.5 \qquad \kappa=0.41 \qquad (39)$$
$$C_{w1}=\frac{C_{b1}}{\kappa}+\frac{(1+C_{b2})}{\sigma}$$

After testing the Spalart-Allmaras turbulence model for cases of a far wake, mixing layer, plane jet flow, round jet flow, radial jet flow and boundary layer with different favorable and adverse pressure gradient, Wilcox concluded [24] that "on balance, Spalart-Allmaras predictions are satisfactory for many engineering applications. It is especially attractive for airfoil and wing applications, for which it has been calibrated. Its failure to reproduce jet-spreading rates is a cause for concern and should serve as a warning that the model has some shortcomings. Nevertheless, the model appears to be a valuable engineering tool."

CHAPTER III

SOLVER DESCRIPTION

Spatial Discretization

The governing equations are discretized by using finite-volume techniques in meshes of mixed element types that may include tetrahedra, pyramids, prisms and hexahedra. The conserved flow variables are stored at the vertices of the mesh and all elements of the grid are handled by a unifying edge-based data-structure, which is more compact in terms of memory overhead, and minimizes the amount of gather-scatter required on parallel computer architectures.

The solver is based on a single unifying edge-based data structure. However, to get to this edge-based data structure, a pre-processing of the original mesh is performed based on a data-structure containing a list of elements (tetrahedra, pyramids, prisms and hexahedra) and a list of nodes identifying the vertices that constitute each element. Each node will be spatially specified by its Cartesian co-ordinates. The control volume for each vertex is constructed connecting the centroids of all the cells that contain the specified node.

Based on the definition of the control volumes, it can be observed that the edges of the original mesh are associated with the faces of the control volumes surrounding the nodes. Therefore, the convective and viscous fluxes for each node can be computed

"along" the edges coming out of each node. Note that the fluxes are not computed in the direction of the edges but in the direction of the normal to the control surfaces forming the control volume around the node and then are assembled using the edge-based data structure. This implies that a list of edges will be computed, and for each edge, the address of the two end points will be stored as well as three coefficients, which represent the x, y and z components of the normal to the face associated with that edge. The magnitude of the normal carries the information of the area of the control surface.

The calculation of the coefficients associated with the edges will be illustrated for the case of a tetrahedral mesh. For any other element, different than tetrahedral, the approach is the same with the corresponding geometrical differences. In three dimensions, the face associated with each edge will be the contribution to that face of all the tetrahedral cells sharing that edge. As shown in Figure 1, the face of the control volume associated with the edge ab will be formed by the triangles m-t3-c2, m-t2-c2, m-t2-c1, m-t1-c1. As can be observed, the triangles m-t3-c2 and m-t2-c2 correspond to the tetrahedral abGH, while the triangles m-t2-c1, m-t1-c1 correspond to the contribution of the tetrahedral abHF. The coefficients associated with the edge ab will contain the information corresponding to the area of the associated face and the direction of the normal to the face that will be computed as the vectorial sum of all the area vectors of all the contributing triangles. Note that similar to the tetrahedrals abGH and abHF, more tetrahedra will contribute to the edge ab until a face completely surrounding the edge is obtained.

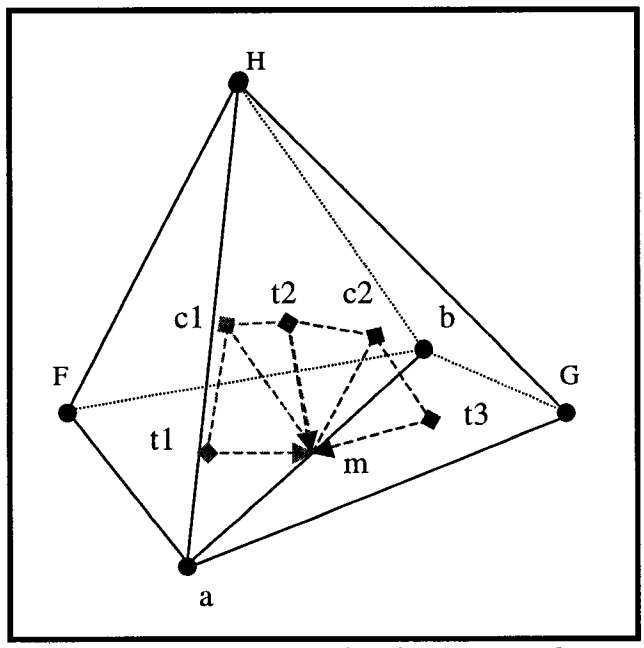

Figure 1. 3D Control Volume Example

Convective and viscous terms

Convective fluxes

The convective terms in the Navier-Stokes Equations are given in equation (19) of Chapter I, which are:

$$\text{Convective Mass Flux - } \int_V \left[\frac{\partial}{\partial x_k} (\rho u_k) \right] dV$$

$$\text{Convective Momentum Flux - } \int_V \left[\frac{\partial}{\partial x_k} (\rho u_j u_k) + \frac{\partial p}{\partial x_j} \right] dV \qquad (40)$$

$$\text{Convective Energy Flux - } \int_V \left[\frac{\partial}{\partial x_k} (e u_k) + p \frac{\partial u_k}{\partial x_k} \right] dV$$

The Gauss Theorem relates the surface integrals to volume integrals by the expression

$$\int_{S(t)} \alpha(t) u \cdot n \, dS = \int_{V(t)} \nabla \cdot (\alpha u) dV \tag{41}$$

The convective fluxes will be computed as surface integrals based on the Gauss Theorem as:

$$\text{Convective Mass Flux - } \int_S (\rho u_k \cdot n_k) dS$$

$$\text{Convective Momentum Flux - } \int_S \left[\rho u_j u_k \cdot \hat{n}_k + p \hat{n}_j \right] dS \tag{42}$$

$$\text{Convective Energy Flux - } \int_S [e u_k \cdot \hat{n}_k + p u_k \cdot \hat{n}_k] dS$$

Therefore, the discretized expression of the convective fluxes along the edge connecting to nodes (n1) and (n2) will be of the form:

$$qs1 = \left(n_x \cdot v_x^1 + n_y \cdot v_y^1 + n_z \cdot v_z^1\right)$$
$$qs2 = \left(n_x \cdot v_x^2 + n_y \cdot v_y^2 + n_z \cdot v_z^2\right)$$
$$Convective\_Mass\_Flux = \frac{(qs1 \cdot \rho_1 + qs2 \cdot \rho_2)}{2}$$
$$Convective\_Momentum\_Flux\_x = \frac{\left(qs1 \cdot v_x^1 + qs2 \cdot v_x^2\right)}{2} + \frac{(p_1 + p_2)}{2} \cdot n_x \tag{43}$$
$$Convective\_Momentum\_Flux\_y = \frac{\left(qs1 \cdot v_y^1 + qs2 \cdot v_y^2\right)}{2} + \frac{(p_1 + p_2)}{2} \cdot n_y$$
$$Convective\_Momentum\_Flux\_z = \frac{\left(qs1 \cdot v_z^1 + qs2 \cdot v_z^2\right)}{2} + \frac{(p_1 + p_2)}{2} \cdot n_z$$
$$Convective\_Energy\_Flux = qs1 \cdot \frac{(\rho_1 + p_1)}{2} + qs2 \cdot \frac{(\rho_2 + p_2)}{2}$$

Where qs1 and qs2 represent the scalar product of the velocity at each node times the surface vector of the face crossed by that edge. The superscript indicates the node (n1) or (n2), and the subscript the vector component x, y or z.

The viscous fluxes

The viscous terms in the Navier-Stokes equations, assuming a thin layer approximation in all directions, are given in equations 22, 23 and 24 of Chapter II as:

Viscous Mass flux – There is no viscous fluxes in the conservation of mass equation.

$$\text{Viscous Momentum flux – x component } \int_V \mu\left[\frac{4}{3}\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}\right]dV$$

$$\text{Viscous Momentum flux – y component } \int_V \mu\left[\frac{\partial^2 v}{\partial x^2}+\frac{4}{3}\frac{\partial^2 v}{\partial y^2}+\frac{\partial^2 v}{\partial z^2}\right]dV \qquad (44)$$

$$\text{Viscous Momentum flux – z component } \int_V \mu\left[\frac{\partial^2 w}{\partial x^2}+\frac{\partial^2 w}{\partial y^2}+\frac{4}{3}\frac{\partial^2 w}{\partial z^2}\right]dV$$

Viscous Energy flux –

$$\int_V \left\{ \begin{array}{l} \mu\cdot u\cdot\left[\frac{4}{3}\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}\right]+\mu\cdot v\cdot\left[\frac{\partial^2 v}{\partial x^2}+\frac{4}{3}\frac{\partial^2 v}{\partial y^2}+\frac{\partial^2 v}{\partial z^2}\right]+\mu\cdot w\cdot\left[\frac{\partial^2 w}{\partial x^2}+\frac{\partial^2 w}{\partial y^2}+\frac{4}{3}\frac{\partial^2 w}{\partial z^2}\right] \\ -\frac{\partial^2 q}{\partial x^2}-\frac{\partial^2 q}{\partial y^2}-\frac{\partial^2 q}{\partial z^2} \end{array} \right\} dV$$

The final discrete viscous terms obtained in this manner form a nearest neighbor stencil. The viscous terms for a vertex, i, depend only on values at i and at vertex k, such that k is joined to i by a mesh edge. In three dimensions, for the full Navier-Stokes equations, this would require the storage of nine coefficients per edge. However, the local edge-based coefficient matrix is symmetric about the diagonal [30]. Thus, only six coefficients per edge are required for the discretization of the viscous terms. Neglecting the cross derivative terms, the number of coefficients can be reduced to three per edge. Finally, note that by adopting the thin-layer form of the Navier-Stokes equations, only a single coefficient per edge is required to compute the discretization of the viscous terms as the discretization of a Laplacian. The viscous fluxes will be computed based on the Gauss Theorem as:

Viscous Mass flux – There is no viscous fluxes in the conservation of mass equation.

$$\text{Viscous Momentum flux} - \int_S \mu \cdot \alpha \cdot \frac{\partial V_i}{\partial x_i} \cdot \hat{n} \cdot dS = \sum_{k=1}^{n} \mu \cdot \alpha \cdot \left[\frac{u(n1)_k - u(n2)_k}{\Delta_k} \right] \cdot n_k \qquad (45)$$

$$\text{Viscous Energy flux} - \int_S \left\{ \cdot \alpha \cdot \mu \cdot u_k \cdot \left[\frac{u(n1)_k - u(n2)_k}{\Delta_k} \right] - \left[\frac{p(n1)_k - p(n2)_k}{\Delta_k} \right] \right\} \cdot n_k \cdot dS$$

where α represents the viscous edge coefficient for the viscous flux discretization. Note that the viscosity is considered to be locally constant to allow the calculation of the edge coefficients in a pre-processing phase.

Therefore, the discretized expression of the viscous fluxes along the edge connecting to nodes (n1) and (n2) will be of the form:

$$ed\_aver = \frac{[ed(n1) + ed(n2)]}{2}$$
$$visc\_aver = \frac{[visc(n1) + visc(n2)]}{2}$$
$$u\_aver = \frac{[u(n1) + u(n2)]}{2}$$
$$v\_aver = \frac{[v(n1) + v(n2)]}{2}$$
$$w\_aver = \frac{[w(n1) + w(n2)]}{2}$$
$$\mu = ed\_aver + visc\_aver$$

$$Viscous\_Momentum\_Flux\_x = \mu \cdot \alpha \cdot [u(n2) - u(n1)]$$
$$Viscous\_Momentum\_Flux\_y = \mu \cdot \alpha \cdot [v(n2) - v(n1)]$$
$$Viscous\_Momentum\_Flux\_z = \mu \cdot \alpha \cdot [w(n2) - w(n1)]$$

$$\begin{aligned} Viscous\_Energy\_Flux = \; & Viscous\_Momentum\_Flux\_x \cdot u\_aver + \\ & Viscous\_Momentum\_Flux\_y \cdot v\_aver + \\ & Viscous\_Momentum\_Flux\_z \cdot w\_aver + \\ & rkv \cdot \alpha \cdot [p(n2) \cdot \rho(n2) - p(n1) \cdot \rho(n1)] \end{aligned} \quad (46)$$

Where ed_aver, visc_aver, u_aver, v_aver and w_aver represent respectively the average eddy viscosity, the average physical viscosity, the average velocity in the x-component, the average velocity in the y-component and the average velocity in the z-component between the values in node n1 and node n2. The term μ_aver is the total average viscosity defined as the sum of the average eddy viscosity and the average physical viscosity. The α term in equation (46) is the viscous edge coefficient defined as:

$$\alpha = \frac{\hat{l}_{1-2} \cdot \hat{n}}{\left\| \hat{l}_{1-2} \right\|} \tag{47}$$

where $\hat{l}_{1-2}$ represents the vector joining the two edge end points, and $\hat{n}$ is the face normal associated with the edge.

Artificial dissipation

Von Neuman and Richtmeyer introduced the concept of artificial dissipation to mitigate the problems of second-order three-point schemes associated with instabilities and oscillations of the solution in regions of large gradients. The concept is to add terms to the scheme to simulate viscosity on the scale of the mesh. These added terms act as numerical viscosity that damps high frequency oscillations. Furthermore, these additional dissipative terms must be carefully constructed to ensure that the accuracy of the scheme is preserved in the inviscid region where convective terms dominate, as well as in the boundary layer and wake region, where the artificial viscosity must be much smaller than the physical viscous terms.

The artificial dissipation operator is formulated as a global undivided Laplacian operating on a blend of the flow variables and their second differences:

$$D(w) = \Omega \lambda \left[\nabla^2 u \right] \tag{48}$$

where

$$u = \kappa_1 \cdot w - \kappa_2 \cdot \nabla^2 w \tag{49}$$

$$\nabla_i^2 w = \sum_{k=1}^{Neighbors} \left[w_i - w_k \right] \tag{50}$$

where Ω represents the volume of the control volume being considered and $\nabla^2 w$ is the undivided Laplacian of w. The first term of equation (49) constitutes a strong first order dissipation term which is necessary to prevent non-physical oscillations in the vicinity of a shock. To preserve the second order accuracy of the scheme, this term must be turned off in regions of smooth flow. In this research the use of this dissipation term was not necessary for any of the test cases, and the value of k_1=0.0 and k_2=1.0 were used exclusively.

In Equation (48) the overall scaling of the artificial dissipation is accomplished via the factor λ [30], which in the case of scalar dissipation, has been taken as proportional to the maximum eigenvalue, lul+c.

$$\lambda = \kappa \cdot \max(eigenvalue) \tag{51}$$

While for the matrix dissipation model, α is defined as:

$$\lambda = \kappa \cdot T|\Lambda|T^{-1} \tag{52}$$

The T matrices on the right hand side of equation (52) represent the eigenvectors associated with the linearization of the equations of inviscid compressible flow normal to the control volume face, while the $|\Lambda|$ matrix is a diagonal matrix containing the absolute values of the four eigenvalues associated with these equations. Of these four eigenvalues, three are repeated, leaving three distinct eigenvalues which are proportional to: u, u+c, u-c, where u is the velocity normal to the control volume face and c the speed of sound [25].

In both cases, scalar dissipation and matrix dissipation, the artificial dissipation scaling factor is proportional to κ, where κ is a constant of value 20 determined empirically to produce the best results for steady calculations. This value of κ=20 will be referred to as the nominal scaling factor of the artificial dissipation throughout the rest of this investigation.

Since most of this research is concerned with the study of highly turbulent regions associated with massively separated flows, it is critical to give special attention to the correct treatment of the turbulence of the flow. It is crucial to ensure that the artificial dissipation does not damp out oscillations generated by small eddies that must be captured by the solution. As will be shown in Chapter VI, a detailed study was performed, using the case of homogeneous decaying turbulence in a periodic domain, to observe the stability and resolution of the scheme using different levels of artificial dissipation.

Steady Solver

Neglecting unsteady terms, the steady-state form of the conservative equation (19) in Chapter II can be written as:

$$R^*(W) = 0 \tag{53}$$

where W is the solution vector, and R* represents the spatial discretization.

Introducing the pseudo-time variable τ and rewriting equation (53) as

$$\frac{\partial W}{\partial \tau} + R^*(W) = 0 \tag{54}$$

Equation (54) can be advanced in time in two different ways. If the residual, R*(W), is evaluated at the pseudo-time (n), the current iteration being (n+1), an explicit scheme is formulated as:

$$\frac{W^{n+1} - W^n}{\Delta \tau} + R^*(W^n) = 0 \tag{55}$$

Explicit schemes are very easy to implement and parallelize but the pseudo-time step size is limited by the mesh size. Therefore, as the mesh size decreases the allowable time step gets smaller leading to an excessive number of pseudo-time steps to reach the steady state.

If the residual, R*(W), is evaluated at the pseudo-time (n+1), an implicit scheme is obtained:

$$\frac{W^{n+1} - W^n}{\Delta \tau} + R^*(W^{n+1}) = 0 \tag{56}$$

This scheme is unconditionally stable for any pseudo-time step. Linearizing the residual about the pseudo-time step (n), equation (56) can be re-written as:

$$\frac{\Delta W}{\Delta \tau} + \left[R^*(W^n) + \frac{\partial R^*}{\partial W} \cdot \Delta W \right] = 0 \tag{57}$$

which can be re-ordered as:

$$\left[\frac{I}{\Delta\tau}+\frac{\partial R^*}{\partial W}\right]\cdot\Delta W=-R^*(W^n) \tag{58}$$

Therefore,

$$\Delta W=\left[\frac{I}{\Delta\tau}+\frac{\partial R^*}{\partial W}\right]^{-1}\cdot\left\{-R^*(W^n)\right\} \tag{59}$$

and $\left[\frac{\partial R^*}{\partial W}\right]$ is the Jacobian, which represents the change of the residual with respect to changes in the solution values. The Jacobian consists of a large sparse matrix for which the sparsity pattern depends on the stencil of the residual. Each non-zero entry consists of a 5x5 submatrix. It is useful to consider the graph of this sparse matrix as the set of edges joining row and column numbers identifying non-zero block sub-matrices. For a nearest neighbor stencil, the graph of the Jacobian matrix is shown in Figure 2.

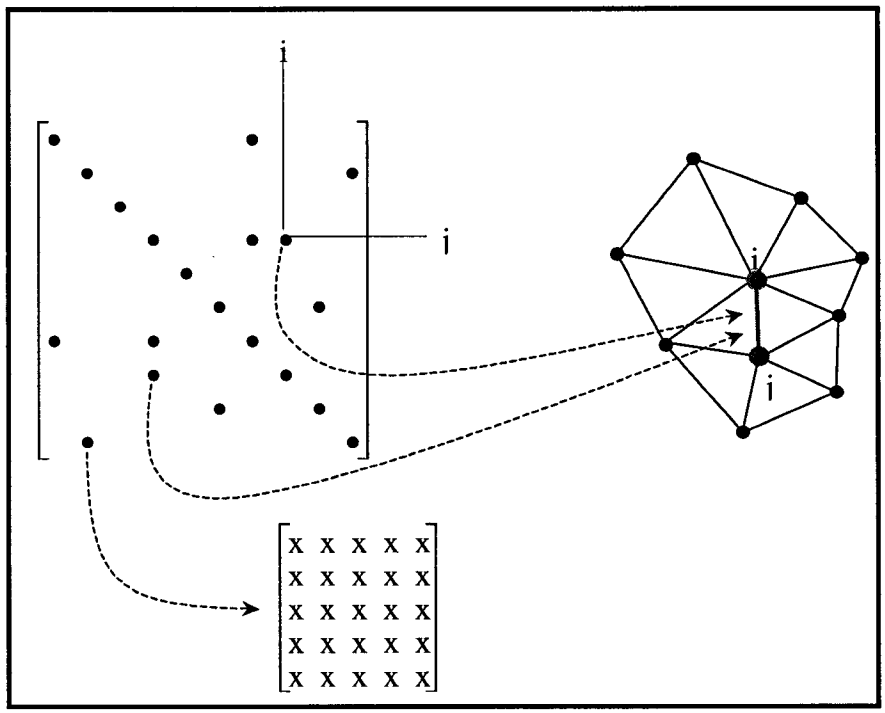

Figure 2. Illustration of graph of block-structured jacobian matrix arising from linearization of unstructured mesh discretization

In our case, the Point-Jacobian approximation will be used for most of the mesh points. For these points, only the non-zero block matrices of the diagonal will be considered for the Jacobian. For points in the viscous region, in which there is a large degree of grid anisotropy, directional smoothing will be achieved by solving implicitly along lines normal to the boundary layer using a tridiagonal solver. For these points, a Line-Jacobian approximation will be used, adding to the Point-Jacobian approximation two block sub-matrices (upper and lower) per edge joining, normal to the boundary layer, two of these points in the viscous region (Figure 2). A graph algorithm is used to identify the points to which the line solver will be applied in a pre-processing phase. Each edge in the mesh is assigned a weight that represents the degree of coupling. Edge weights can be taken, for example, as the inverse of the edge length. For each point, the

ratio of maximum to average weight is an indication of the local anisotropy in the mesh at each point. The points are sorted according to this ratio. The first point in this list is picked as the starting point for a line. The line is built by adding the original point to the neighboring point to which it is most strongly connected based on the edge weights. Each point can only belong to one line and the maximum to minimum edge weight ratio must be greater than a pre-determined value. The line terminates when no additional point can be found. The algorithm results in a set of lines of variable length. In isotropic regions, lines containing just one point are obtained and the point-wise scheme is recovered.

Finally, the corrections will be added to the flow variables using a three-stage implicit multistage scheme with stage coefficients optimized for high frequency damping. The scheme is defined by,

$$
\begin{aligned}
&Q_0 = W^n \\
&Q_k = Q_0 - \alpha_k \cdot \left[\frac{I}{\Delta\tau} + \frac{\partial R^*(Q_{k-1})}{\partial W} \right]^{-1} \cdot R^*(Q_{k-1}) \qquad k = 1,2,3 \qquad (60) \\
&W^{n+1} = Q_m
\end{aligned}
$$

where α_k are the stage coefficients and are defined as α_1=0.5321, α_2=1.3711 and α_3=2.7744.

Multigrid

The basic idea behind multigrid methods is to accelerate the convergence of the solver by computing corrections on a coarser grid than the initial fine grid. The explanation for this approach is based on the frequency distribution of the error of the

numerical solution. Examining the spectral analysis of the residual it is observed that high frequencies are easily damped out with several iterations of an explicit solver. However, low frequency errors are less sensitive to the application of an explicit solver, which is expected, considering the local nature of the information employed by the numerical scheme. Therefore, in a multigrid iteration, the high frequency errors associated with local information are eliminated by the application of the solver on the fine grid, and the low frequency errors associated with more global information are reduced by the application of the solver on a coarser grid, in which the low frequency errors appear as high frequency errors. Typically a multigrid scheme begins by eliminating high frequency errors on the fine mesh. The smoothed solution is then transferred to a coarser mesh and on this coarser mesh corrections are obtained. These corrections will be interpolated back to the fine grid in order to update the solution. This procedure can be applied recursively on a sequence of coarser and coarser grids, where each grid-level is responsible for eliminating a particular frequency bandwidth of errors [26]. Note that multigrid is a convergence acceleration technique which can be applied to any existing discretization. Therefore, the application of multigrid has no effect on the accuracy of the computed solution, but greatly improves the efficiency of the calculation.

For structured mesh cases the construction of coarse mesh levels starting from a fine mesh is quite straightforward since this only requires removing rows and/or columns of grid points from the initial fine mesh. However, for unstructured mesh applications the use of multigrid is not that simple. Coarse levels meshes can no longer be formed as subsets of points of the fine mesh. This has prompted the development of graph-based

methods such as the agglomeration multigrid method for unstructured grids. Multigrid techniques have been successfully extended to unstructured grids using an agglomeration multigrid algorithm.

An agglomeration multigrid technique is used to enhance convergence to the steady state in pseudo-time. The coarse levels are constructed by fusing together neighboring fine grid control volumes to form a smaller number of larger and more complex control volumes in the coarse grid (Figure 3 and Figure 4). A graph algorithm is used to generate the agglomerated levels. This algorithm can be described as follows [28, 29]:

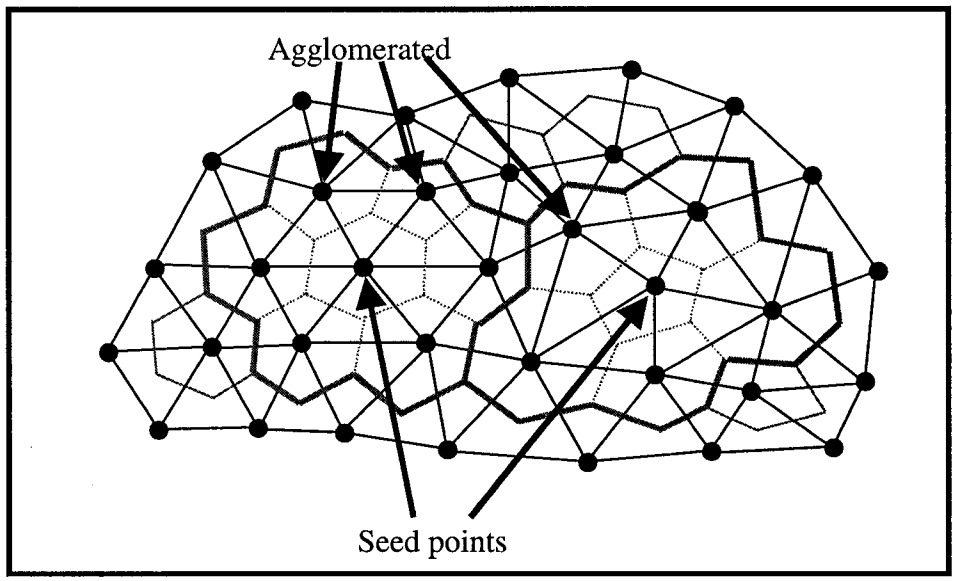

Figure 3. Illustration of agglomeration procedure for coarse level construction in a multigrid agglomeration algorithm.

Step 1. Pick a starting vertex on a surface element. Agglomerate control volumes associated with their neighboring vertices which are not already agglomerated.

Step2. Define a front as comprised of the exterior faces of the agglomerated control volumes. Place the exposed edges in a queue.

Step3. Pick a new starting vertex as the unprocessed vertex incident to a new starting edge which is chosen from the following choices given by order of priority:

a) An edge on the front that is on the solid wall.

b) An edge on the solid wall

c) An edge on the front that is on the far-field boundary.

d) An edge on the far field boundary.

e) The first edge in the queue.

Step 4. Agglomerate all neighboring control volumes of the current point which have not been already agglomerated to another vertex.

Step 5. Update the front and go to step 2 until the control volumes for all vertices have been agglomerated.

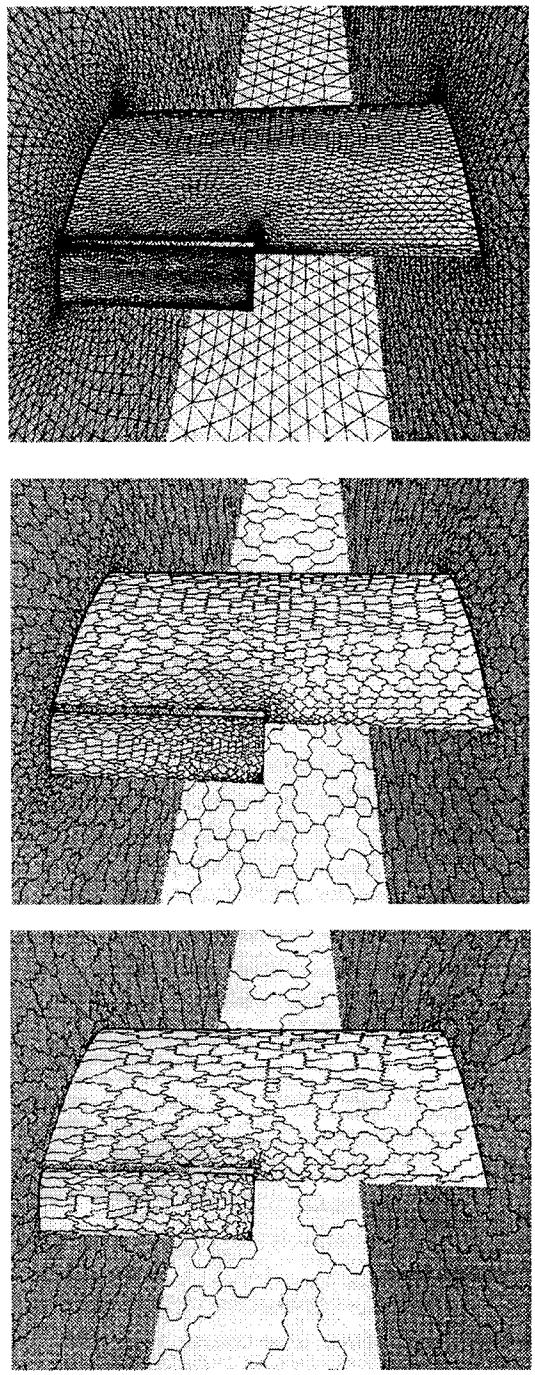

Figure 4. Sample agglomerated multigrid levels used for the computation of the flow over a wing between walls

Fluxes in the coarser levels and Inter-grid operators

Once the different grid levels have been created, the solution of the fine grid must be passed to the coarser levels and the corrections of the coarse levels must be interpolated back to the fine grid. In the case of agglomerated meshes, the construction of the inter-grid operators is particularly simple since all the different grids are fully nested. When going from the fine to the coarse levels, the flow variables as well as the residuals are passed to the coarse levels. For each new cell, the residual is calculated as the sum of the residuals of the agglomerated cells that form the new cell. In the case of the flow variables, the new values are calculated as the volume weighted sum of the flow variables of the agglomerated cells.

On the coarse levels, the agglomerated cells contain segmented edges (Figure 3) which are replaced by straight-line edges in order to simplify the flux integration. The new direction and magnitude associated with the straight-line edge is computed as the vector sum of the normals of the segmented edge, thus ensuring identical flux integration on the new composite edge. Simple injection is employed for the prolongation operator. The correction computed in the coarse level agglomerated cell is applied directly and equally to all fine-level control volumes, which are contained within the coarse level cell.

Parallelization

The solver is parallelized by partitioning the domain using a graph partitioner [27] and communicating between the different partitions, running on different processors, using the Message-Passing Interface (MPI) library. This allows the use of the

solver in distributed memory architectures as well as shared memory machines. The partioning of the computational domain is done on vertices resulting in cut edges that are handled constructing "ghost vertices" as explained in the following section.

Ghost points

At the partition boundaries, edges joining mesh points are cut and each of these points are assigned to different processors. In these cases, the processors hosting these points must communicate in order to compute the fluxes along the cut edges. This is handled in the following manner:

1) The edges cut by the partition boundary are assigned to one processor and a ghost point is constructed in this processor (Figure 5). The ghost point refers to the physical point at the other end of the cut edge, which has been assigned to another processor.

2) The fluxes are computed along edges and accumulated at the vertices (real points and ghost points).

3) The fluxes accumulated at a ghost point must be summed with the flux contributions of the physical point they represent in order to complete the total residual at this point. At this stage, the two processors must communicate to pass the information of the fluxes associated with the ghost point.

4) The updates for all points are calculated by time-stepping the computed residual. Notice that this operation is only applied to physical points and no inter-processor communication is required.

5) Again the processors must communicate to update the values of the flow variables at the ghost point. Then the process can be repeated starting from step 2.

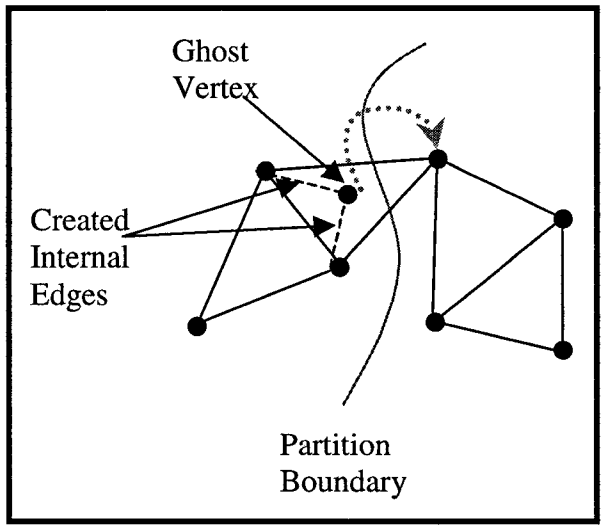

Figure 5. Illustration of creation of ghost points at inter-processor boundaries

Weighted vertices

Since a line-solver is used in some regions of the domain and line-solvers are inherently sequential, any line split between two processors will result in a processor remaining idle while the off-processor portion of the line is computed in another processor. Therefore, the mesh must be partitioned in such a manner that all the points contained within a line remain assigned to the same processor. This is achieved using a weighted-graph-based mesh partitioner called CHACO [27].

The original unweighted mesh is contracted along the implicit lines to produce a weighted graph. Unity weights are assigned to the original vertices and edges. Edges, which are part of an implicit line, are contracted and a single point is formed as the line edges are merged (Figure 6). Merging points produce merging edges and the weights of

the merged points and edges are the sum of the weights of the constituent points and edges. The weighted contracted mesh is then partitioned using CHACO, which generates balanced partitions of weighted points and minimizes the intersection of weighted edges by partition boundaries. Once the partition is completed, the mesh is de-contracted and each partition is assigned to a processor. Since the implicit lines reduce to a point in the contracted mesh, they can never be broken by the partitioning process.

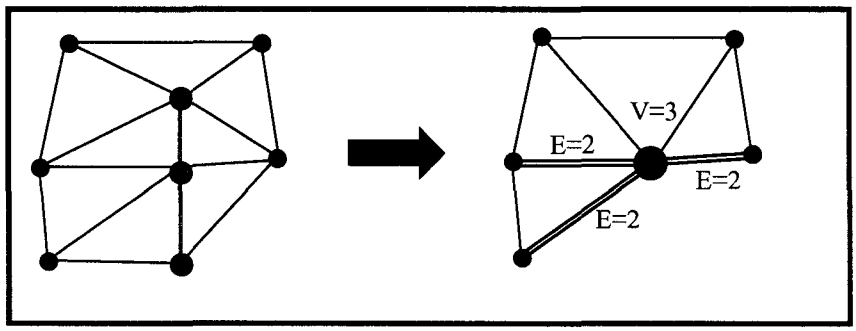

Figure 6. Illustration of edge contraction and creation of weighted graph for mesh partitioning. Contracted line is represented in red.

Partitioning the different grid levels

Since the different grid levels are fully nested, the partition of the fine grid could be used to infer a partition to all the other grid levels. However, this approach, although it minimizes the inter-grid communication, gives little control on the quality of the partition of the coarse levels. Therefore, each level is partitioned independently. This results in unrelated coarse and fine grid partitions. To minimize the inter-grid communication, the coarse level partitions are assigned to the same processor as the fine-grid partition with which they share the maximum number of points.

Unsteady Solver Implementation

Turbulence is an inherently unsteady three-dimensional phenomenon. Therefore, the first necessary step to compute turbulent flows will be to extend the steady solver to an unsteady solver capability. Explicit schemes are well suited for unsteady applications in which the time scale of interest is comparable to the spatial scales. However, explicit time steps may become too restrictive for low frequency cases, or in other words, long time scales. Therefore, it is desirable to develop a fully implicit method in which the time step is only determined by the physics of the flow and not by the cell size. This is done by discretizing the time derivative using a three-point backward difference scheme and solving the non-linear equations at each time step with the steady-state unstructured agglomeration multigrid solution algorithm presented in the previous chapter.

Starting from equation (19), the continuous set of unsteady governing partial differential equations is given by:

$$\frac{\partial U}{\partial t}+\frac{\partial f(U)}{\partial x}+\frac{\partial g(U)}{\partial y}+\frac{\partial h(U)}{\partial z}=0 \tag{61}$$

The spatially discretized equations can be written as:

$$\frac{\partial U}{\partial t}+R(U)=0 \tag{62}$$

where R(u) denotes the discretization of the spatial derivative terms in equations (61). Making a three point backward approximation for the time derivative yields

$$\frac{3}{2\cdot\Delta t}\cdot U^{n+1}-\frac{2}{\Delta t}\cdot U^{n}+\frac{1}{2\cdot\Delta t}\cdot U^{n-1}+R(U^{n+1})=0 \tag{63}$$

Equation (63) is now treated as a steady state equation. A new unsteady residual is defined, which has to be driven to zero (or at least to the truncation error):

$$R^*(W) = 0 \tag{64}$$

where W is the approximation to U^{n+1}. This unsteady residual is defined as:

$$R^*(W) = \frac{3}{2 \cdot \Delta t} \cdot W + R(W) - S(U^n, U^{n-1}) \tag{65}$$

with the source term

$$S(U^n, U^{n-1}) = \frac{2}{\Delta t} \cdot U^n - \frac{1}{2 \cdot \Delta t} \cdot U^{n-1} \tag{66}$$

remains fixed throughout the solution procedure at each time step.

The implicit method presented above will enable larger time steps than an explicit method, but this does not necessarily imply an efficient solution process overall. Unsteady solutions will be tedious to obtain as will be shown in the next chapters in which several unsteady solutions are presented for different flows. Moreover, for DES solutions in which detailed turbulent flow must be captured, the time scale of the unsteady solver will be determined by the smallest eddies that must be resolved. This will result in quite small time scales that will sum up into very long computational times. Nevertheless, implicit schemes are still justified, since the time scales are always larger than the time scales required by an explicit scheme.

As shown in equation (60), the unsteady residual, R*(W), is formed by a source term, $(2/\Delta t)\ U_n - (1/2\ \Delta t)\ U_{n-1}$, that changes at each time iteration and a term, $(3/2\Delta t)W$, that changes for each sub-iteration. This unsteady residual will be solved iteratively using the steady solver presented in Chapter III. The number of iterations required in each time step to converge to a solution (two orders of magnitude reduction of the residual will generally be considered acceptable convergence) is directly proportional to the size of the time step. This implies that for large time steps, a higher number of sub-iterations are required, and for small time steps, a lower number of sub-iterations are necessary to converge to the solution in each time step. The net effect is that the overall computational time is relatively independent of the time step used, but the unsteady accuracy of the solution can be compromised by an inappropriate (too large) time step. In any case, the number of multigrid cycles required to achieve two orders of magnitude reduction of the residual will be problem dependent.

The computational workload required by the unsteady solutions of highly turbulent flows is alleviated by the use of parallel computing. Parallel computing substantially increase the speed of the flow solution. The use of Coral, a PC cluster of 96 processors located in the Institute for Computer Applications in Science and Engineering (ICASE) at NASA Langley Research Center (Hampton) and Helios, an HP10000 of 64 processors at Old Dominion University (Norfolk), was crucial for this research. Most of the cases presented in this research would have been impossible to solve without the use of a parallel code and a multiprocessor machine.

CHAPTER IV

FLOW AROUND A CIRCULAR CYLINDER

Introduction

The flow around a circular cylinder is a well-known case which has been widely studied computationally and experimentally. Although the geometry of the case is simple, the associated flow field is enriched with fundamental fluid mechanics phenomena. At low Reynolds numbers (below 40), the flow around a circular cylinder is characterized by symmetric eddies aft of the cylinder. Around Re = 40 the wake become unstable and the flow is characterized by periodic vortex shedding, referred to as Karman vortex shedding. This vortex shedding has been widely studied and detailed measurements of the Strouhal numbers (dimensionless shedding frequencies) over a wide range of Reynolds numbers have been reported over the years. A compilation of the Strouhal-number vs. Reynolds-number correlation results is shown in Figure 7 [20]. The scatter of the data is attributed to the boundary conditions at the cylinder ends and the 3D effects which appear at Re~190.

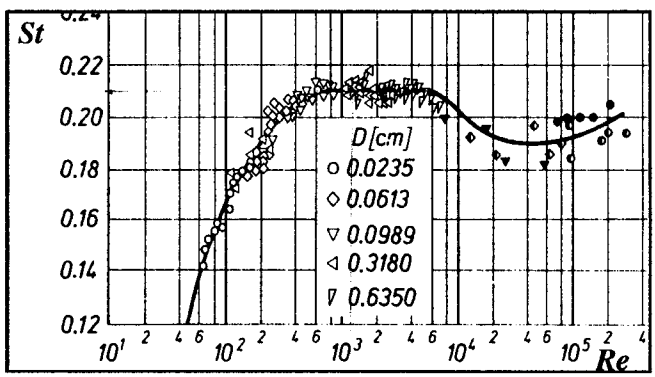

Figure 7. Experimental Strouhal-Reynolds Number correlation for the flow over a circular cylinder. Reproduced from [20].

The unsteadiness of the wake due to vortex shedding has to be captured by the numerical computation through an accurate simulation of the flow-field time behavior. This case is used in this research as the basis for validation of the unsteady RANS solver, and for assessing grid resolution and time step requirements for accurately predicting the vortex shedding frequency observed in the cylinder flow.

Cases with turbulent boundary layer separation have been studied to test the capabilities of techniques such as LES and DES [11]. This research focused on a very detailed description of the wake aft of the cylinder including length of the re-circulation bubble and Reynolds stresses distributions. This is not the main goal of this case in this particular work. As previously stated, the objective in this case is to test the unsteady RANS solver, observing the effect of grid and time-step resolution in the Strouhal number computation.

Computational Domain

This case consists of the flow around a 3D circular cylinder between parallel walls at a Mach number of 0.2 and a Reynolds number of 1,200. The computational domain has an aspect ratio of 1 and a side length of 100 cylinder diameters in the plane normal to the cylinder span. A length of two cylinder diameters is employed in the span direction that extends along the entire crossflow domain. Two different meshes of 252,490 and 631,225 grid points were used with a normal grid spacing for the first point closest to the wall of 5×10^{-5} cylinder diameters. The grid points were distributed in

planes of 25,249 grid points per plane, perpendicular to the span direction and symmetrically distributed along the span. Ten planes were used for the coarse mesh and twenty-five planes were used for the fine mesh. Three different views of the computational domain are shown in Figures 8, 9 and 10.

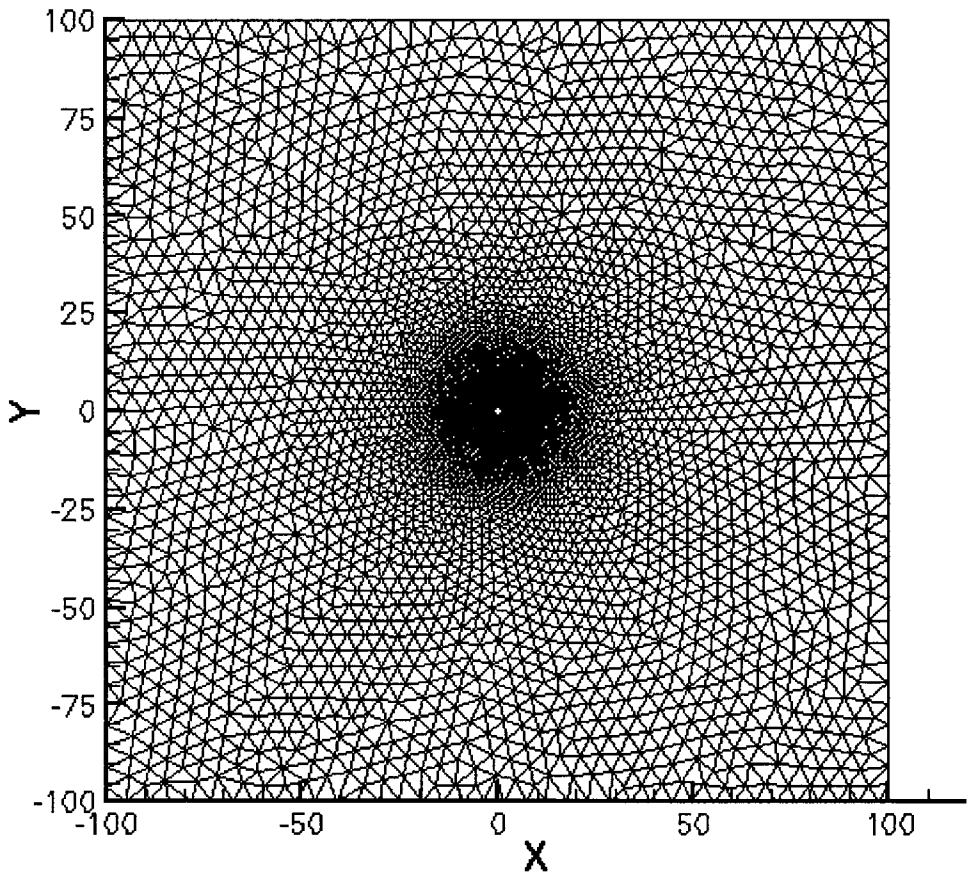

Figure 8. Two-dimensional side view of the computational domain for circular cylinder case. Fine grid of 631,225 points.

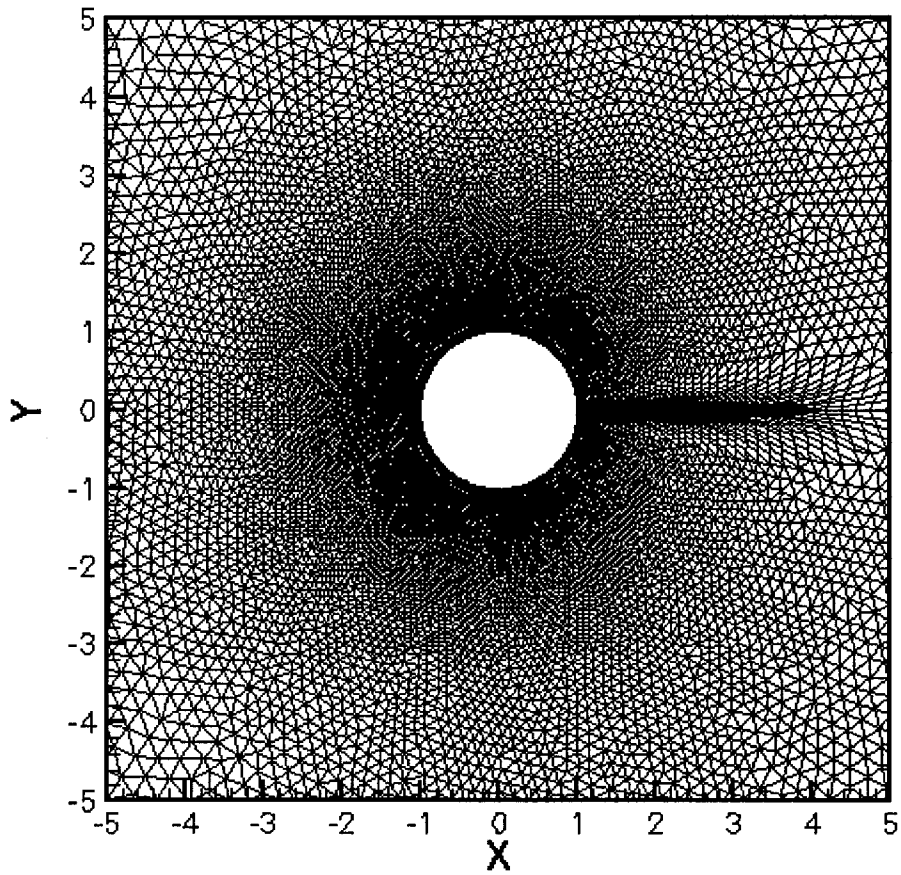

Figure 9. Two-dimensional side view of unstructured grid used for computation of flow over circular cylinder. Number of points=631,225, Wall resolution=$5x10^{-5}$ cylinder diameters.

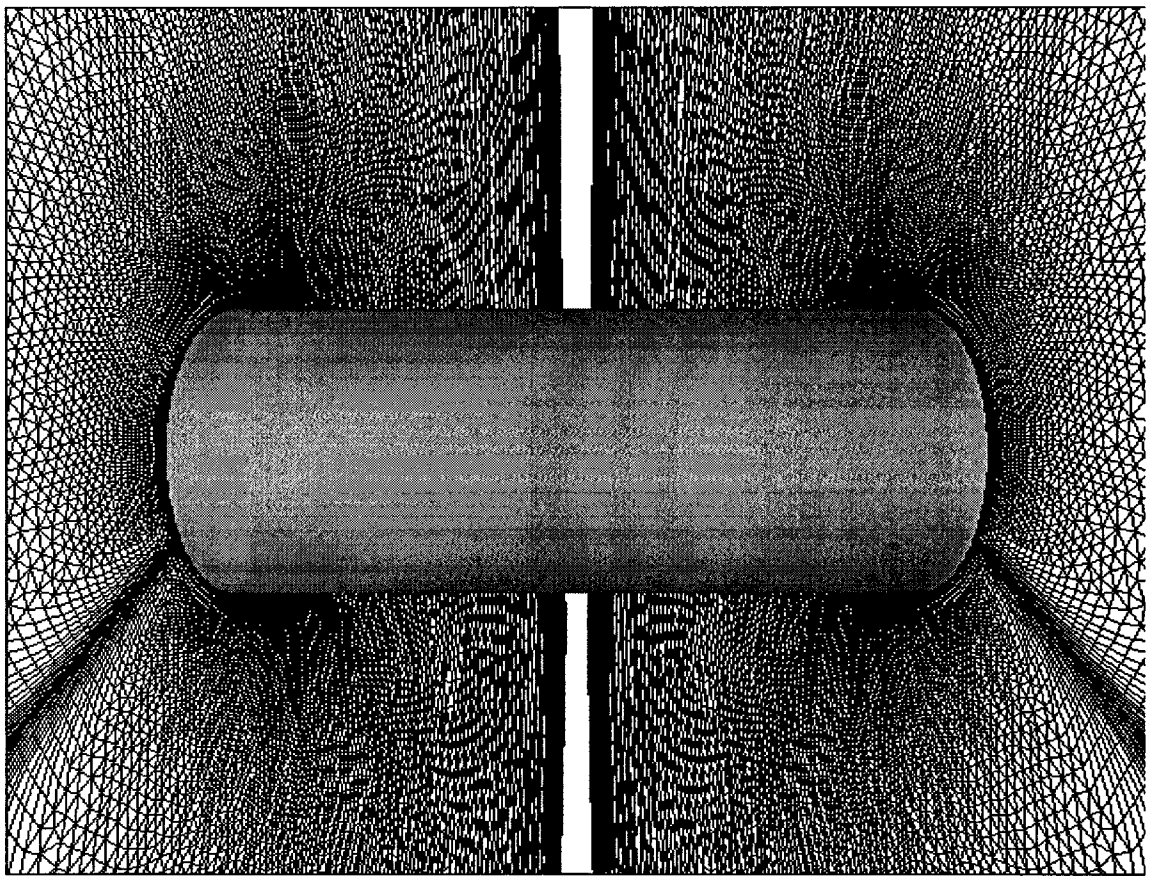

Figure 10. Three-dimensional view of the computational domain for circular cylinder case. Unstructured mesh on side walls of 25,249 points per wall.

Three different time steps of 0.25, 0.125 and 0.05 were used to observe the effect of the time step size on the results. The time is non-dimensionalized as $t = t_o/(d/U_\infty)$ where d is diameter of the cylinder and U_∞ is the freestream velocity. The number of sub-iterations per time step was varied to obtain a residual reduction of two orders of

magnitude per time iteration (Figure 11). The use of an iterative procedure to solve the unsteady residual at each time step requires a number of sub-iterations which grows as the outer time step size is increased.

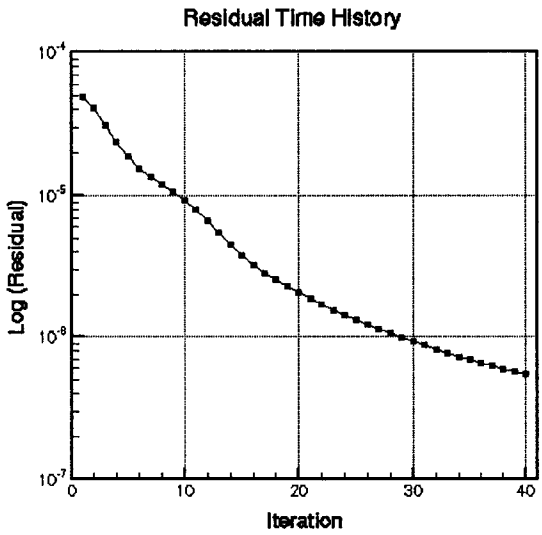

Figure 11. Sample convergence rate of the density residual for one time step and 40 sub-iterations obtained on grid of Figures 8 through 10 using four multigrid levels.

The one-equation Spalart-Allmaras turbulence model was used for all calculations in fully turbulent mode to avoid issues related to transition. This will be shown to affect some of the results. However, the time history of the force coefficients, that represent the main objective of this case, will be unaffected.

In all cases, the agglomeration multigrid strategy was used with four levels (Table 4). All runs were performed in parallel using 16 Pentium II 400 MHz processors in a PC cluster at ICASE.

Multigrid Level	Number of Nodes	Number of Edges	Number of Boundary Nodes
1	631,225	2,267,501	59,560
2	11,318	43,646	906
3	3,677	13,324	913
4	1,508	5,457	247

Table 4. Multigrid Level Description for grid of Figures 8 through 10.

Inviscid (slip velocity) boundary conditions were applied at the end-walls and no-slip boundary conditions were applied on the surface of the cylinder. The three-dimensional simulations reported herein were also compared with two-dimensional simulations of flow around a circle using a validated two-dimensional unstructured solver [21, 22] and found to agree well in terms of force coefficient histories and shedding frequency.

Results

Table 5 shows the Strouhal numbers computed for each mesh and each time step. Convergence is achieved as the time step is reduced and the mesh size increased. A second-order accurate convergence behavior was observed as the time-step was reduced, validating the accuracy of the three-point backwards difference scheme used to discretize the time step. Note how the error is reduced by a factor of 2.99 for the coarse grid and by a factor of 3.42 for the fine grid, as the time step is reduced by a factor of 2 from 0.5 to 0.25, assuming 0.20833 as the grid converged solution. From the smallest time step results, the solution can be seen to be grid converged, at least with respect to the prediction of the vortex shedding frequency. The computed Strouhal number compares very well with the experimental value of St = 0.21 [31]. Figure 12 shows the time history

of the lift coefficient, while the oscillatory pattern corresponding to the vortex shedding is shown in Figure 13.

		Time Step		
		0.5	*0.25*	*0.1*
Grids	*0.252 Million Points*	0.19249	0.20304	0.20833
	0.631 Million Points	0.19379	0.20408	0.20833

Table 5. Predicted Strouhal Number for Various Grid and Time Step Size

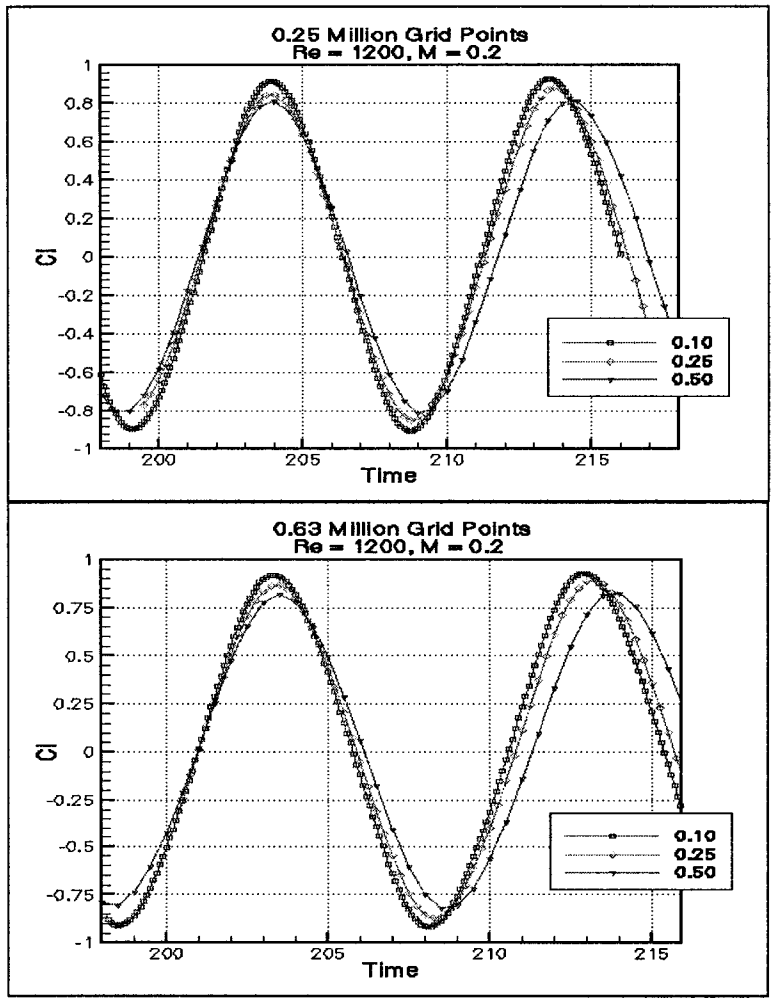

Figure 12. Computed lift coefficient time history for the flow over a circular cylinder using three different time steps. Mach=0.2, Re=1,200

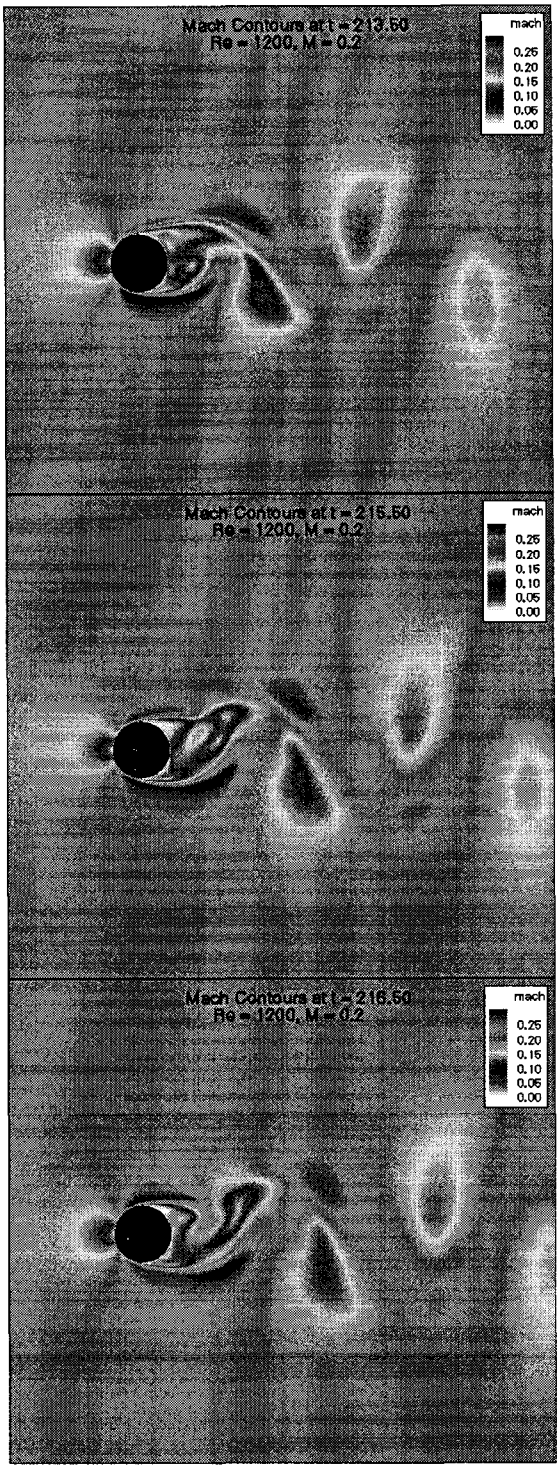

Figure 13. Mach contours at three different time snapshots for flow over a circular cylinder. Mach=0.2, Re=1,200 .

Figure 14 shows the pressure distribution over the surface of the cylinder. The pressure distribution was computed by averaging results at different times distributed along four oscillations. For this calculation, results obtained using the finest mesh and the smallest time step were used. As can be observed from Figure 14, the computed pressure distribution compares more closely with experimental results at a higher Reynolds number than the one used for these computations. This is likely due to the use of the turbulence model in the fully turbulent mode, in order to avoid the issues of transition prediction, which affects the separation point location with the consequent effect on the pressure distribution. Similarly, the backpressure C_{pb} = -1.27 compares closely to the C_{pb} = -1.30 at Re=27,700 measured by Linke [32]. The mean value of the computed drag coefficient is Cd=1.30 compared to Cd=1.20 as measured by Wieselsberger [32, 33] for Re=30,000.

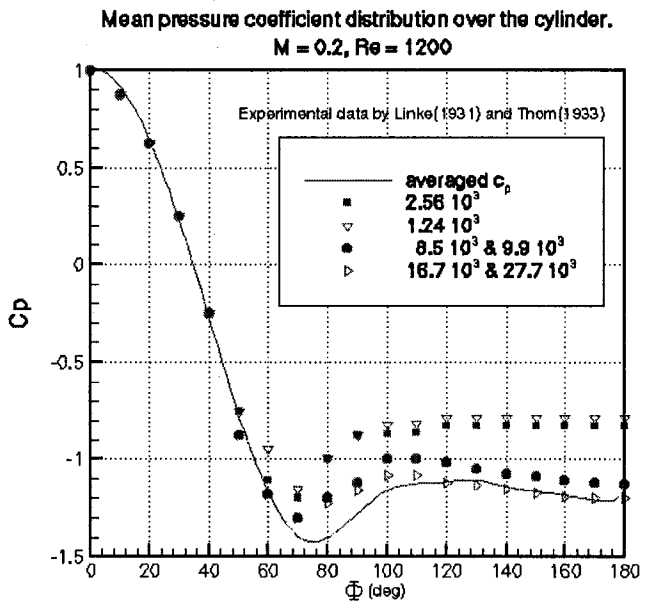

Figure 14. Computed mean pressure distribution over the cylinder surface compared with experimental data. Experimental data extracted from [32].

The mean skin friction over the cylinder surface is shown in Figure 15. Similar to the pressure distribution, the mean skin friction distribution over the cylinder surface was computed by averaging results at different times distributed along four oscillations. For this calculation, results obtained using the finest mesh and the smallest time step were used.

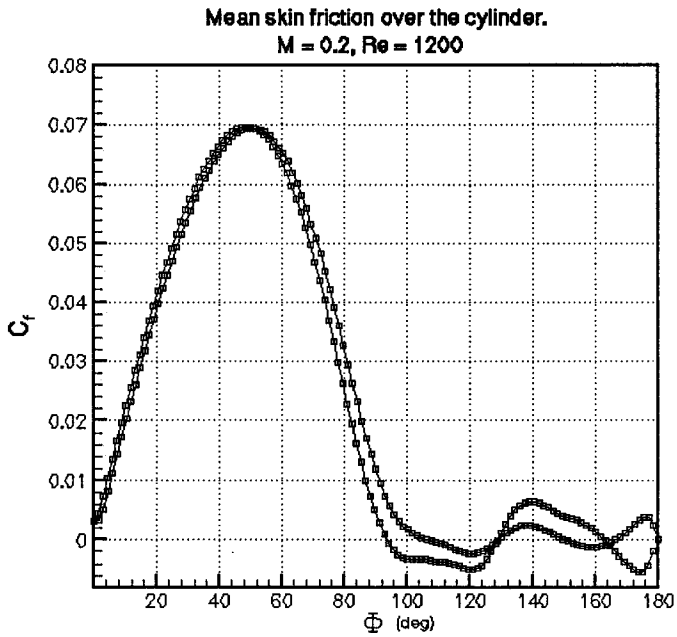

Figure 15. Computed mean skin friction over the cylinder surface. M=0.2, Re=1,200.

Conclusions

The results obtained for the flow around a circular cylinder are very satisfactory and demonstrate the successful implementation of the unsteady terms, making the solver capable of time accurate calculations based on a second order implicit scheme. Since turbulence is inherently unsteady and three dimensional, the solver capability for

unsteady calculations was the first step toward a Detached Eddy Simulation (DES) capability.

CHAPTER V

DETACHED EDDY SIMULATION

Detached Eddy Simulation (DES) was introduced by Spalart in 1999 [8] as a hybrid technique that combines RANS and LES in a non-zonal manner. DES is based on the Spalart-Allmaras one equation RANS model in which the length scale, d, which is traditionally taken as the shortest distance at any given point to the closest wall, is replaced as the minimum between the distance to the wall and a length proportional to the local grid spacing (LGS). The mathematical expression of this is given by

$$d_{DES} = \min (d, C_{DES} \times LGS) \tag{67}$$

where C_{DES} represents a model constant which has been taken as 0.65 in previous work [10, 40]. Traditionally, on structured grids, LGS is taken as the maximum grid spacing over all three directions. In our particular case, the definition of LGS has been modified for unstructured grids by taking it as the maximum edge length connecting a given vertex. In boundary layer regions, LGS far exceeds the distance to the wall, d, and the standard Spalart-Allmaras RANS turbulence model is recovered. However, away from the boundaries the distance to the closest wall exceeds C_{DES} x LGS and the model becomes a simple one-equation sub-grid-scale (SGS) model with the mixing length proportional to the grid spacing. This effect is illustrated by plotting contours of the distance or length scale function for both the RANS and DES models in Figure 16, where it is observed that both models employ the same length scales near the wall, but

use vastly differing length scales in the regions far removed from the wall, where the DES model reverts to an LES mode and a Smagorisnky–like expression for the eddy viscosity is obtained.

Based on the definition of the length scale performed by DES, it is evident that grid isotropy is necessary. This is not an unsolvable problem for anisotropic meshes, but the DES length scale must be redefined to take into account the anisotropy of the mesh in stretched meshes.

Moreover, a good mesh is crucial to DES. Good mesh is defined as mesh that concentrates points in the regions where high levels of vorticity are anticipated and DES is expected to be most important, capturing large concentrations of small eddies. These regions are mainly zones with massively separated flows for which DES was specifically designed. Note that in RANS, it is the mean flow that is being computed, and the role of grid refinement is to minimize mesh influence. Beyond a certain level of grid refinement, the solution accuracy does not improve and becomes limited by turbulence modeling inadequacies. In DES, grid refinement adds physical resolution of the flow by increasing the number of flow features being captured. Nonetheless, another good characteristic of DES is that, in the case of being applied on a mesh which is too coarse to take advantage of all the DES potential, it will behave as a RANS calculation. The solution obtained in this case will not display all the detail expected from a DES calculation but will maintain a "fairly good" averaged value, that may be appropriate for certain engineering tasks. Note that the term "fairly good" is very subjective and requires clarification. In some

flow regimes, characterized by attached flow, these solutions will be within acceptable error intervals, but as the flow starts to separate the solution will deteriorate, and for cases of massive separation, the solution will be quite poor with error percentages that can go up to 50% in C_L, as will be illustrated for a NACA 0012 at a 60 degree angle of attack in Chapter VIII.

Unstructured meshes exhibit flexibility in terms of mesh adaptivity that can be very useful for DES. In a related effort, Spalart [23], in the "Young Person's Guide to Detached Eddy Simulation Grids" (YPG), has described the process of grid design and assessment for DES, defining important regions in the solution and offering guidelines for grid densities within each region. In the YPG, the advantages of unstructured meshes in concentrating points in regions of interest and in coarsening the mesh away from these areas, are pointed out. The YPG also stresses the preference of isotropic cells in DES regions.

DES is based on the Spalart-Allmaras RANS turbulence model and therefore, it maintains some of its characteristics. The SA (Spalart-Allmaras) turbulence model is a useful engineering tool that exhibits its best qualities for attached flows, since it has been calibrated for aerodynamic purposes. As concluded by Wilcox [24], this model presents its worst discrepancies solving jet-like free shear flows (40% discrepancies for spreading rates). On the other hand, far-wake and mixing layer flow results are quite satisfactory (within 14% for spreading rate). In summary, this model appears to be a good starting point for computing massively separated flows especially compared to other turbulence

models. In Table 6 [24], the computed spreading rates for five different flows (far wake, mixing layer, plane jet, round jet and radial jet) using four different turbulence models (Baldwin-Barth, Spalart-Allmaras, k-ω, k-ε) are compared to the measured experimental values. The conventional definition of spreading rate for wakes is the value of the similarity variable, $\eta = y\sqrt{\rho U_\infty^2 / D x}$, where the velocity defect is half of its maximum value, with D the diameter of the circular body generating the wake. Similarly for the plane jet, round jet and radial jet, the spreading rate is the value of y/x where the velocity is half its centerline value. For the mixing layer, the spreading rate is defined as the difference between the values of y/x, where $(U-U_2)^2/(U_1-U_2)^2$ is 9/10 and 1/19. Note also that the SA turbulence model is one of the preferred models in industry because of its simplicity and reasonable accuracy. Industrial turbulence models must try to capture the physics of the flow by introducing the minimum possible complexity.

	One-Equation Models		Two-Equation Models		Experimental
Flow	Baldwin-Barth	Spalart-Allmaras	k-ω	k-ε	Measured
Far Wake	0.315	0.341	0.339	0.256	0.365
Mixing Layer	-	0.108	0.105	0.098	0.115
Plane Jet	-	0.156	0.101	0.108	0.10-0.11
Round Jet	-	0.246	0.088	0.120	0.086-0.096
Radial Jet	-	0.166	0.099	0.094	0.096-0.110

Table 6. Comparison of spreading rates of different free shear flows computed using the Spalart-Allmaras turbulence model and other turbulence models [24].

In contrast, it must also be mentioned that, since DES is based on the SA turbulence model, it retains some of its weaknesses. That is, DES depends on SA to determine transition from laminar to turbulence and more important, DES relies on SA to determine the separation location.

Finally, it must be stated that even though DES is not perfect, and has some inherent problems, it is acceptable in many situations. It is a good approximation for complicated highly turbulent flows, and maintains a good balance between the obtained results and the cost to obtain them, in time and computational resources.

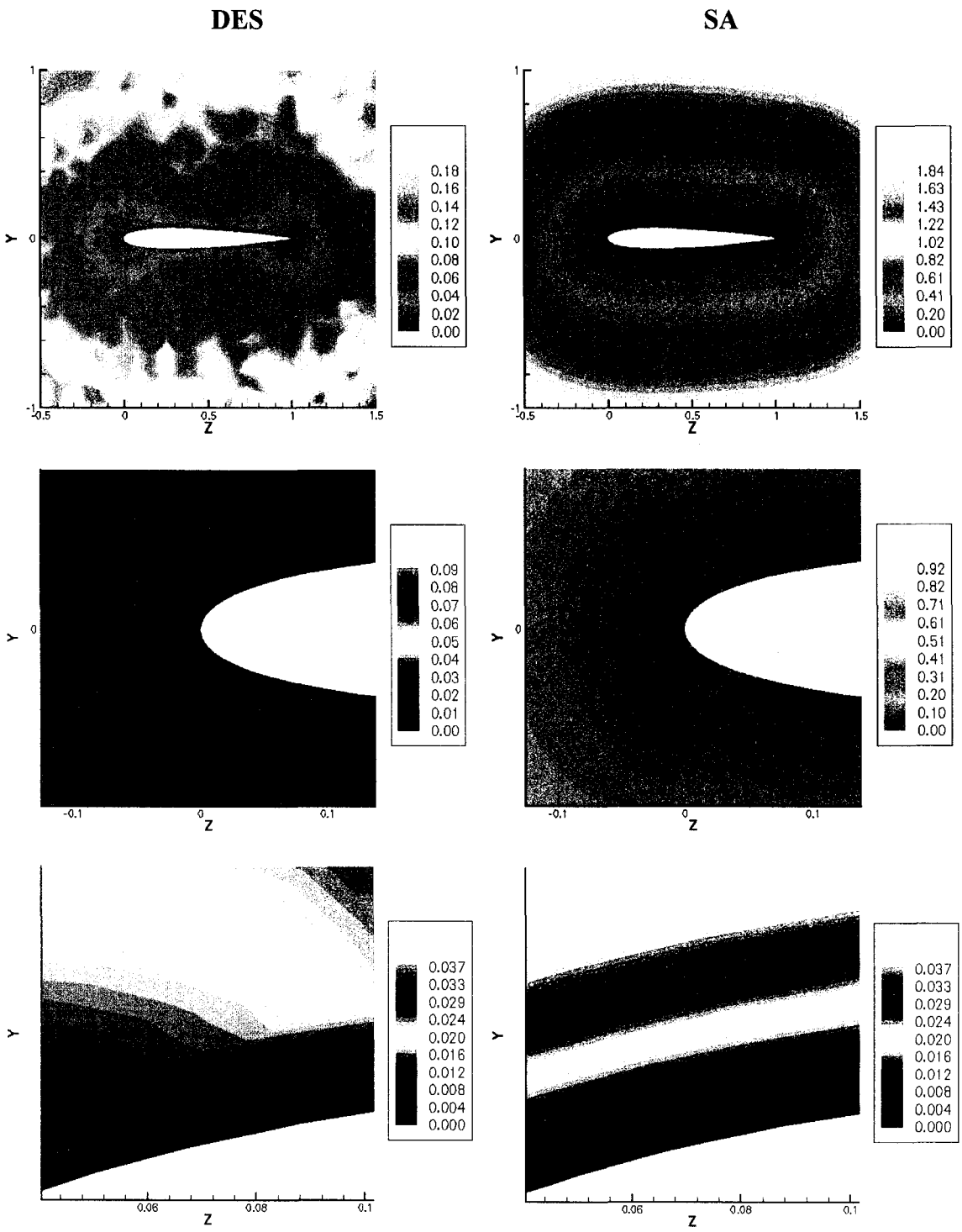

Figure 16. Comparison of the length scale used by the Spalart-Allmaras turbulence model and DES for an unstructured mesh used to compute the flow over a NACA 0012 airfoil shown in Figures 58&59.

CHAPTER VI

DECAYING HOMOGENEOUS TURBULENCE

Description of the case

In this chapter, DES is used in a pure LES mode to evaluate the capability of the modified Spalart-Allmaras single equation turbulence model to act as a fair Sub-Grid-Scale model. Different values of the constant C_{DES} will be tested to validate the value of 0.65 as the optimum as stated by Shur et al. [10]. This case will also be used to extract information about the magnitude and effect of the artificial dissipation of the numerical scheme as compared to the eddy viscosity of the turbulence model.

This test case is based on the experiment performed by Comte-Bellot and Corrsin [34] which consists of studying the correlation coefficient of turbulent velocities behind a regular grid spanning a uniform airstream. This approximates isotropic turbulence since, as stated by Simmons & Salter [35], "the streamwise evolution of the temporally stationary turbulence field set up by a regular grid spanning a steady, uniform duct flow resembles the time evolution of the mathematical ideal of isotropic turbulence". The condition of isotropy is defined by the invariance under coordinate rotation or reflection of the statistically averaged properties of turbulence. Since many of these properties involve two or more spatial locations, isotropy requires homogeneity as well. For simplicity the motion can be restricted to be an incompressible, Newtonian fluid with

zero mean velocity everywhere. This can be visualized as a random motion with zero mean velocity in an infinite domain decaying with time. Obviously, this kind of flow cannot be tested experimentally. However, an approximation can be obtained by using the simplest Eulerian space-time correlation by measuring at two different points behind a grid in a uniform airstream in the streamwise direction, and choosing a time delay for the measurement at the second point that 'cancels' the mean flow displacement.

Computational techniques allow us to perform virtual experiments that would be impossible in reality. It is not necessary to use a space-time correlation to approximate an isotropic decaying turbulence because a decaying random motion in a periodically "infinite" domain can be computationally simulated. Therefore, the computational test case will consist of a square symmetric box with periodic boundary conditions in all directions which is initialized with random values and phases in each node, but with a prescribed three dimensional energy spectrum. The flow inside the computational domain is computed in time and the energy spectrum is observed as it decays, in order to study the decay rate as compared to experimental results and previous computational tests [10, 40, 34]. By "correct" decay it is understood that the numerical scheme will not pollute the energy spectrum and a –5/3 Kolmogorov slope will be recovered in the inertial sub-range. Moreover, the rate of decay should be correct as compared to the experimental results.

Computational domain.

The computational domain consists of a symmetric cube of unit dimension. Four different meshes were used: two coarse grids of 32,768 nodes symmetrically distributed in all directions (32x32x32), in one case formed by prisms and in the other formed by hexahedra, and two fine grids of 262,144 nodes (64x64x64), again one formed by prisms and one formed by hexahedra.

Figure (17) shows a three-dimensional sample view of the computational domain.

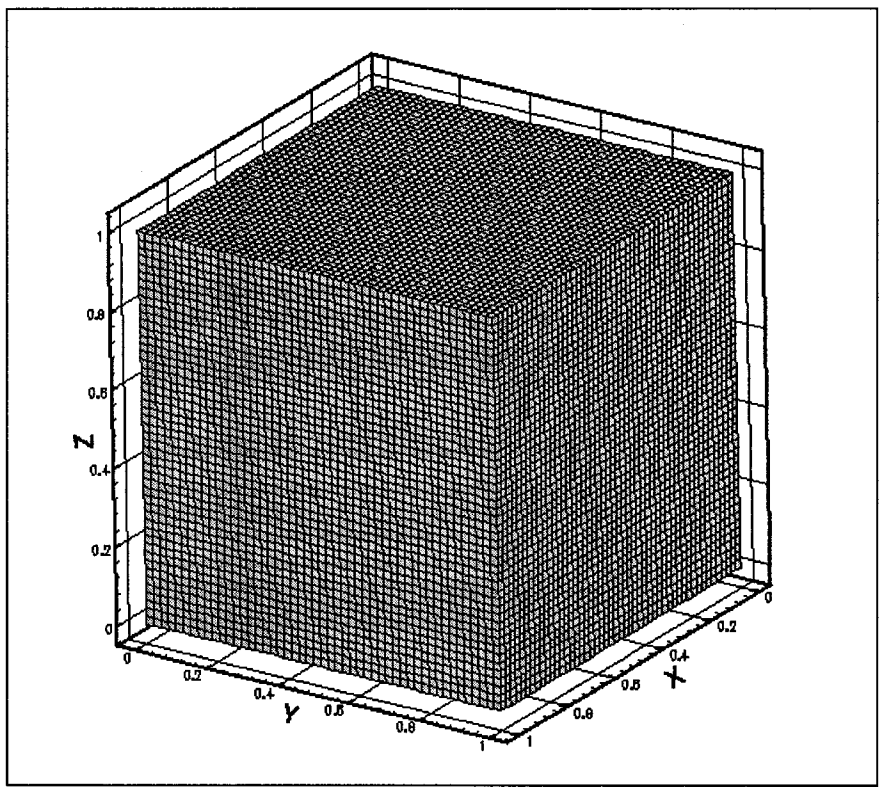

Figure 17. Three-dimensional view of the computational domain for the decaying homogenous turbulence case. 64x64x64 mesh shown.

Initial condition and boundary conditions.

Since the objective is to initialize the computational domain with a velocity field with a pre-defined energy spectrum, the initialization is mainly performed in the Fourier domain. The flow in the computational domain is initialized with an arbitrary periodic velocity field in the Fourier domain and the pre-defined energy spectrum is enforced by multiplying the velocity components by the value of the energy spectrum for the wave number associated with each node. For this velocity field in the Fourier domain to be a realistic velocity field in the physical domain it is necessary to enforce certain conditions. First, symmetry is required with respect to the center of the computational domain to assure real numbers (no imaginary part) in the physical domain when the inverse Fourier transform is performed to go from the Fourier space to the physical space. Second, the velocity field has to comply with the continuity condition. A detailed description of the process follows to initialize the variable values at each node:

1. Assign to each node a wave number:

$$\begin{aligned} k_1(i_x,i_y,i_z) &= i_x \quad , \quad i_x = 1,\ldots,N \\ k_2(i_x,i_y,i_z) &= i_y \quad , \quad i_y = 1,\ldots,N \\ k_3(i_x,i_y,i_z) &= i_z \quad , \quad i_z = 1,\ldots,N \end{aligned} \tag{68}$$

N being the number of nodes in each direction (32 for the coarse grid or 64 for the fine grid).

2. Assign to each node a velocity vector in Fourier space with its components formed by random Gaussian amplitudes of zero average:

$$\begin{aligned}\hat{u}_1(i_x,i_y,i_z) &= (-\ln U_1)^{1/2}\cdot[\cos(2\cdot\pi\cdot U_2)+i\cdot\sin(2\cdot\pi\cdot U_2)]\\ \hat{u}_2(i_x,i_y,i_z) &= (-\ln U_1)^{1/2}\cdot[\cos(2\cdot\pi\cdot U_2)+i\cdot\sin(2\cdot\pi\cdot U_2)]\\ \hat{u}_3(i_x,i_y,i_z) &= (-\ln U_1)^{1/2}\cdot[\cos(2\cdot\pi\cdot U_2)+i\cdot\sin(2\cdot\pi\cdot U_2)]\end{aligned} \tag{69}$$

where

$$\begin{aligned}U_1 &= \textit{uniform random number between } [0,1]\\ U_2 &= \textit{uniform random number between } [0,1]\end{aligned}$$

3. Considering that the shape of the energy spectrum is known, it is projected onto the velocity field computed in step 2 by multiplying the velocity components by the value of the energy spectrum for the wave number associated with each node.

$$\begin{aligned}\hat{u}_1(i_x,i_y,i_z) &\rightarrow \hat{u}_1(i_x,i_y,i_z)\cdot f(k)\\ \hat{u}_2(i_x,i_y,i_z) &\rightarrow \hat{u}_2(i_x,i_y,i_z)\cdot f(k)\\ \hat{u}_3(i_x,i_y,i_z) &\rightarrow \hat{u}_3(i_x,i_y,i_z)\cdot f(k)\end{aligned} \tag{70}$$

where

$$k(i_x,i_y,i_z) = \sqrt{k_1^{\,2}(i_x,i_y,i_z)+k_2^{\,2}(i_x,i_y,i_z)+k_3^{\,2}(i_x,i_y,i_z)} \tag{71}$$

4. The velocity components are forced to be symmetric with respect to the center of the cube. In this way, real values for the physical velocity field are obtained when the three dimensional inverse Fourier transform is computed .

$$\hat{u}^*_i(i_x,i_y,i_z) \Leftrightarrow \hat{u}_i(-i_x,-i_y,-i_z)$$
$$\hat{u}^*_i(i_x,i_y,i_z) \Leftrightarrow \hat{u}_i(N-i_x,N-i_y,N-i_z) \tag{72}$$

$$\hat{u}(0,0,0) = \frac{1}{2}\cdot\left[\hat{u}(0,0,0) + \hat{u}^*(0,0,0)\right] \tag{73}$$

5. The incompressibility condition ($\nabla \cdot u = 0$) is enforced:

$$\hat{u}_j(i_x,i_j,i_z) = (\delta_{pj} - \frac{k_p(i_x,i_y,i_z)\cdot k_j(i_x,i_y,i_z)}{k^2(i_x,i_y,i_z)})\cdot \hat{u}_p(i_x,i_j,i_z) \tag{74}$$
$$j = 1,2,3 \qquad p = 1,2,3$$

Being the incompressibility condition in Fourier space:

$$k_i \cdot \hat{u}_i = 0 \tag{75}$$

since:

$$\hat{u}_i = u\cdot e^{-ikx_i}$$
$$\frac{\partial u_i}{\partial x_i} = -ik\cdot u\cdot e^{-ikx_i} = -ik\cdot\hat{u}_i \tag{76}$$

6. Finally, the three-dimensional inverse Fourier transform is computed and the physical velocity field (Figure 18, 19, 20) with a characteristic energy spectrum is obtained.

The boundary conditions are periodic in all directions to emulate an "infinite" computational domain. The variables are initialized with equal value for opposite nodes

in opposite boundary faces and the residuals are forced to be the average of the residual at these nodes at each iteration. As a result, the updated variable values at the boundary faces are the same and the domain behaves as an infinite domain. The periodic boundary conditions were tested by initializing the velocity field with an average freestream velocity in the x-direction and adding a periodic disturbance of zero average to the velocity value. The result was periodic disturbances moving with average velocity in the x-direction, such that the disturbance would disappear through one face of the domain and re-appear through the opposite face due to the periodic boundary condition.

A time step of 0.01 was used for the runs, where time was non-dimensionalized as

$$t = t_o \cdot L \cdot \left(1.5 \cdot u'^2\right)^{1/2} \tag{77}$$

where L is the computational domain side length and u' represents the initial root-mean-square (rms) of the average velocity fluctuation. The resulting flowfields at t=0.87 and t=2.0 are post-processed to obtain the energy spectra, which are then compared to the corresponding experimental data.

In all cases, it is necessary to obtain the initial eddy viscosity field by pre-converging the turbulence model running with the flow-field held frozen. Once the initialization is completed, the solution is advanced in time using the implicit time-step procedure described in Chapter III.

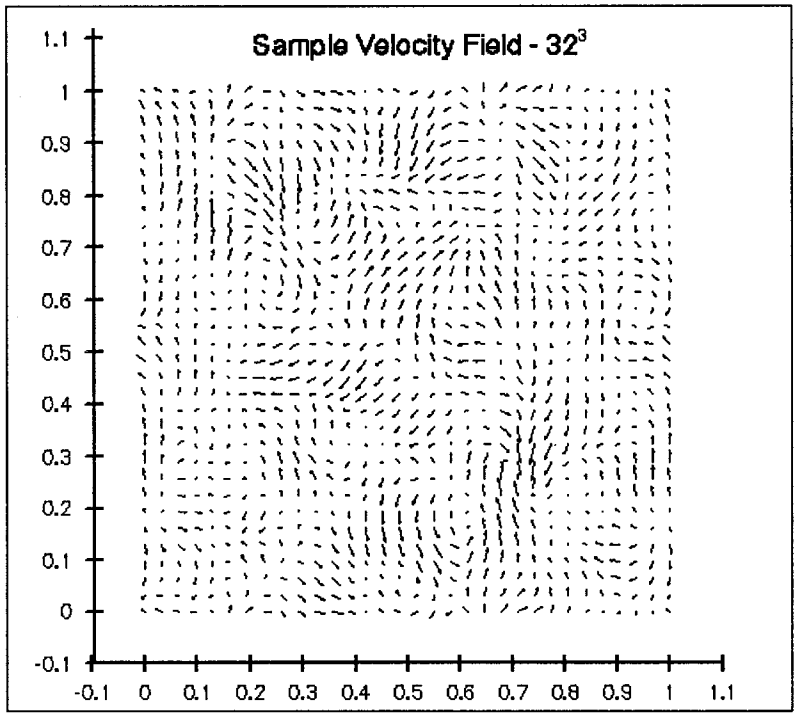

Figure 18. Sample two-dimensional cut of the initial velocity field for the 32x32x32 node mesh.

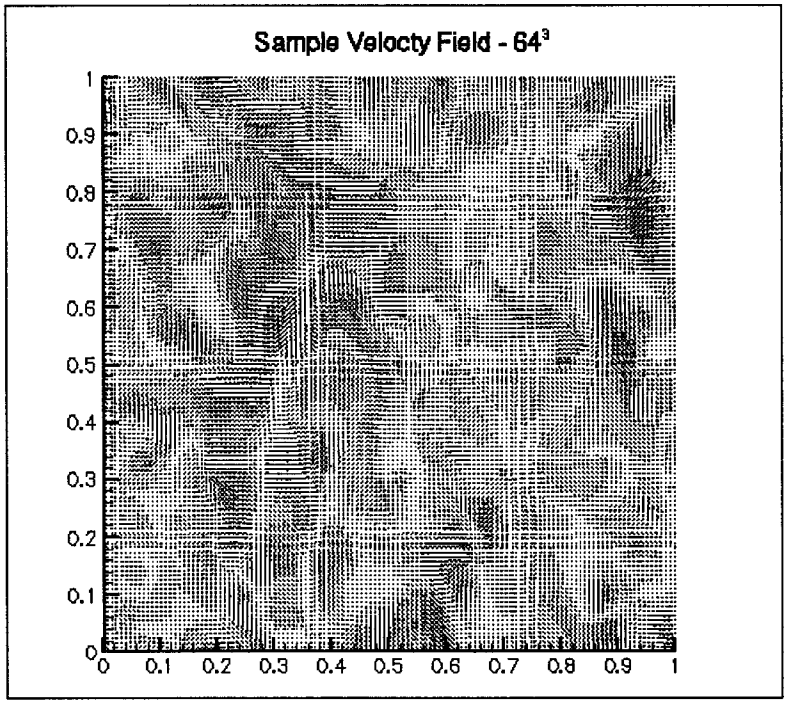

Figure 19. Sample two-dimensional cut of the initial velocity field for the 64x64x64 node mesh.

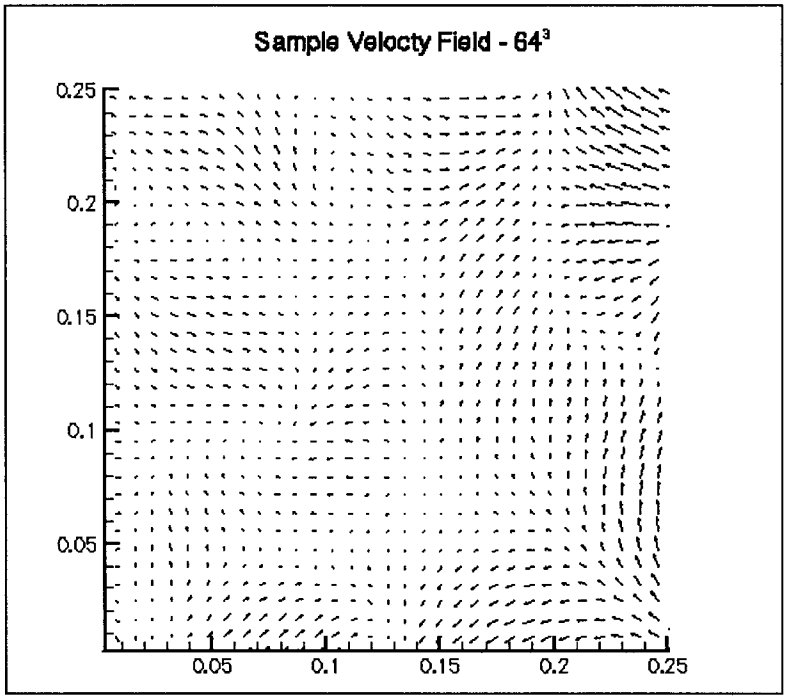

Figure 20. Close up view of the sample two-dimensional cut of the initial velocity field for the 64x64x64 node mesh

Cases Tested and Results

Three different kinds of tests were performed.

1. First, the decaying homogeneous turbulence was run for four different levels of artificial viscosity (vis2 = 3, 4, 5, 20; 20, being the value generally employed for steady calculations in RANS mode). All cases were run with and without the turbulence Sub-Grid-Scale model to afford an evaluation of the effect of the eddy viscosity on the overall solution. Two different hexahedral meshes of 32,768 and 262,144 mesh points were employed for these runs. The objective of these runs was to assess the effect of the artificial viscosity as compared to the eddy viscosity computed by the DES Sub-Grid-Scale model, and therefore, to be able to determine appropriate levels of artificial viscosity for accuracy and stability.

2. Second, different values of C_{DES} (0.25, 0.55, 0.65, 0.75) were tested for two hexahedral meshes of 32,768 and 262,144. For all these tests a fourth of the nominal value of the artificial viscosity scaling factor was used. In this case the objective was to conclude if the C_{DES} value of 0.65 is the optimum as stated by Shur et al. [10].

3. Finally, four different meshes were compared: two meshes of 32,768 nodes, one formed by hexahedral and one by prismatic cells, and another two meshes of 262,144, one formed by hexahedral cells and the other by prismatic cells.

In these runs the C_{DES} value used was 0.65 and the artificial viscosity level was set to a fourth of its nominal value. In this case, the two meshes were compared to observe the difference produced by different cell shapes.

Test 1

The results obtained for the first set of cases are shown in Table 7 and from Figure 21 to Figure 26. These figures illustrate the computed energy spectra in both grids (32,768 and 262,144 nodes) at two time levels for different values of the artificial dissipation scaling factor starting with the nominal value, i.e. the value generally employed for steady calculations in the RANS mode. As can be observed in Figure 21 and 22, the finer scales decay more rapidly than do the experimental values. When the same simulation is performed with the eddy viscosity turned off, little difference in the energy spectra is observed, suggesting that the eddy viscosity values are overwhelmed by the levels of artificial dissipation. Repeating the same computation for lower scaling factors of the artificial dissipation terms (0.25, 0.20, 0.15 of the nominal value) substantially better agreement is observed at all scales, as can be observed in Figures 23, 24, 25 and 26. However, for some of these cases, stability problems arise when the artificial viscosity is reduced bellow a certain level and the eddy viscosity is not high enough to maintain the stability of the numerical scheme. These cases are marked in Table 7 as "not converged" which indicates that at some point, the numerical scheme became unstable (not enough dissipation) and could not converge. As can be observed in Figures 23 and 24, the results obtained for a scaling factor of the artificial dissipation of 0.25 of the nominal value, produced a good agreement up to k=10 for both grids. The

agreement for lower wave numbers is reasonably good for both grids. The value of a fourth of the scaling factor nominal value for the artificial dissipation was used in consecutive tests since it showed reasonably good results for both grids and did not show any stability problems.

Fraction of nominal artificial dissipation scaling factor	Artificial dissipation scaling value	Sub-Grid-Scale model	Mesh-32^3	Mesh-64^3
1	20	Disabled	Converged	Converged
1	20	Activated	Converged	Converged
¼	5	Disabled	Converged	Not converged
¼	5	Activated	Converged	Converged
1/5	4	Activated	Converged	Not converged
1/6.66	3	Disabled	Not converged	Not converged
1/6.66	3	Activated	Not converged	Not converged

Table 7. Summary of converged/not converged runs of the tests performed with different levels of artificial dissipation and with Sub-Grid-Scale model activated or disabled (Test 1).

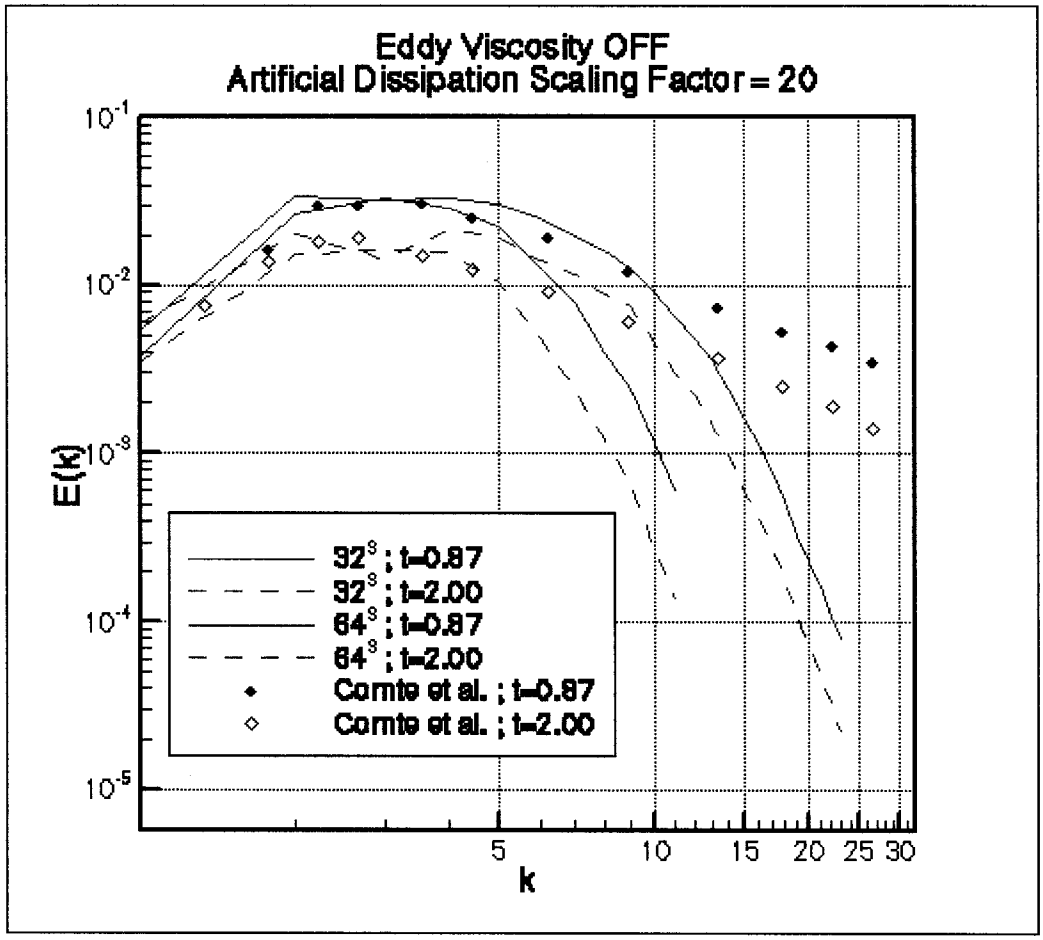

Figure 21. Comparison of the computed and measured energy spectra decay. Sub-Grid-Scale model disabled. Nominal value of the artificial dissipation scaling factor. Computation performed in fine (64^3) and coarse (32^3) meshes.

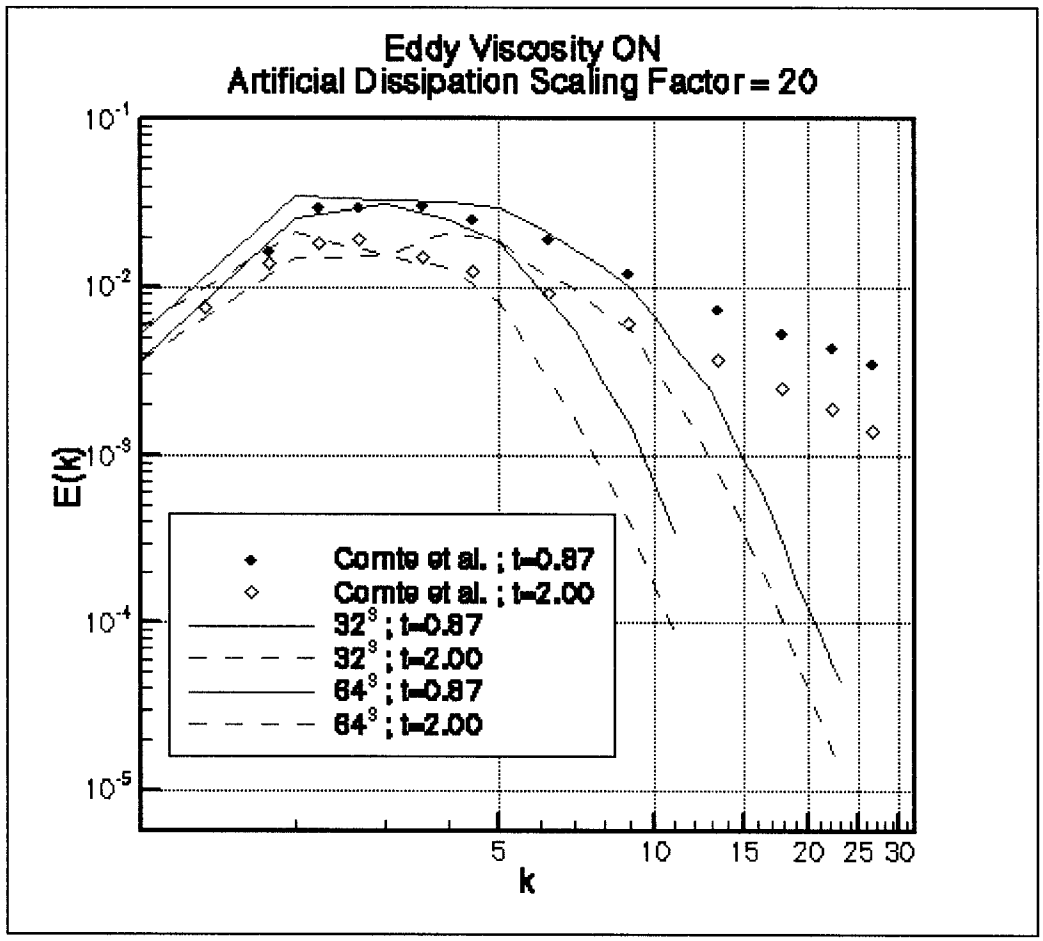

Figure 22. Comparison of the computed and measured energy spectra decay. Sub-Grid-Scale model activated. Nominal value of the artificial dissipation scaling factor. Computation performed in fine (64^3) and coarse (32^3) meshes.

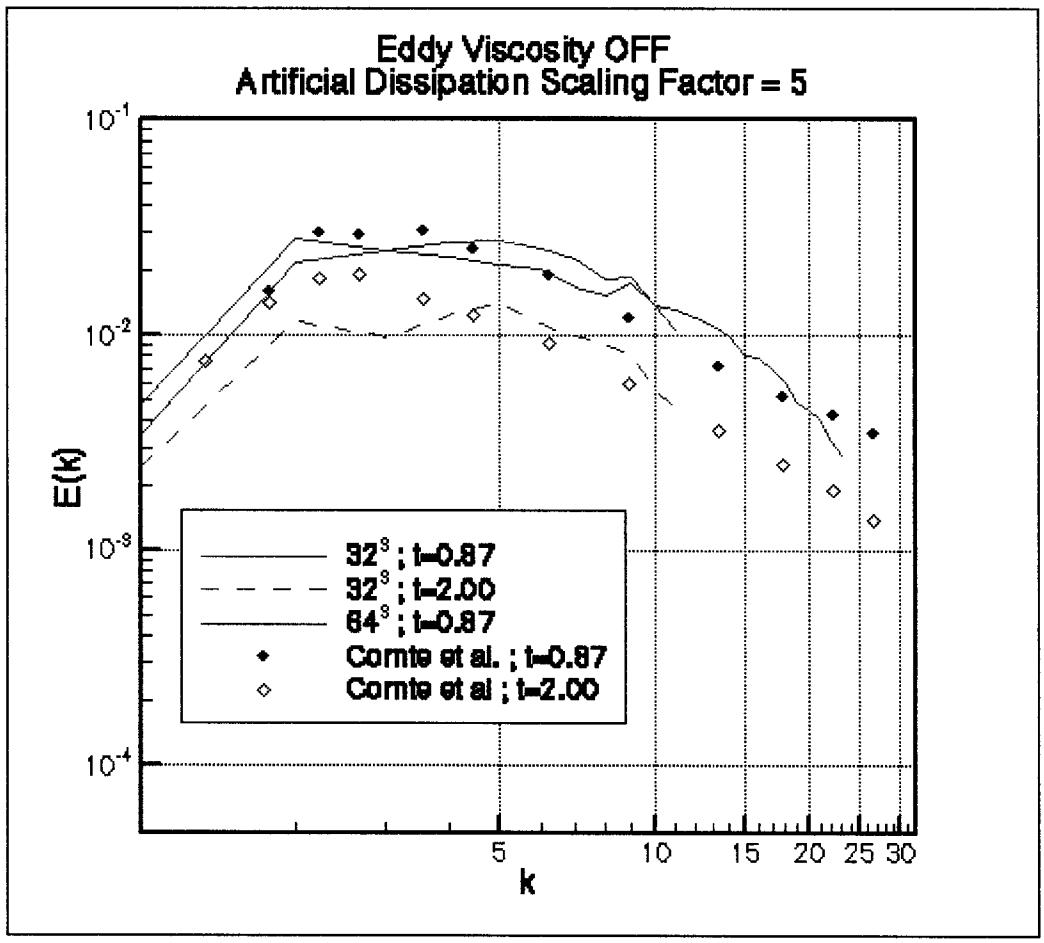

Figure 23. Comparison of the computed and measured energy spectra. Sub-Grid-Scale model disabled. 1/4 of the nominal value of the artificial dissipation scaling factor. Computation performed in fine (64^3) and coarse (32^3) mesh.

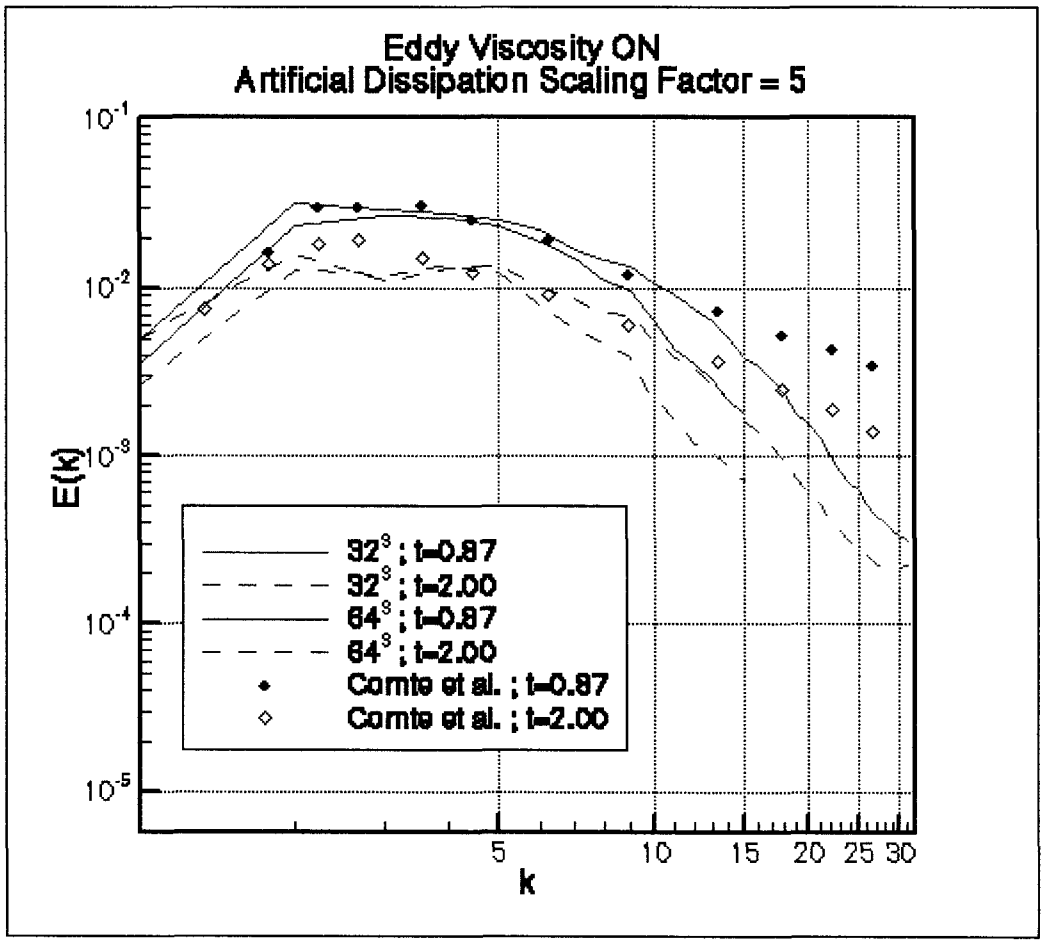

Figure 24. Comparison of the computed and measured energy spectra decay. Sub-Grid-Scale model activated. 1/4 of the nominal value of the artificial dissipation scaling factor. Computation performed in fine (64^3) and coarse (32^3) mesh.

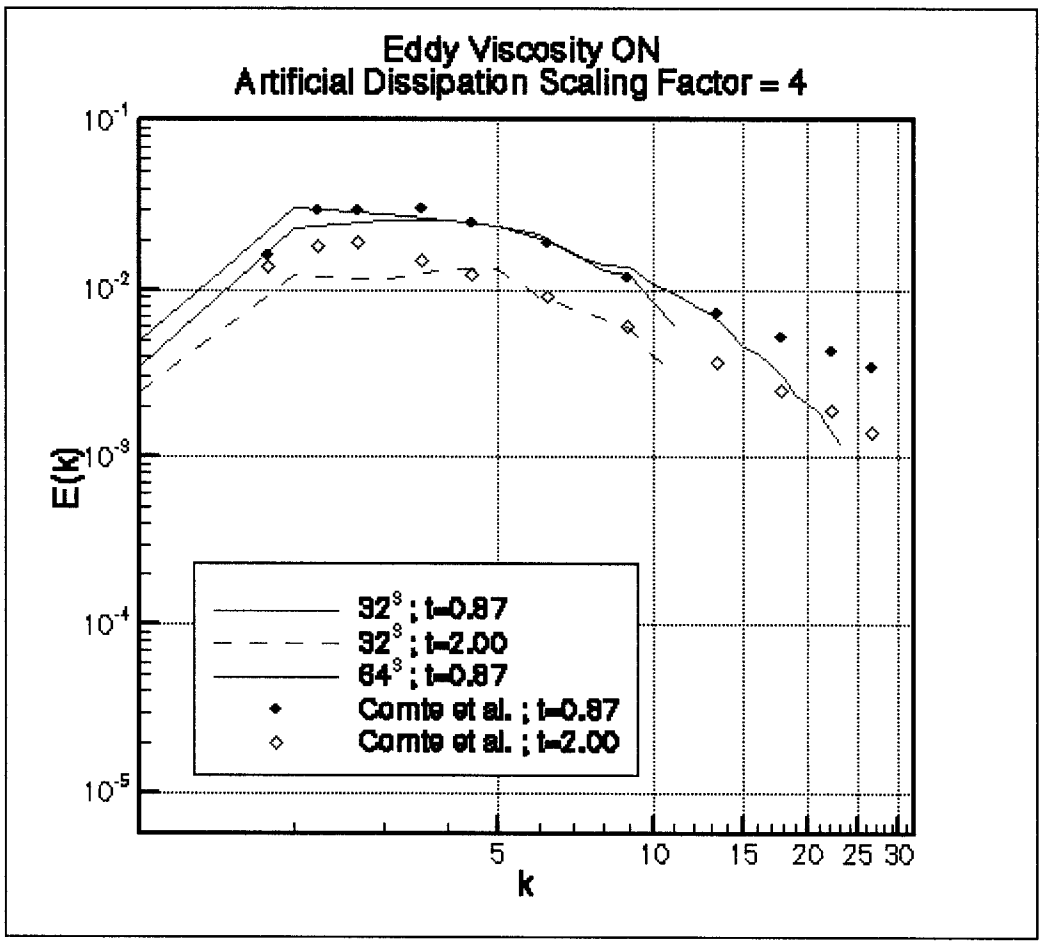

Figure 25. Comparison of the computed and measured energy spectra decay. Sub-Grid-Scale model activated. 1/5 of the nominal value of the artificial dissipation scaling factor. Computation performed in fine (64^3) and coarse (32^3) mesh.

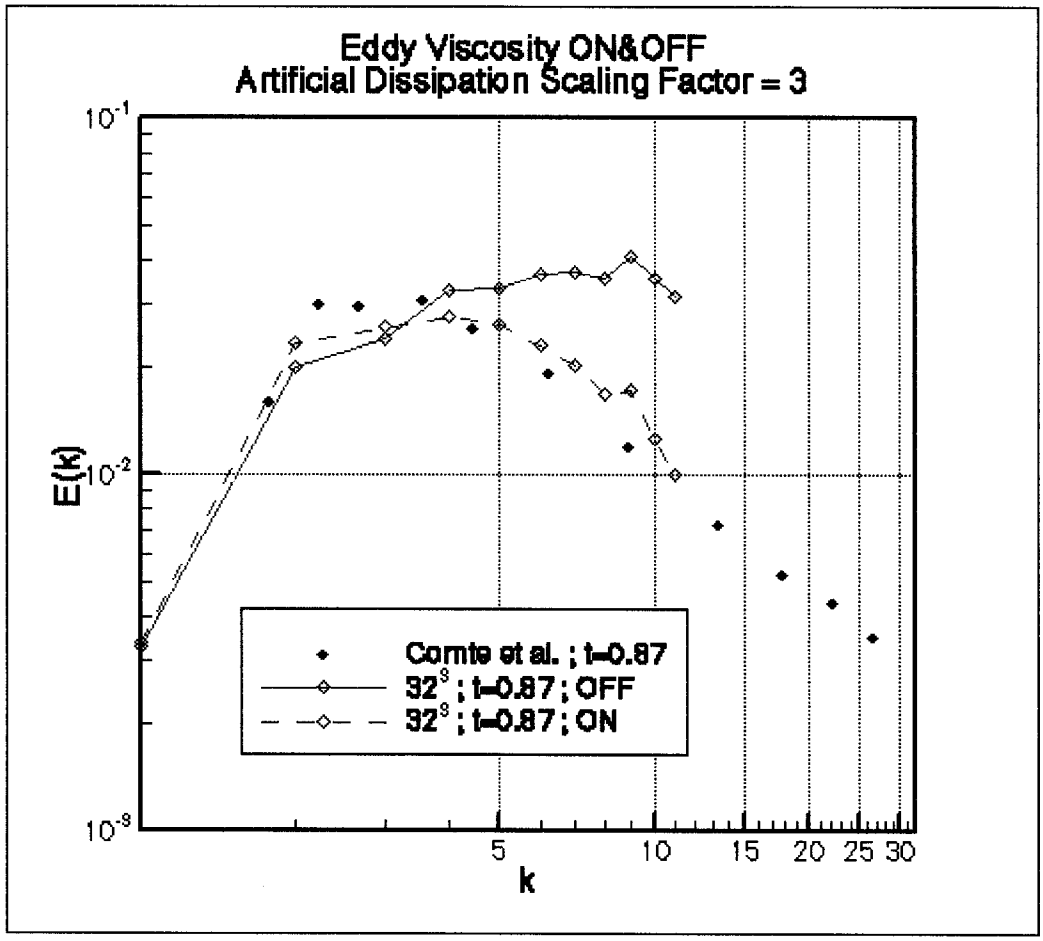

Figure 26. Comparison of the computed and measured energy spectra. (1/6.66) of the nominal value of the artificial dissipation scaling factor. Computation performed in coarse (32^3) mesh.

Test 2

The results of the second test are shown in Figures 27 and 28. In this case, different values of the constant C_{DES} were tested for two hexahedral meshes of 32,768 and 262,144 nodes. This constant acts as a proportionality constant for the eddy viscosity computed by the Sub-Grid-Scale model, which will affect the velocity decay rate of the finer scales. Shur et al. [10] also computed this case and concluded that C_{DES}=0.65 was the optimum value. All these cases were computed using a fourth of the nominal value of the artificial dissipation scaling factor. As can be observed in Figures 27 and 28, C_{DES}=0.65 and C_{DES}=0.5 are the two values which give the best results. While C_{DES}=0.25 did not converge for some of the cases and C_{DES}=0.75 is too dissipative.

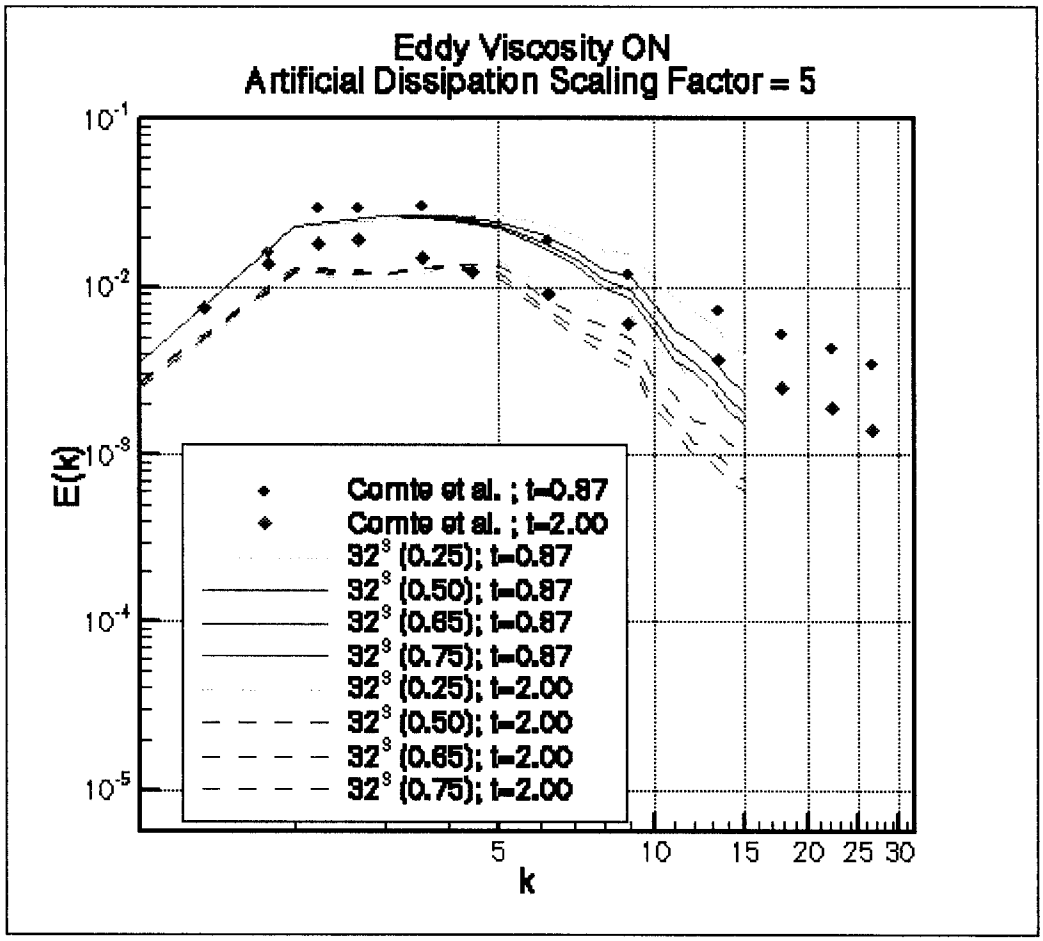

Figure 27. Comparison of the computed and measured energy spectra decay for different values of the C_{DES} constant. Computation performed in coarse (32^3) mesh with (1/4) of the nominal artificial dissipation scaling factor. Sub-Grid-Scale model activated.

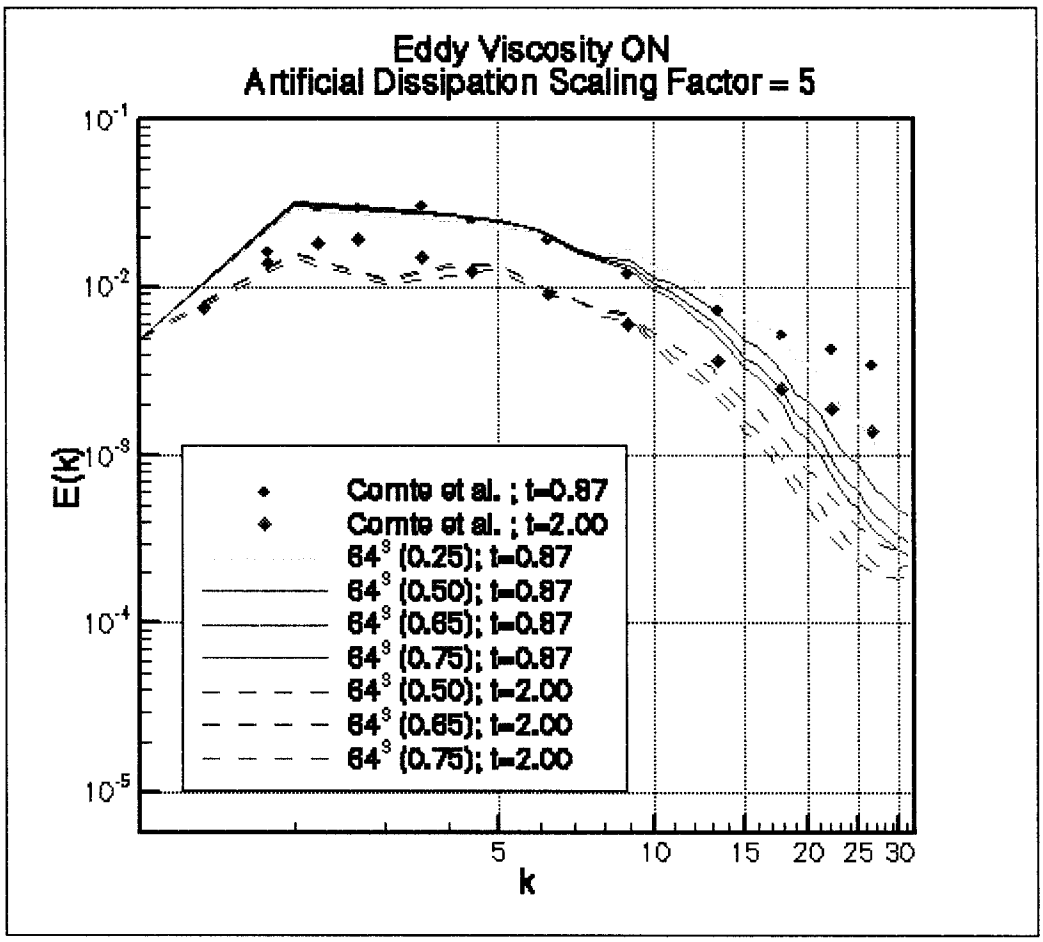

Figure 28. Comparison of the computed and measured energy spectra decay for different values of the C_{DES} constant. Computation performed in fine (64^3) mesh with (1/4) of the nominal artificial dissipation scaling factor. Sub-Grid-Scale model activated.

Test 3

Figures 29, 30, 31 and 32 show the results obtained for test 3 in which the difference between hexahedral meshes and prismatic meshes was examined for meshes of 32,768 and 262,144 nodes. All these cases were computed using C_{DES}=0.65 and a fourth of the nominal value of the artificial dissipation scaling factor. The results obtained show that prismatic element meshes produce higher levels of dissipation than hexahedral element meshes. This is a reasonable result since the length scale used by the Sub-Grid-Scale model was defined as the maximum edge length incident on each node. Considering that the prismatic cells were constructed by dividing the hexahedral cells in two prisms using a diagonal plane, new longer edges defined by the diagonal planes appear in the prismatic mesh. This definition of the length scale is taken from the original DES definition by Shur et al [10] and produces the wrong effect of making a prismatic cell, theoretically finer than a hexahedral mesh and more capable of capturing small eddies, more dissipative since the length scale used by the turbulence model is approximately 1.5 times higher, stimulating the eddy viscosity generated by the Sub-Grid-Scale model. A better definition of the length scale is necessary to avoid effects such as this, especially in meshes in which DES will be applied to regions containing different element shapes. In our test cases this issue will not be decisive since the DES regions will be formed exclusively by tetrahedral elements.

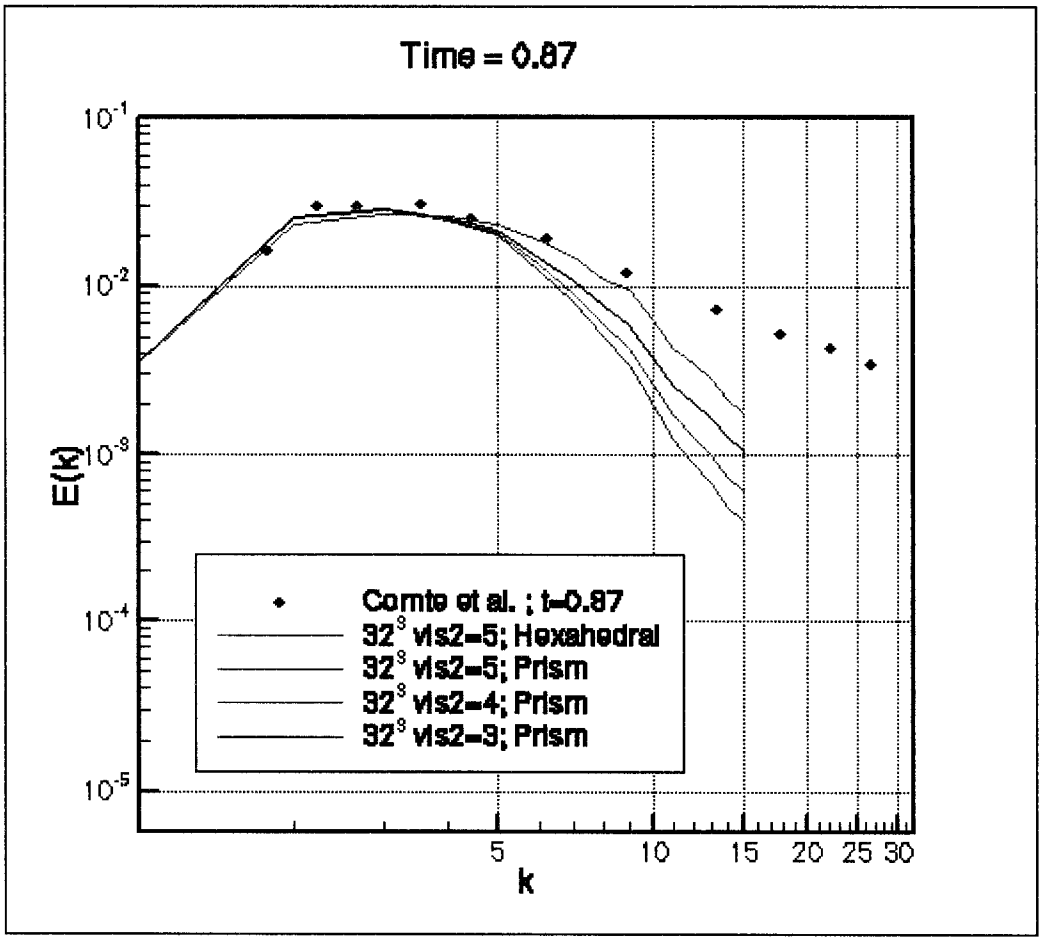

Figure 29. Comparison of the computed and measured energy spectra decay to t=0.87 for different mesh types (Hexahedral & Prism). Computation performed in coarse (32^3) mesh with an artificial dissipation scaling factor of 5. Sub-Grid-Scale model activated.

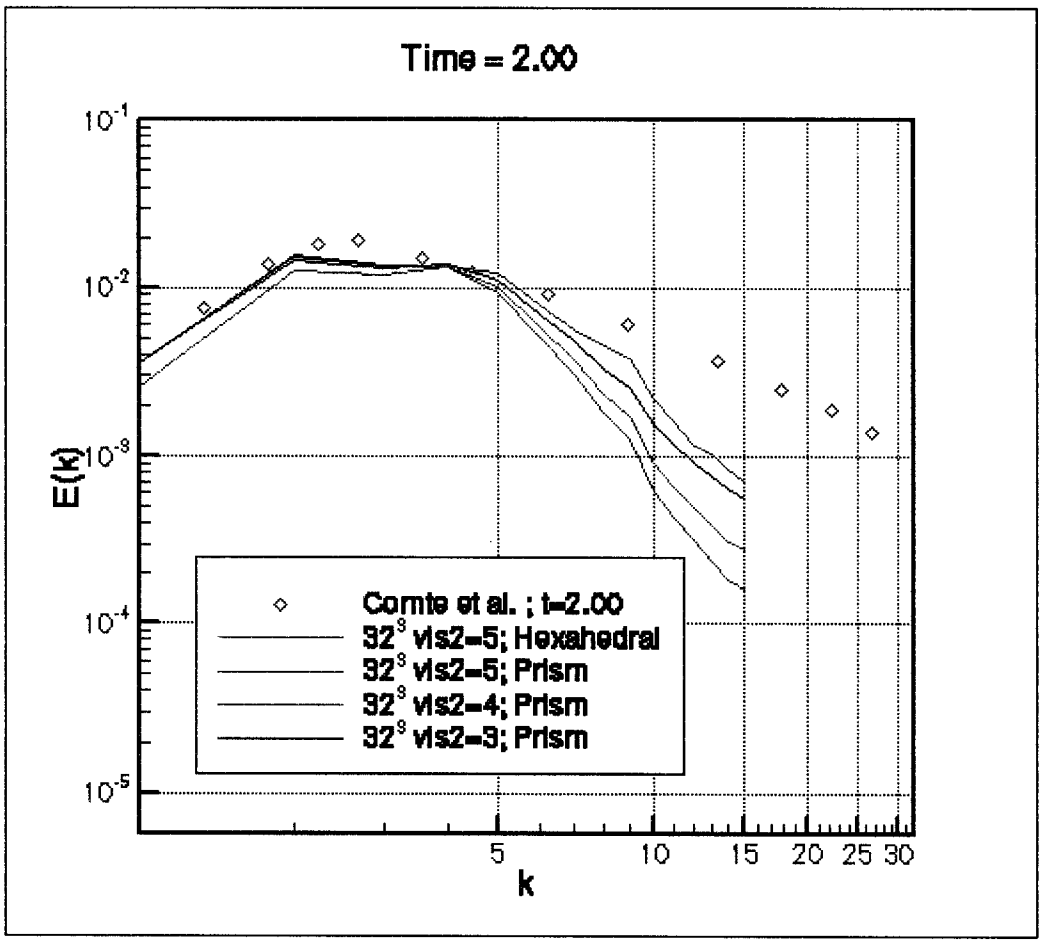

Figure 30. Comparison of the computed and measured energy spectra decay to t=2.00 for different mesh types (Hexahedral & Prism). Computation performed in coarse (32^3) mesh with an artificial dissipation scaling factor of 5. Sub-Grid-Scale model activated.

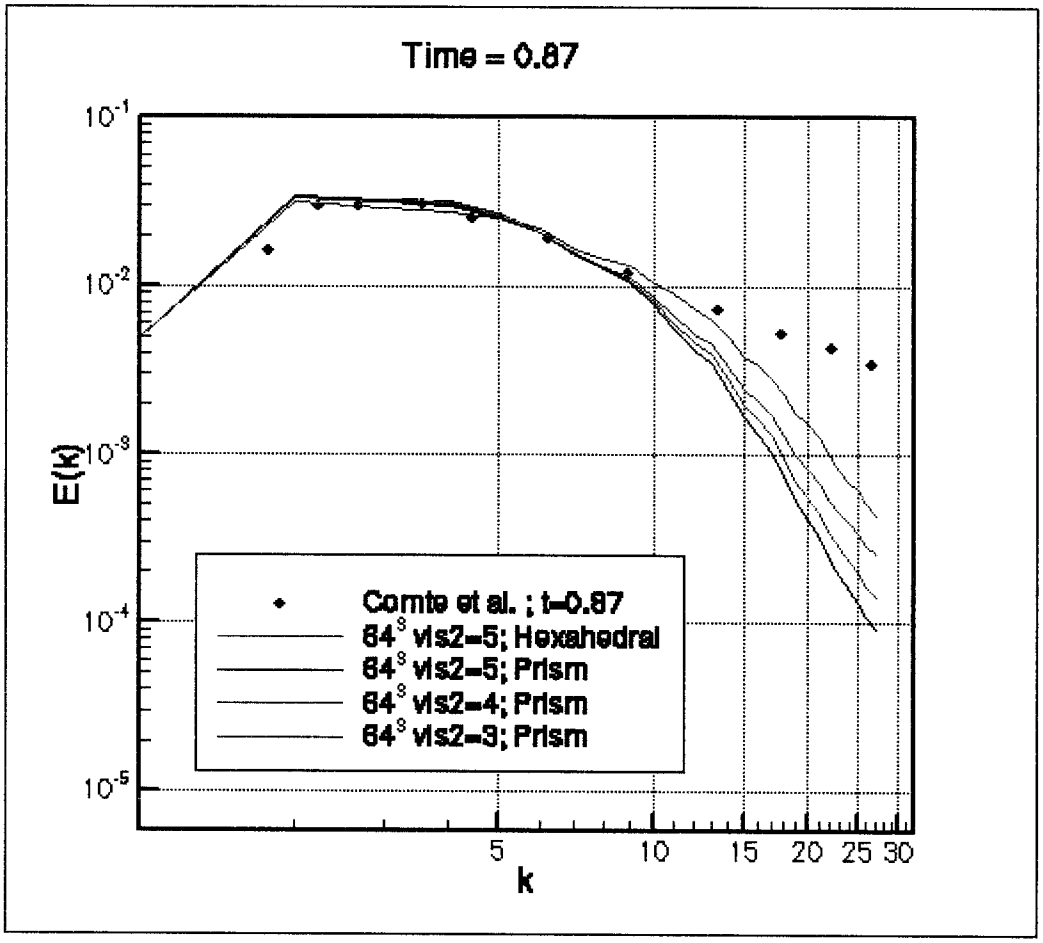

Figure 31. Comparison of the computed and measured energy spectra decay to t=0.87 for different mesh types (Hexahedral & Prism). Computation performed in coarse (64^3) mesh with an artificial dissipation scaling factor of 5. Sub-Grid-Scale model activated.

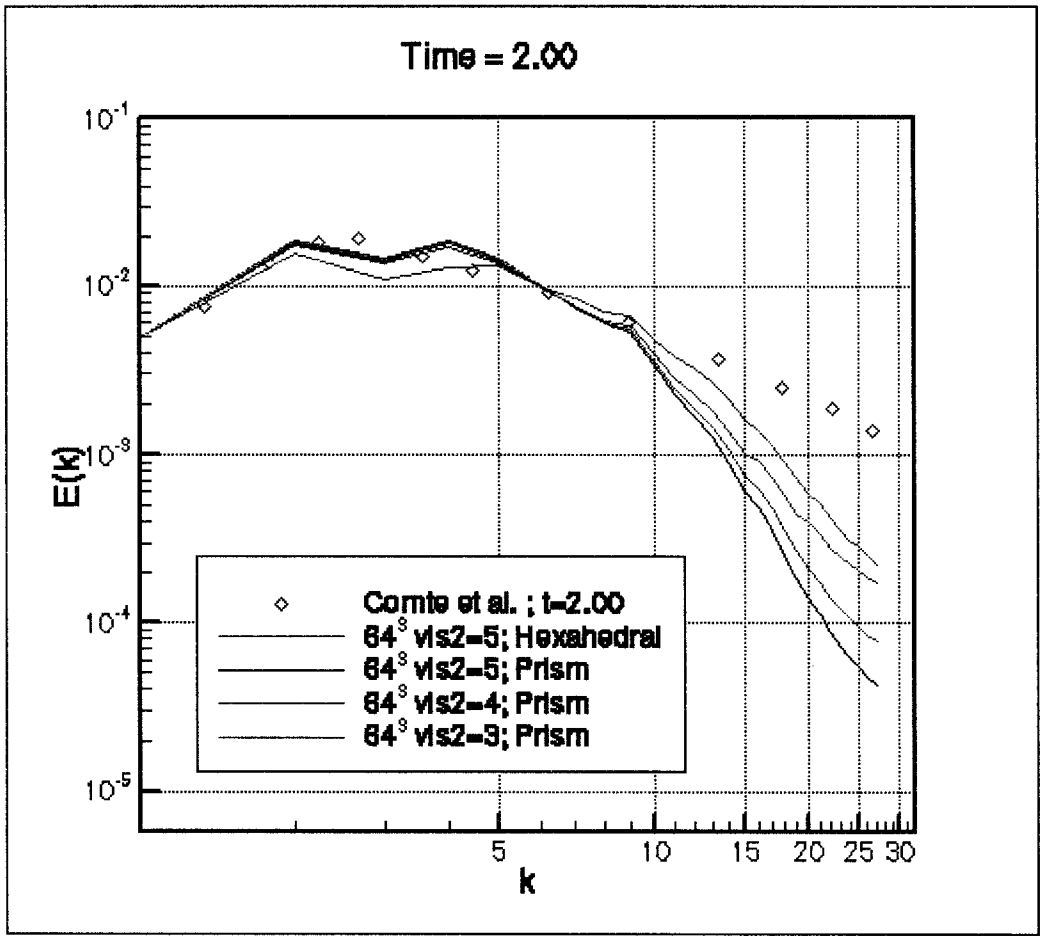

Figure 32. Comparison of the computed and measured energy spectra decay to t=2.00 for different mesh types (Hexahedral & Prism). Computation performed in coarse (64^3) mesh with an artificial dissipation scaling factor of 5. Sub-Grid-Scale model activated.

Conclusions

The test case of decaying homogeneous turbulence in a periodic domain was used to investigate DES in a pure LES mode and to assess the effects of artificial dissipation, C_{DES} values, and mesh type in its performance. The results showed that a fourth of the nominal value of the artificial dissipation scaling factor yields adequate solutions without compromising the stability of the numerical scheme. This means that the solution closely reproduced the decaying energy spectrum up to a wave number of 10 in the fine mesh without polluting the solution with undesired numerical effects. The lowest value to which the numerical dissipation can be reduced before the numerical scheme becomes unstable will be test dependant. Nevertheless, at this point, a fourth of the nominal value seems quite reasonable assuming a good quality mesh. The conclusions by Shur et al. [10] were corroborated and a C_{DES} value of 0.65 was taken to be appropriate for maintaining a good equilibrium between excessive dissipation and numerical stability.

The different mesh types (hexahedral and prismatic cells) revealed potential problems when using meshes of mixed elements in DES regions. The definition of the length scale of the Sub-Grid-Scale model as the maximum edge length incident on each node can produce the wrong effect in cases in which smaller cell volumes can have longer edges, as is the case for prismatic shapes. In such cases, higher levels of eddy viscosity will be computed by the Sub-Grid-Scale model, and at the same time smaller eddies will be captured by smaller cell shapes. This will produce excess dissipation in meshes with smaller cell shapes which were expected to produce better results.

CHAPTER VII

FLOW AROUND A SPHERE

Introduction

The flow over a sphere is analogous to the flow over a cylinder in some respects but presents significant differences due to axial symmetry rather than plane symmetry. Three-dimensional flows compared to two-dimensional flows present even more complicated kinematic and vortical interactions and therefore, remain less understood.

The flow around a sphere will adopt very different characteristics depending on the Reynolds number. For very low Reynolds numbers ($Re < 0.1$, called creeping flow) inertial forces are negligible and the streamlines are symmetric with respect to the center of the sphere. For $Re > 1.0$, the inertial effects become significant and the flow becomes asymmetrical fore and aft. At $Re = 24$, separation occurs at the rear of the sphere and a thin standing vortex ring is formed. The point of separation moves forward with increasing Re numbers until it reaches a stationary point at 81 degrees azimuthal from the forward stagnation point at $Re \cong 1.0 \times 10^4$. The wake will become fully developed for $Re \cong 100$ and at $Re \cong 140$ the vortex ring will start to be shed periodically. The oscillatory behavior will extend up to $Re \cong 2 \times 10^5$, with the wake becoming increasingly chaotic but keeping a laminar boundary layer on the surface of the sphere and a laminar separation point around 81 degrees azimuthal from the forward stagnation point. For

higher Re (Re > 2 x 10^5) the boundary layer will become turbulent and the separation point will move backward decreasing the size of the wake and sharply reducing the drag. The Reynolds number at which the boundary layer switches from laminar to turbulent flow is known as the critical Reynolds number and is characterized by a dramatic reduction of the drag. The overall effect of Reynolds number in the flow around a sphere is summarized in Figures 33, 34, 35 and 36 [36].

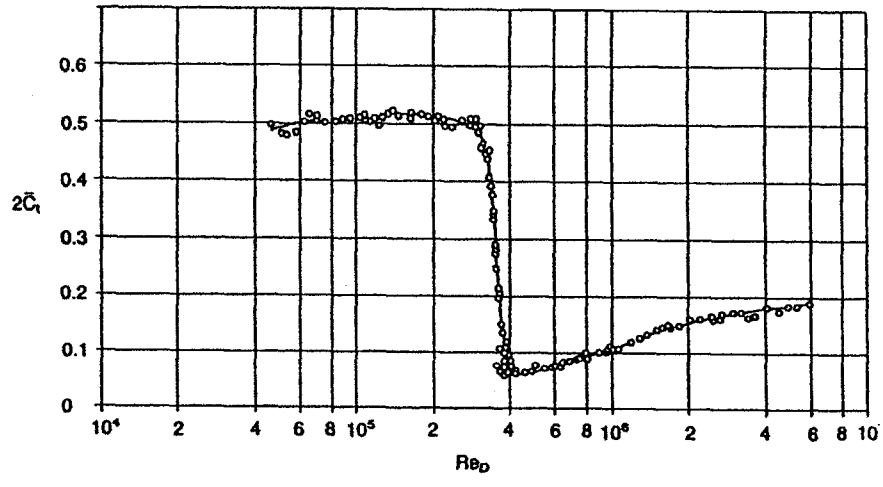

Figure 33. Experimental measurement of the drag coefficient of a sphere as a function of the Reynolds number. Reproduced from [36]

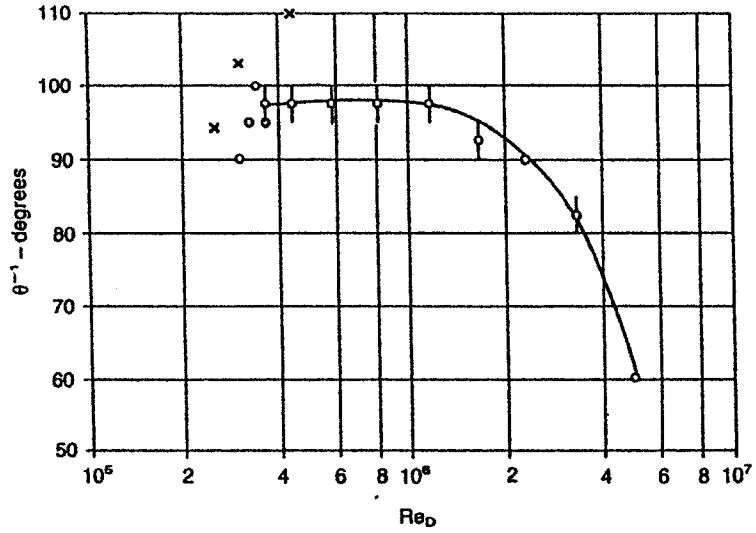

Figure 34. Experimentally measured transition angle as a function of the Reynolds number. Reproduced from [36]

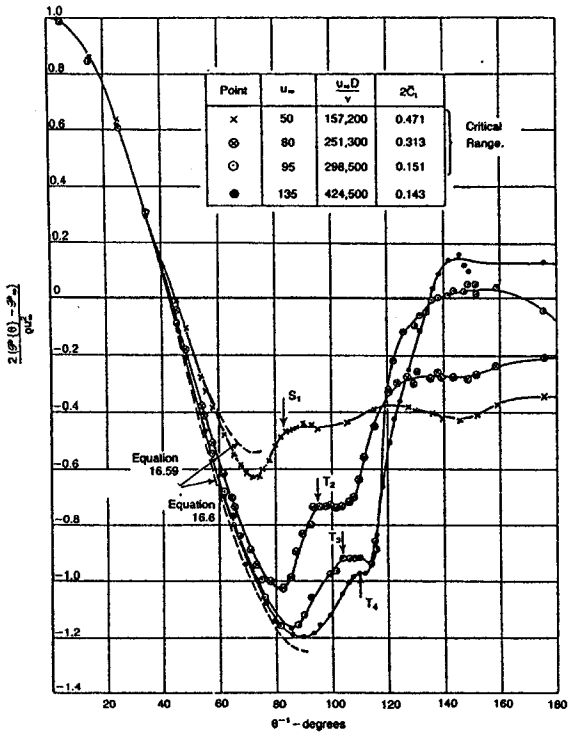

Figure 35. Experimental measurements of the pressure on the surface of a sphere for different Reynolds numbers. Reproduced from [36].

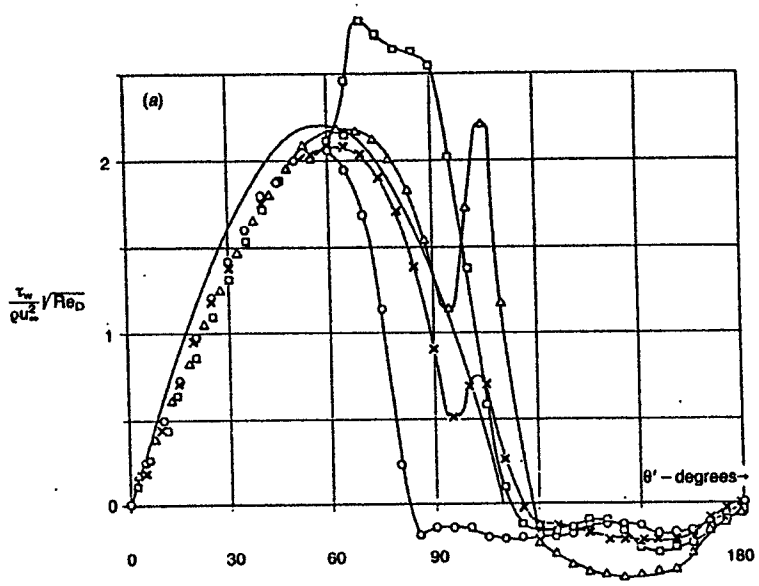

Figure 36. Experimental measurements of the skin friction on the surface of a sphere for different Reynolds numbers: –o–, Re=1.62x10^5; –x–, Re=3.18x10^5; –Δ–, Re=5.00x10^6. Reproduced from [36].

Extensive experimental studies of the sphere wake have shown the existence of a main instability mode related to the large scale shedding of the wake characterized by a Strouhal number of 0.185, which is practically constant in the range of Re=10^4 (the Strouhal Number is non-dimensionalized based on the free-stream velocity and the sphere diameter). Experiments have shown, that beginning at Re ≅ 800, a second mode of instability coexists with the main mode up to a threshold Reynolds number, with some disagreement about its value. Most of the experimental investigations captured both modes at Re=10^4. In this research only the main mode was captured.

From a computational point of view, to be able to accurately capture these instabilities the numerical scheme must resolve the small eddies and at the same time account for the large scale eddy shedding. Therefore, it is difficult to properly predict

this flow based on some form of the Reynolds-Averaged Navier Stokes (RANS) equations since, as was explained previously, the RANS approach fails for flows characterized by large regions of separation in which a very wide range of scales are present. A different approach, such as DES, must be used to capture the small scales and only model the Sub-Grid-Scales.

The flow around a sphere is the ideal case to test the DES approach. Constantinescu et al. [12] predicted the flow around a sphere for $Re=10^4$ in LES and DES cases using second-order and fifth-order upwind schemes for the convective terms in a structured mesh code. The most favorable agreement with the experimental results was obtained in LES and DES cases using the fifth-order discretization and the model coefficient $C_{DES}=0.65$. For all these computations, the transition location was forced using a 'turbulence index'. The index is zero in the laminar region and can be assigned values higher than one for fully turbulent regions. The effect of transition will be shown to be extremely important in the subsequent study.

In this research, the case studied will be the flow around a sphere at a Mach number of 0.2 and a Reynolds Number of 10,000. At this point, it is important to mention that the initial computations were made assuming fully turbulent flow, that is, not forcing the transition from laminar to turbulent flow at a designated point but instead running fully turbulent flow over the whole surface of the sphere to avoid dealing with transition. This produced solutions associated with higher Reynolds numbers than the modeled value of 10^4 with consequences in the pressure and skin friction distribution

over the surface of the sphere, as will be shown later. Similar effects were observed in Chapter IV for the flow over a circular cylinder.

The unsteady flow solutions obtained using URANS and DES will be compared focusing on the unsteady drag history and its frequency content, as well as mean pressure distribution and mean skin friction over the sphere surface. The effect of the artificial viscosity will be investigated based on the conclusions obtained from the decaying homogeneous turbulence in the periodic domain case presented in Chapter VI.

Computational Domain

The computational domain is a cubic box with an aspect ratio of 1 and a side length of 100 sphere diameters in all three directions. The center of the sphere is located in the center of the computational domain as shown in Figures 37. Figure 38 shows a detailed view of the mesh on the surface of the sphere.

The computational mesh is composed of 766,625 nodes. The mesh was generated using VGRIDns, a grid generator developed at NASA Langley Research Center [37]. VGRIDns uses the advancing layer method, allowing the specification of the initial normal coordinate for the first cell nearest a solid surface. To define grid spacing, VGRIDns uses "background sources" which can be placed anywhere in the computational domain. The mesh spacing at any location will depend on the distance to each source, source strength and source spacing. In this case, a single source was placed in the center of the sphere.

A multigrid convergence acceleration technique of four levels (Table 8) was utilized in all computations. All the computations were performed in parallel using 16 PIII 500-800 MHz processors of a PC cluster at ICASE. The unsteady calculations were performed using a time step of 0.05, where the time was non-dimensionalized using the freestream velocity and the sphere diameter. Thirty sub-iterations were used per time step to ensure a residual reduction of two orders of magnitude as shown in Figure 39. Each time step (of 30 sub-iterations per time step) took an average wall-time of 565 seconds.

Multigrid Level	Number of Nodes	Number of Edges
1	766,625	5,056,985
2	3,112	23,876
3	454	3,823
4	67	681

Table 8. Multigrid level description of the unstructured mesh used to compute the flow around a sphere. Mesh shown in Figure 37 and 38.

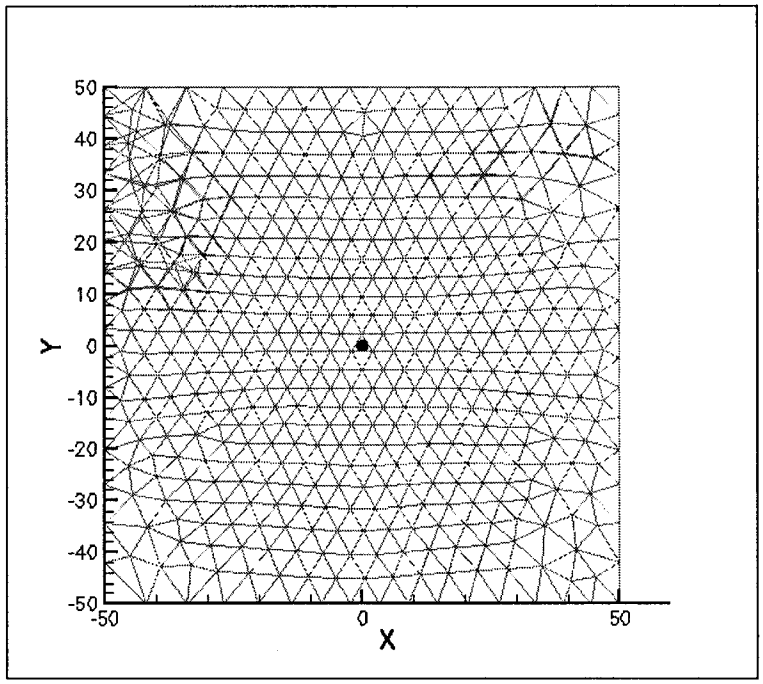

Figure 37. Two dimensional view of the computational domain.

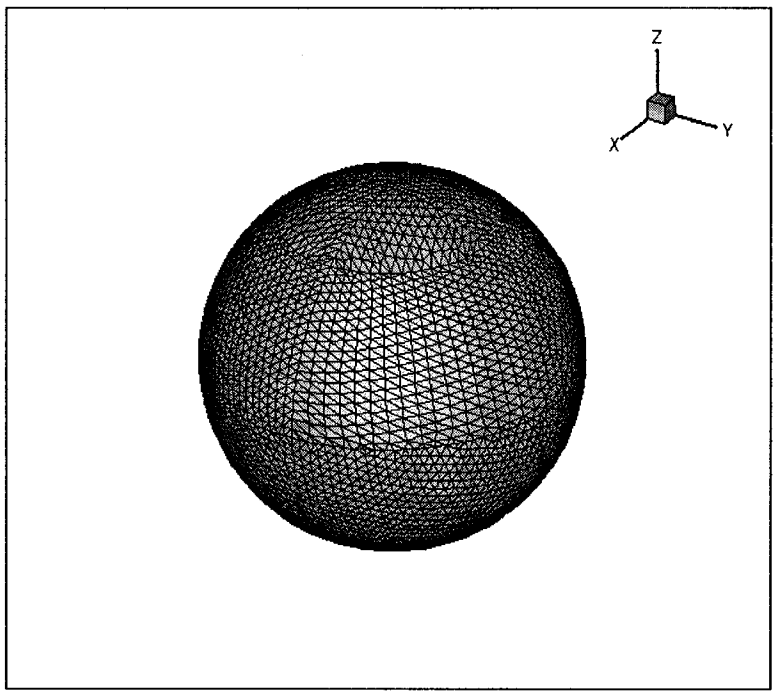

Figure 38. View of the surface mesh for the sphere test case

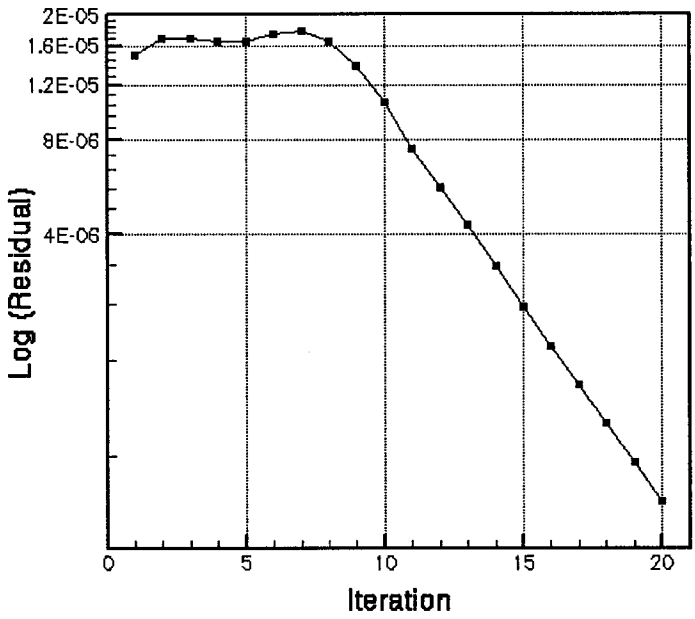

Figure 39. Sample convergence rate of the density residual for one time step and 20 sub-iterations obtained on grid of Figure 40 and 41 using four multigrid levels.

Steady Results

Following the procedure used by Constantinescu et al. [12], the accuracy of the numerical approach was established by comparing the results obtained with previous computational and experimental results for the steady flow regime at a Reynolds number 250. The drag coefficient was computed and the results obtained are compared to other simulation results and experimental data in Table 9. The agreement is satisfactory for all the cases tested. Because previous results were based on incompressible simulations, and the current solver is a density-based compressible formulation, the importance of compressible effects was also investigated by running the simulation at Mach numbers of 0.2 and 0.1, both with and without a low Mach number pre-conditioner. The low Mach number pre-conditioner is imperative for flows containing regions of low Mach number flow, such as the stagnation regions in this case. The low Mach number pre-conditioner

was implemented by modifying the dissipation terms in the residual as described in reference [38].

	Cd
Constantinescu et al. (2000)	0.70
Johnson and Patel (1999)	0.70
Experimental	0.70 - 0.72
M = 0.1	0.7141
M = 0.2	0.7014
M = 0.1 Low Mach Number pre-conditioner	0.6961
M = 0.2 Low Mach Number pre-conditioner	0.6950

Table 9. Computed Steady Drag Coefficient for Flow over Sphere at Re = 250 compared with Experimental and Previous Computational Values

Unsteady Results

For the unsteady runs, the flow around a sphere is computed at a Mach number of 0.2, without any additional low Mach number preconditioning, and a Reynolds number of 10^4. At this Reynolds number, the detached vortex sheet from the sphere is fully turbulent while the boundary layer on the sphere remains laminar. The Strouhal number associated with the vortex shedding at this Reynolds number is in the range of 0.185-0.200 depending on the investigation. The differences in the measurements of the Strouhal number of the different investigations is mainly due to the influencing parameters and the measurement techniques of the different investigations.

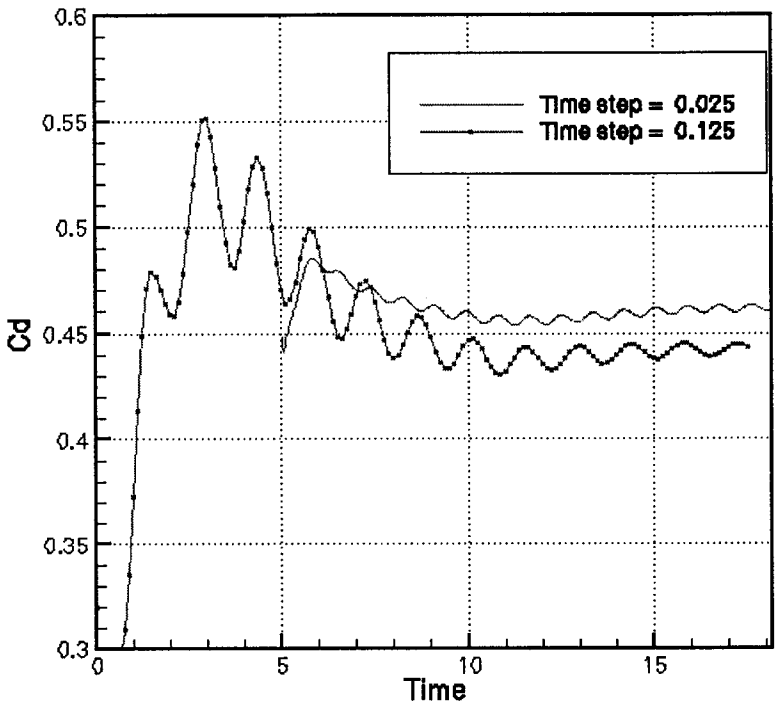

Figure 40. Time history of the force coefficients using RANS

The RANS run produced the expected results with a good average drag coefficient value close to 0.4, but with a very poor solution of the wake oscillations. The Spalart-Allmaras turbulence model is excessively dissipative in this region, suppressing all the small eddy effects and providing a non physical smooth time history of the drag coefficient with no frequency energy content information, as can be observed in Figure 40.

The Mach number contours depicted in Figure 41 corroborate the difference in the predicted flow using the regular Spalart-Allmaras URANS turbulence model and the detached eddy simulation (DES). DES exhibits a wider range of scales present in the

flow while regular RANS models tend to suppress the smaller scales. This is the effect expected from DES since the length scale redefinition increases the relative magnitude of the destruction term in the Spalart-Allmaras model, diminishing the importance of the eddy viscosity and allowing instabilities to develop.

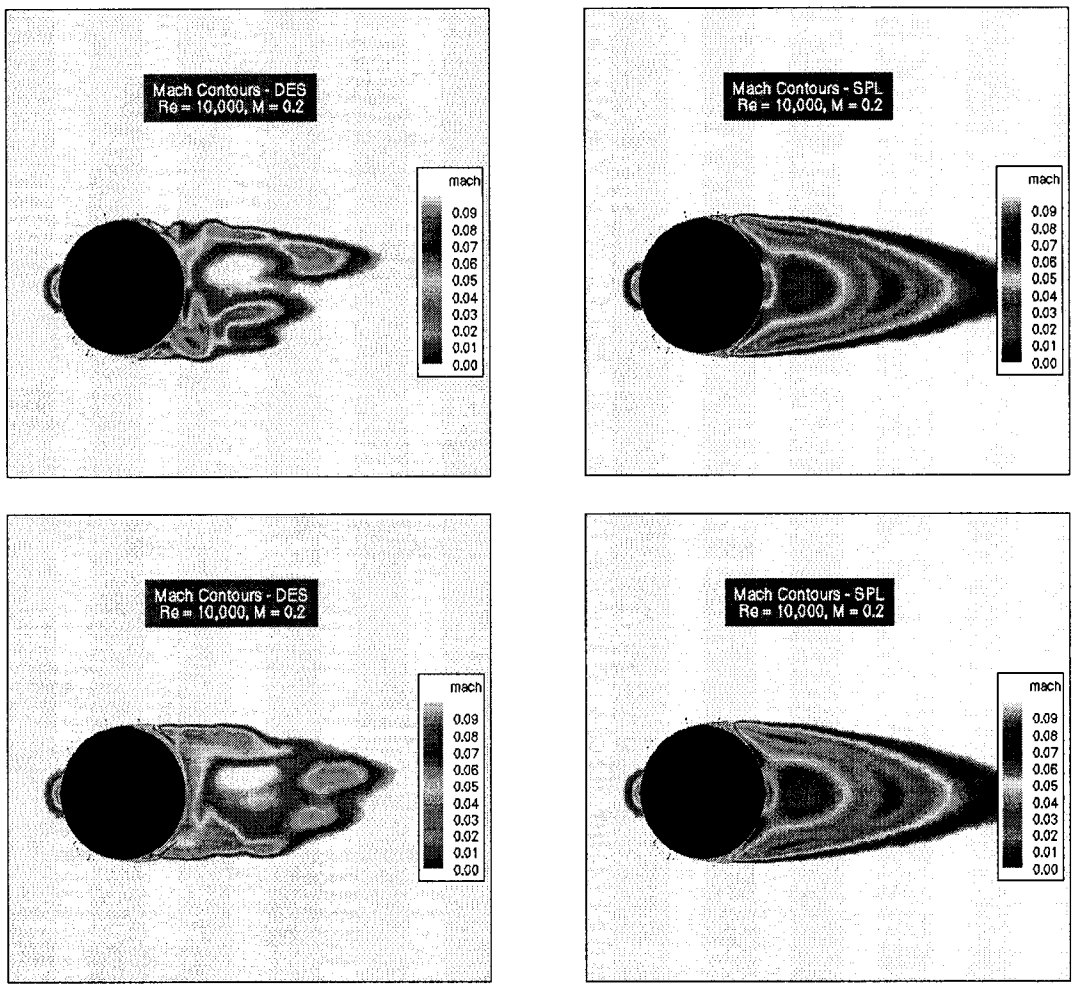

Figure 41. Comparison of Mach contours at different time snapshots computed using DES (left) and URANS (right).

Four different DES calculations were made, the first run was performed using the nominal scaling factor for the artificial dissipation and the second run using one fourth of

the nominal scaling factor value. As was demonstrated in Chapter VI, in the case of decaying homogenous turbulence in a periodic domain, one fourth of the nominal value of the scaling factor for the artificial dissipation yielded fairly good results in terms of accuracy and stability. However, reducing the levels of artificial dissipation produced solutions associated with higher Reynolds numbers than the one being targeted. Note that all the solutions computed for the flow around a sphere were computed in the fully turbulent mode to avoid dealing with transition from laminar to turbulent flow, thus producing solutions more closely associated with higher Reynolds numbers flow phenomena. This effect is particularly evident in this case, in which the solution obtained is extremely sensitive to the Reynolds number. This was observable in the shift in separation location and the pressure coefficient and skin friction distribution over the surface of the sphere. For all the calculations, a C_{DES} value of 0.65 was used.

The time history of the drag coefficient, the energy spectrum and the pressure coefficient and skin friction distribution obtained using the nominal value of the artificial dissipation are shown in Figures 46 through 49. A mean value of the drag coefficient of 0.433 was obtained as compared to the experimental value of 0.45. However, the energy spectrum of the streamwise drag coefficient reveals a peak corresponding to a Strouhal number of 0.10 compared to the 0.18 - 0.2 values reported experimentally. The artificial dissipation for this case is too dissipative, damping out most of the effects of the small eddies. The mean pressure and the mean skin friction distribution were computed by averaging over the azimuthal direction (19 test points separated by 10 degrees) and over at least 20 time units taking solutions every 0.5 time units (at least 40 solutions). The

pressure coefficient distribution and skin friction distribution (Figures 43 and 46) obtained for the DES run with nominal artificial dissipation are in good agreement with the experimental results (Figures 36 and 39). Figure 42 shows the pressure distribution over the surface of the sphere at a sample snapshot showing the necessity of integrating the pressure distribution in the azimuthal direction and in time.

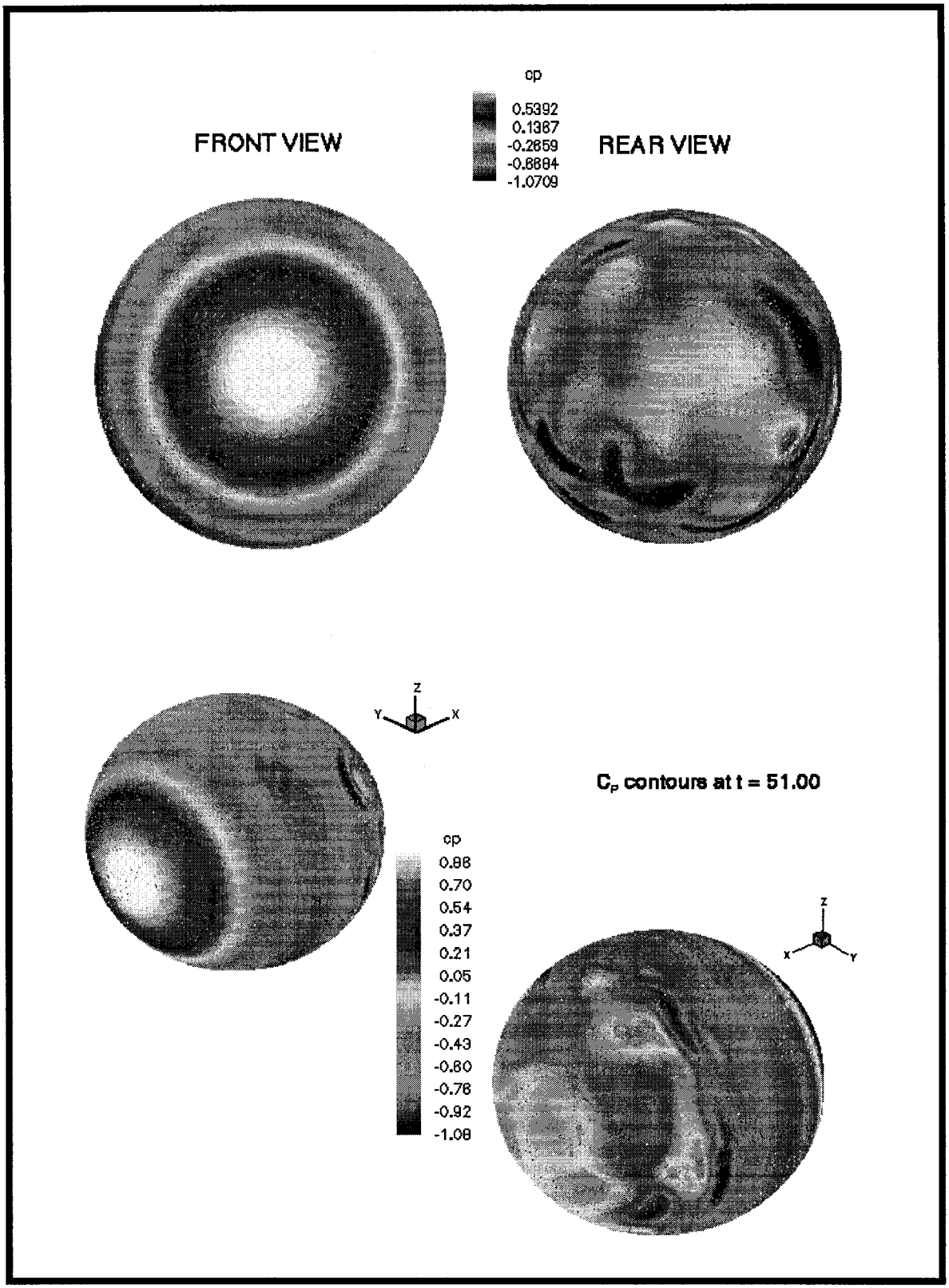

Figure 42. Sample pressure coefficient distribution over the surface of the sphere computed using nominal levels of the artificial dissipation. Mach number=0.2 and Re=10^4.

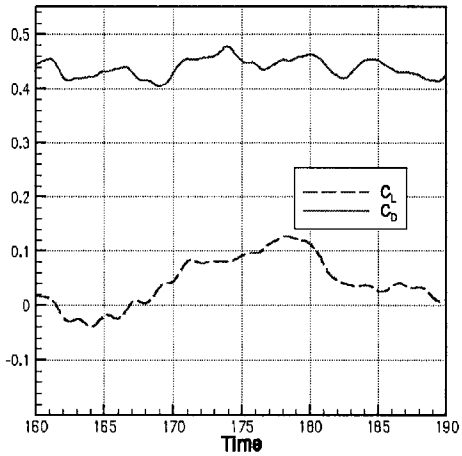

Figure 43. Sample time history of the lift and drag coefficients for nominal artificial dissipation.

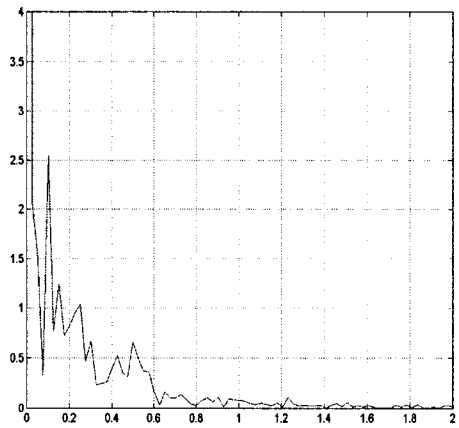

Figure 44. Power spectrum of the streamwise drag coefficient for nominal artificial dissipation.

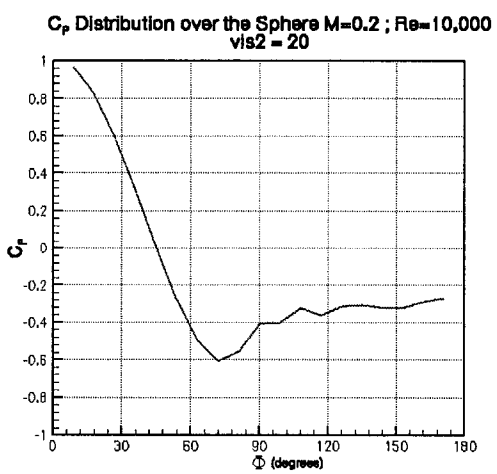

Figure 45. Mean pressure coefficient distribution over the sphere for nominal artificial dissipation.

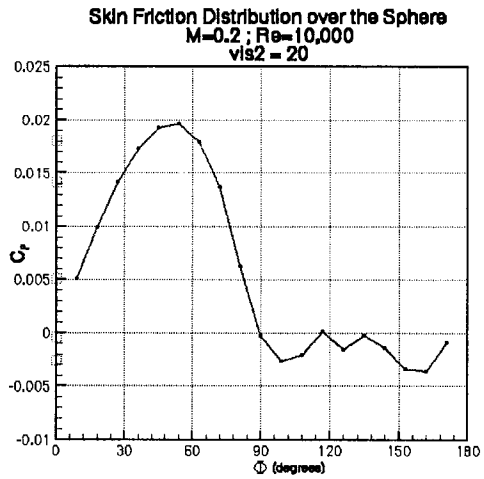

Figure 46. Mean skin friction distribution over the sphere for nominal artificial dissipation.

Utilizing one fourth of the nominal value of the artificial dissipation produced an improvement in terms of the frequency content of the wake, as can be observed in the energy spectrum of the drag time history (Figures 47 and 50) that yielded a Strouhal value of 0.143 as compared to the experimental value of 0.18-0.20. However, the pressure coefficient (Figure 49) and the skin friction (Figure 50) distribution deteriorate producing solutions similar to the distributions produced for higher Reynolds numbers. Reducing the artificial dissipation emphasized the fact that the run was fully turbulent, producing a shift of the separation point similar to critical Reynolds number situations as shown in Figure 35 and 36.

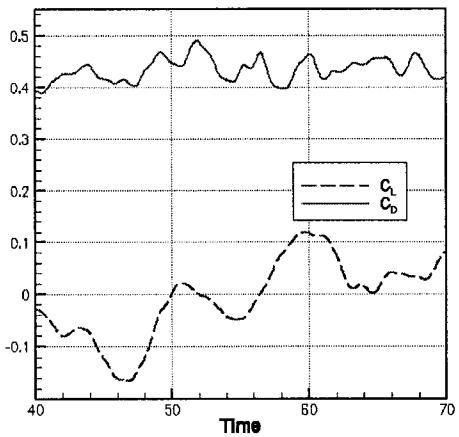

Figure 47. Sample time history of the lift and drag coefficients for a fourth of the artificial dissipation nominal value

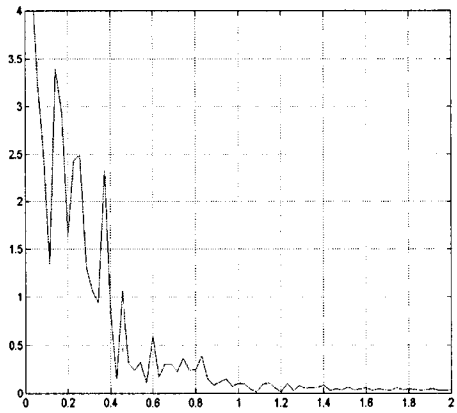

Figure 48. Power spectrum of the streamwise drag coefficient for a fourth of the artificial dissipation nominal value.

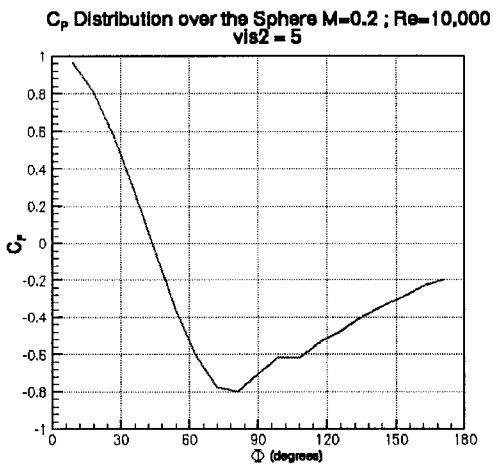

Figure 49. Mean pressure coefficient distribution over the sphere for a fourth of the artificial dissipation nominal value.

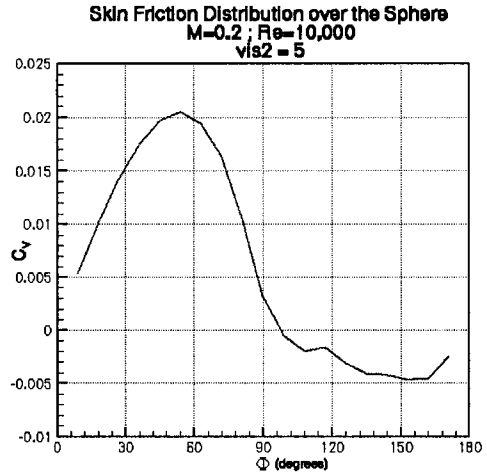

Figure 50. Mean skin friction distribution over the sphere for a fourth of the artificial dissipation nominal value

To be able to capture the instabilities associated with the small eddies and obtain simultaneously a good pressure coefficient and skin friction distribution over the sphere, it was necessary to force laminar separation by enforcing laminar flow ahead of the 90 degree azimuthal location (measured from the front stagnation point). The results obtained are shown in figures 51 through 54.

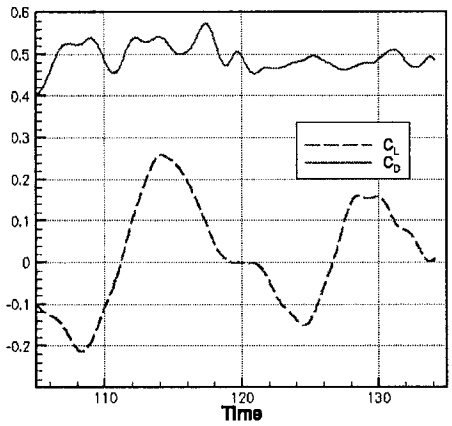

Figure 51. Sample time history of the lift and drag coefficients for a fourth of the artificial dissipation nominal value and laminar flow enforced ahead of 90 degrees azimuthal.

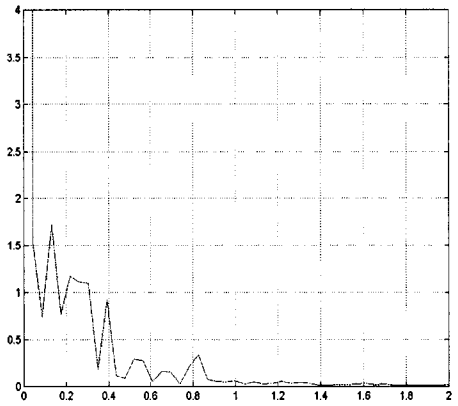

Figure 52. Power spectrum of the streamwise drag coefficient for a fourth of the artificial dissipation nominal value and laminar flow enforced ahead of 90 degrees azimuthal.

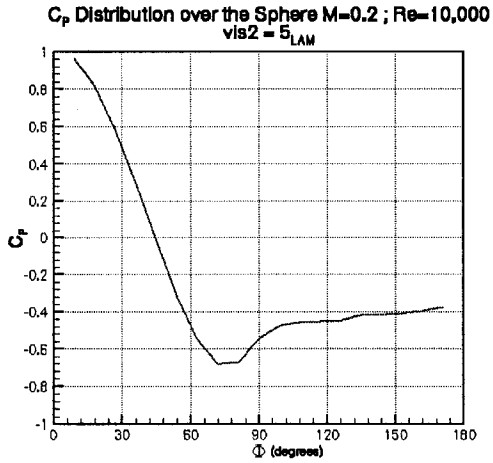

Figure 53. Mean pressure coefficient distribution over the sphere for a fourth of the artificial dissipation nominal value and laminar flow enforced ahead of 90 degrees azimuthal.

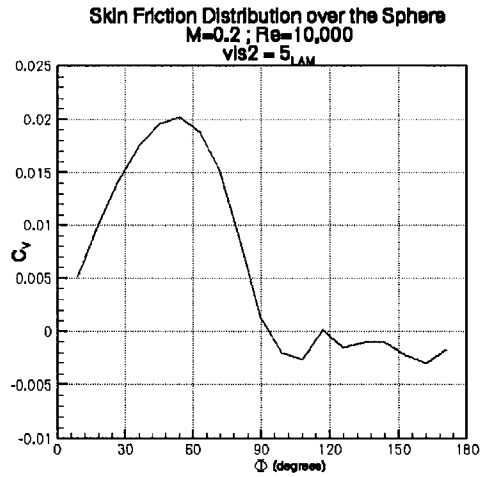

Figure 54. Mean skin friction distribution over the sphere for a fourth of the artificial dissipation nominal value and laminar flow enforced ahead of 90 degrees azimuthal.

Figure 55 highlights the effect of the artificial dissipation in the solution as it shows the transition in the computed lift and drag coefficient time history when the value of the scaling factor of the artificial dissipation is switched to a fourth of its nominal value at 100 time units. In both cases, the eddy viscosity was forced to zero (laminar) ahead of the 90-degree azimuth.

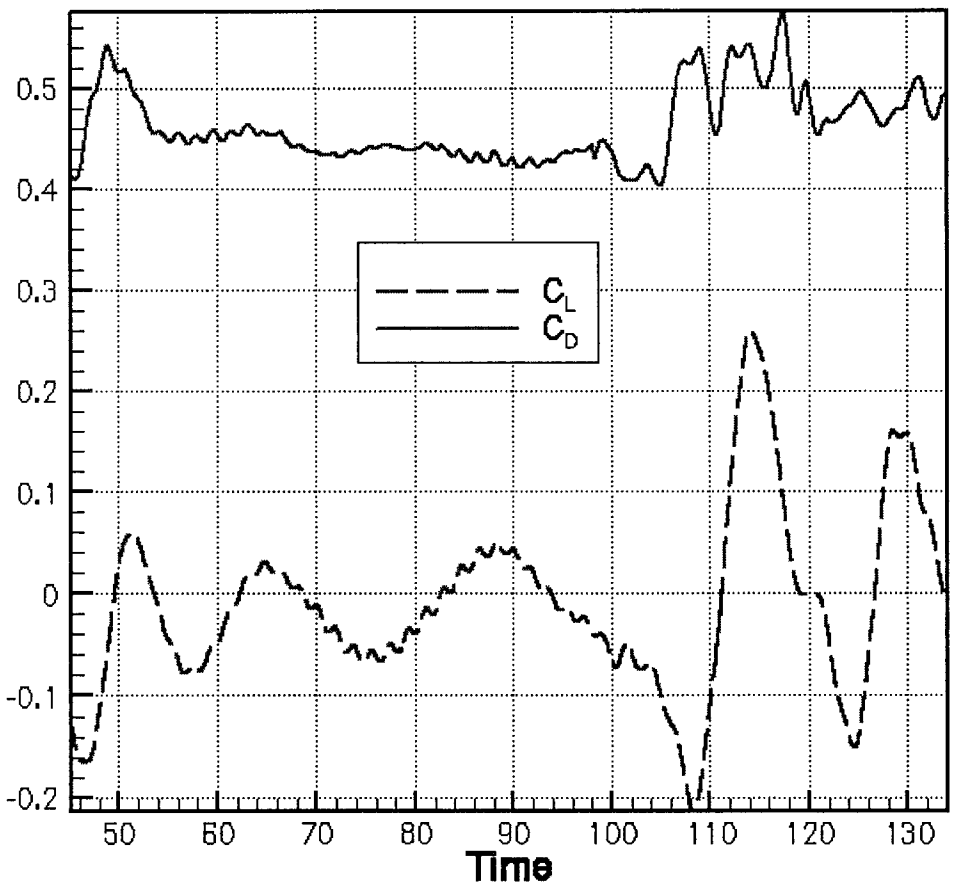

Figure 55. Time history of the lift and drag coefficient when transition from artificial dissipation nominal levels to ¼ of nominal levels is forced at t=100.

Conclusions

The time history of the drag coefficient reveals important differences between URANS and DES. The mean value of the drag coefficient in all cases is close to the experimentally reported value of 0.40. However, the frequency content in each case is completely different. The URANS simulation appears to damp out most of the oscillations present in the DES runs, while the DES runs show a very chaotic oscillatory pattern quite similar to the solutions obtained by Constantinescu et al [12]. Spectral analysis of the time-dependent drag coefficient history reveals a peak corresponding to a Strouhal number ranging between 0.1 and 0.143 as summarized in table 10.

Artificial Dissipation	Strouhal Number	Cd
Experimental	0.18-0.20	0.450
Nominal	0.1000	0.433
½ Nominal	0.1300	0.426
¼ Nominal	0.1429	0.440
¼ Nominal_Laminar	0.1400	0.458

Table 10. Summary of results of the Strouhal number and averaged drag coefficient computed for the flow over a sphere at M=0.2 and Re=10^4 for different levels of artificial dissipation.

In an effort to completely understand the differences between the solutions obtained using the nominal value of the artificial dissipation and a fourth of the nominal value, a new run was performed using half the nominal value of the artificial dissipation factor. The results obtained are included in Table 10 and, as expected, confirm the trend observed for previous runs using the nominal value and a fourth of the nominal value.

Predictions of the mean pressure distribution and mean skin friction distribution over the surface of the sphere are shown in Figures 56 and 57. The surface pressure distribution for nominal values of the artificial dissipation is in good agreement with the experimental results at Re=157,200 in Figure 35, which is in agreement with the results reported by Constantinescu [12]. The pressure distributions obtained using DES and reduced values for the artificial dissipation scaling factor shows a degradation of the solution producing results associated with higher Reynolds numbers than the specified value, as can be observed comparing Figure 56 and Figure 57. The effects of running fully turbulent are magnified by the reduction of the artificial dissipation and it becomes necessary to force laminar flow (zero eddy viscosity) ahead of the 90^{o} location to obtain better results for the pressure and skin friction distribution and good Strouhal number prediction. While the initial goal of this case did not include dealing with transition from laminar to turbulent, the necessity of enforcing laminar separation became apparent when results revealed shifting of the separation point associated with artificial dissipation levels. This stresses one of the main concerns of DES, which is that it relies on a RANS turbulence model to determine transition and separation.

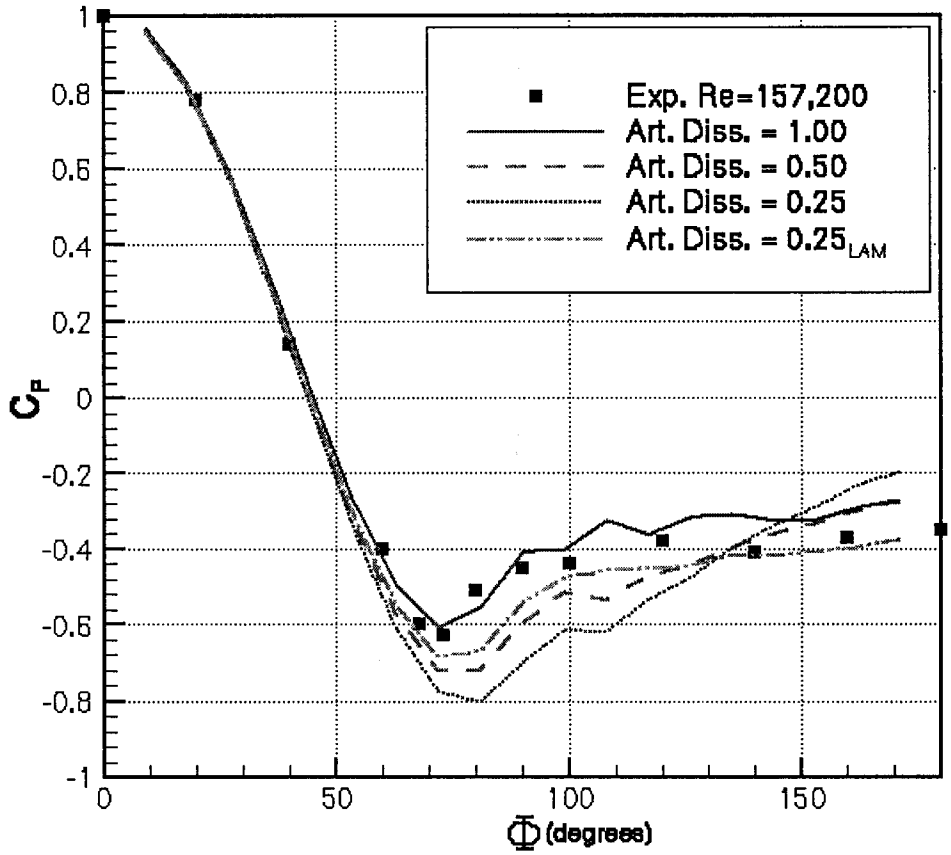

Figure 56. Summary of pressure coefficient distribution results computed for the flow over a sphere at M=0.2 and Re=10^4 for different levels of artificial dissipation.

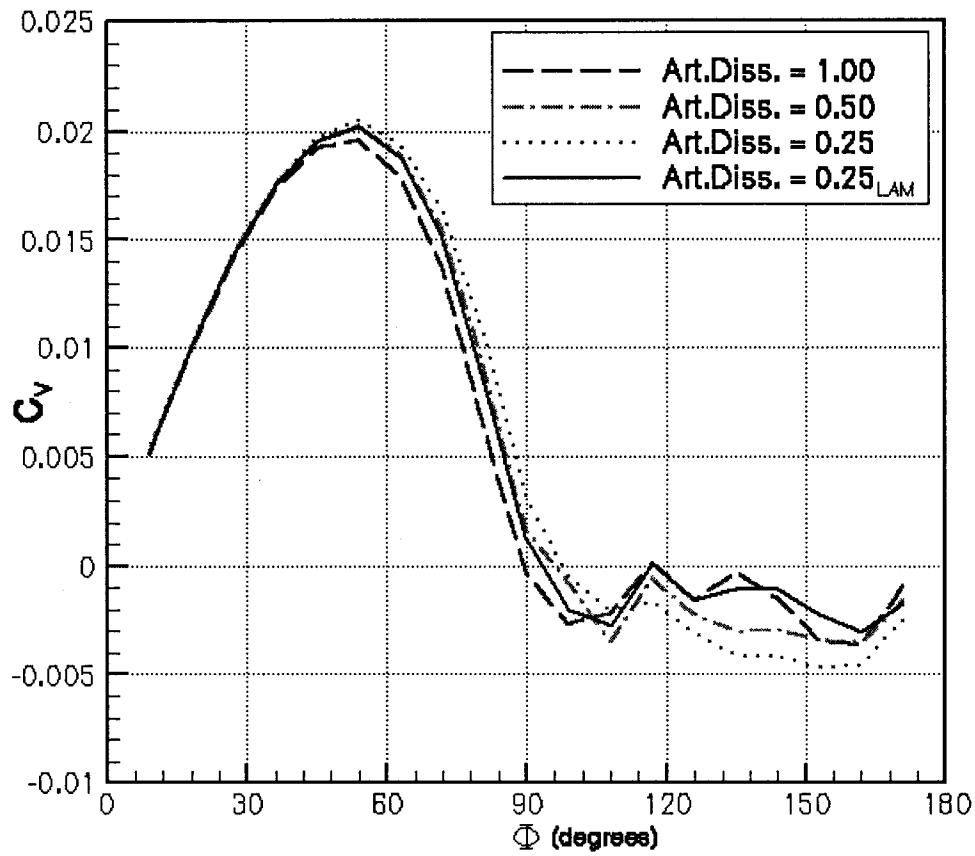

Figure 57. Summary of skin friction distribution results computed for the flow over a sphere at M=0.2 and Re=10^4 for different levels of artificial dissipation.

CHAPTER VIII

FLOW OVER A WING

Introduction

In this case DES is used to compute the flow around a NACA 0012 wing. The objective was to test the DES technique as compared to URANS for flows of aeronautical interest, such as the stall and post-stall regimes of an airfoil. The NACA 0012 is an airfoil that has been studied widely up to its stall angle of attack around 15 degrees. For its post-stall characteristics we will rely on previous computational data [10] and some experimental data [39].

Test Description

URANS and DES were used to compute the flow over a NACA 0012 airfoil at a Reynolds number of 10^5 and a Mach number of 0.25. All the runs performed for the wing case can be organized in two categories:

- A first set of runs was performed to study the differences between the computed stall characteristics of the NACA 0012 obtained using DES and URANS. This included nine test points in the linear pre-stall, stall and post-stall regime up to a 16 degrees angle of attack (AoA = 4, 6, 8, 10, 11, 11.5, 12, 14, 16 degrees). Following the procedure of Shur et al. [10], all cases were computed using the turbulence model in

the fully turbulent mode to avoid dealing with transition prediction issues. Obviously, considering the strong dependence on the Reynolds number of the stall characteristics, it was not expected to obtain the exact stall angle and Cl_{max} value. However, the objective was to compare DES and URANS to identify major differences, assuming the errors derived from the transition issues would be similar in both cases.

- A second set of tests was devised to test DES at high angles of attack by attempting to reproduce the results obtained by Shur et al [10] which showed that DES was able to obtain accurate results of C_L and C_D for 45, 60 and 90 degrees of angle of attack, while URANS over-predicted these values by 50% as compared to the experimental results.

Computational Domain

The computational domain consists of a box of 30 chord-lengths in the x- and y-directions and 2 chord-lengths in the spanwise direction. The NACA 0012 airfoil was located with its leading edge along the z=0 line. Several views of the computational domain are shown in Figures 58 through 61.

A mesh of 1,231,667 points was used with a grid spacing normal to the solid boundary of the first grid point closest to the wall of 10^{-5} chords. VGRIDns [37] was used for the generation of this mesh. As stated in Chapter VII, the mesh spacing at any location will depend on the distance to each source, source strength and source spacing. In this

particular case, a line of uniform sources was placed along the leading edge and another line of uniform sources was placed along the trailing edge. A convergence acceleration multigrid technique of four levels was used in the flow solver. The details for each multigrid level are shown in the table below.

Multigrid Level	Number of Nodes	Number of Edges
1	1,231,667	6,203,383
2	10,517	65,934
3	1,565	8,832
4	229	1,349

Table 11. Multigrid level description of the unstructured mesh used to compute the flow over a NACA 0012 airfoil. Mesh shown in Figure 58, 59, 60 and 61.

The boundary condition at the end walls is "slip" (inviscid) while on the wing surface the no-slip boundary condition is enforced. All runs were computed as fully turbulent to avoid having to trigger the transition point.

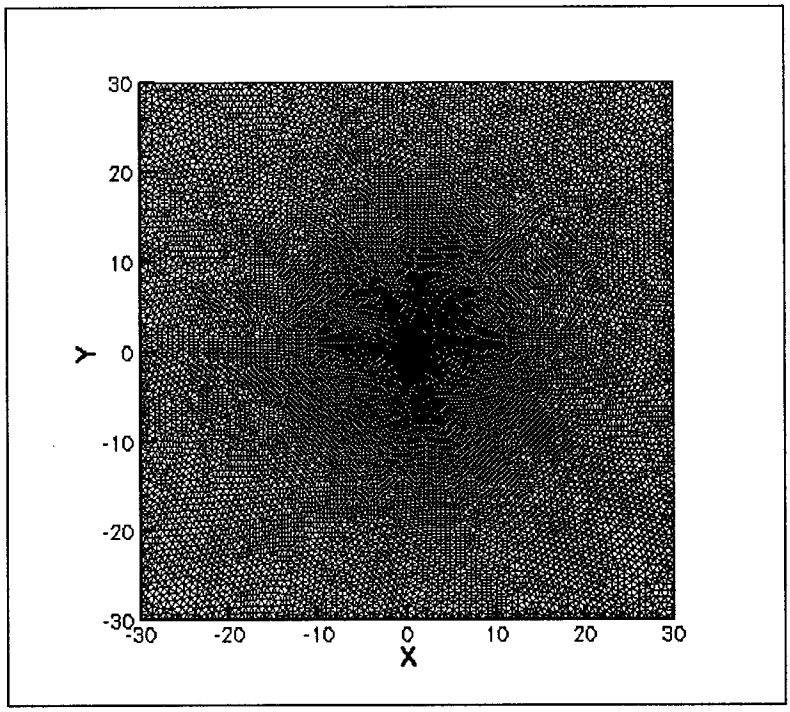

Figure 58. Two-Dimensional side view of the computational domain for the NACA 0012 wing case. Unstructured grid of 1,231,667 points.

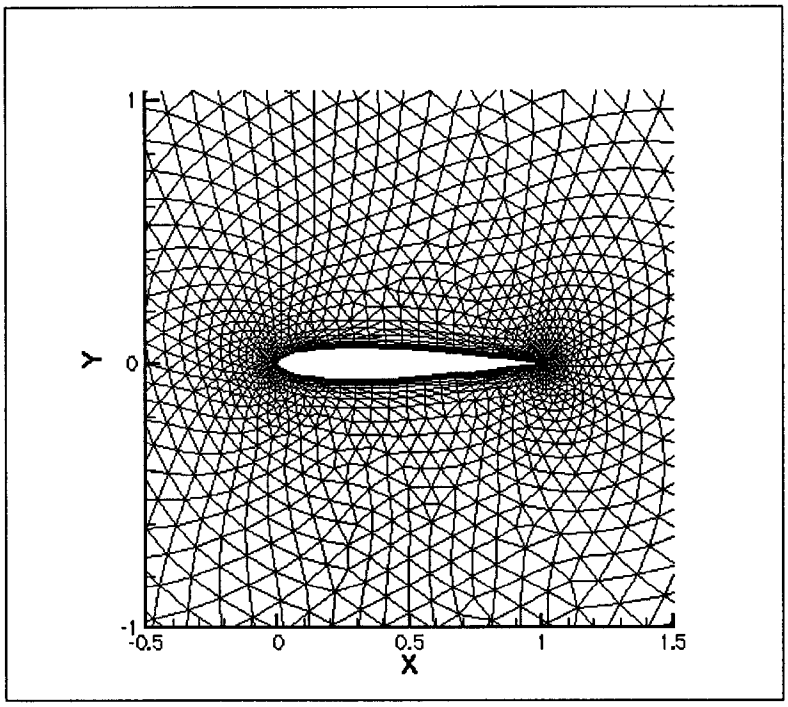

Figure 59. Two-Dimensioanl side view of unstructured grid used for computation of flow over NACA 0012 wing. Number of points=1,231,667, Wall resolution=1x10 $^{-5}$ wing chords.

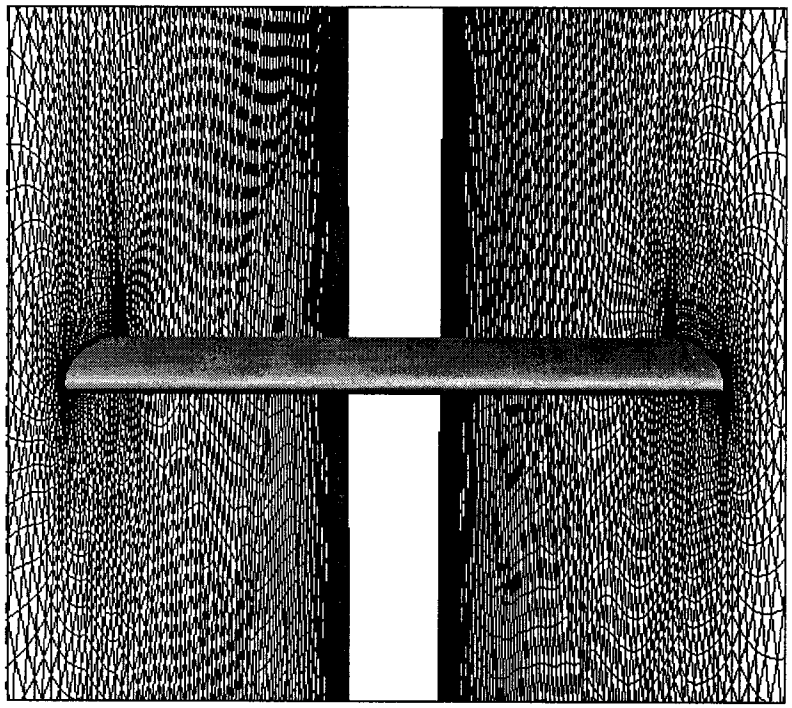

Figure 60. Three-Dimensional view of the computational domain for NACA 0012 wing case. Unstructured mesh of 1,231,667 points.

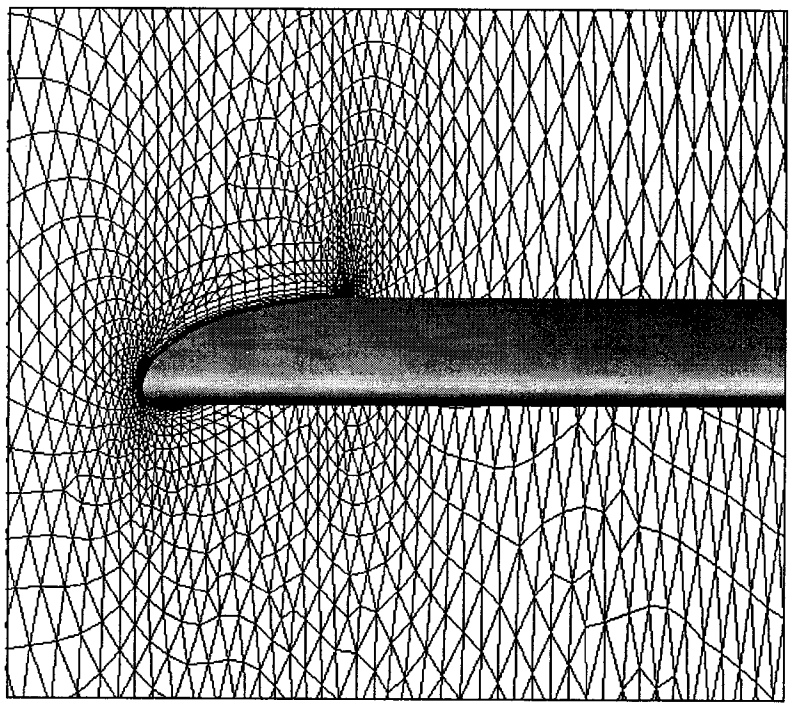

Figure 61. Three-Dimensional detailed view of the wing-wall intersection

In this case, due to time and computing constraints, only one time step of 0.25 x (c/U_∞) was used, where c is the airfoil chord. The calculations were carried out for a maximum of 50 time units for the post-stall cases, where a time unit represents the time it takes for the undisturbed far-field flow to travel one chord length. All the computations were performed in parallel using 16 PIII 400-800 MHz processors of a PC cluster at ICASE

Results for Stall Tests

The time history of the lift and drag coefficient show good agreement between URANS and DES for angles of attack below 11.5 degrees. This was expected since for pre-stall conditions the DES model operates primarily in the URANS mode. However, for the post-stall condition, i.e. angles of attack over 11.5°, the time history of the force coefficients obtained using URANS and DES showed differences similar to the ones observed for the sphere case. The time history of the DES results shows higher unsteadiness than URANS, indicating that more scales are being captured in the separated region, as can be observed in Figures 62 and 63, for sample angles of attack.

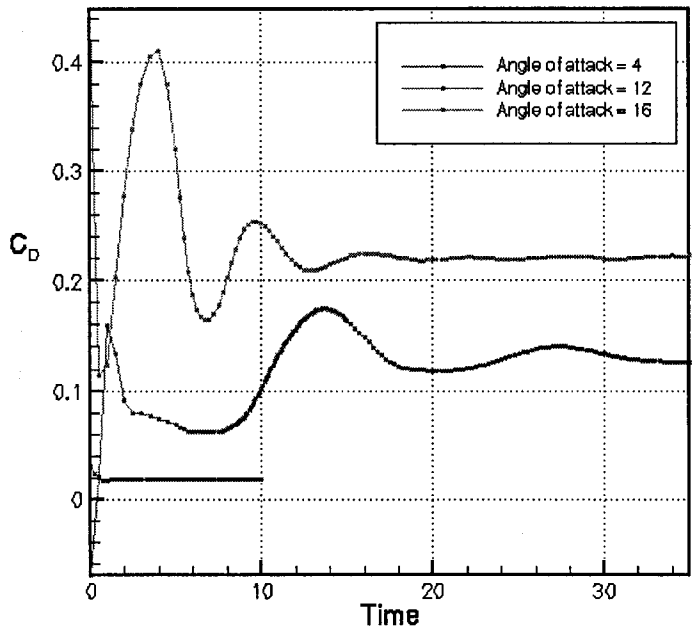

Figure 62. Sample URANS Drag Coefficient Time History for Flow over NACA 0012 Wing at various Angles of Attack. Mach number=0.25, Re=10^5.

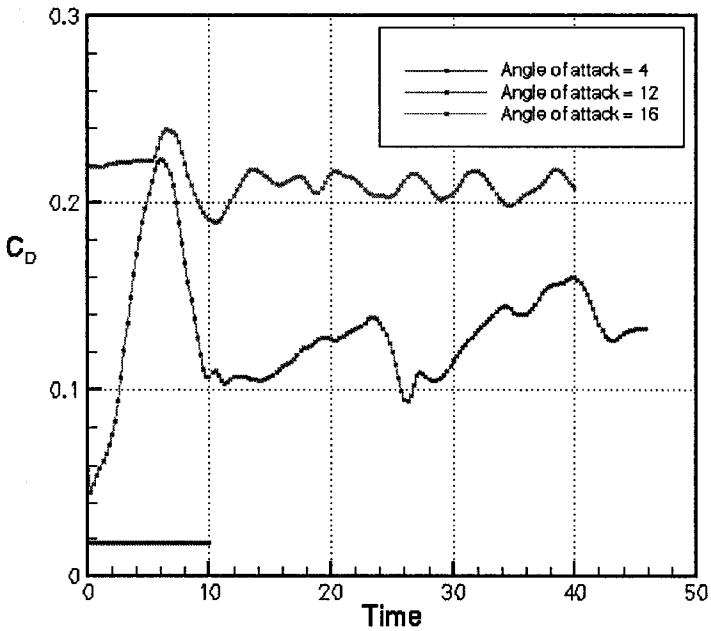

Figure 63. Sample DES Drag Coefficient Time History for Flow Over NACA 0012 Wing at various Angles of Attack. Mach number=0.25, Re=10^5

Figure 64 shows four snapshots of the Mach contours computed using URANS and DES for angles of attack of 12 and 16 degrees. Note that these snapshots are cuts of the computational domain at z=1 and they do not represent similar time frames.

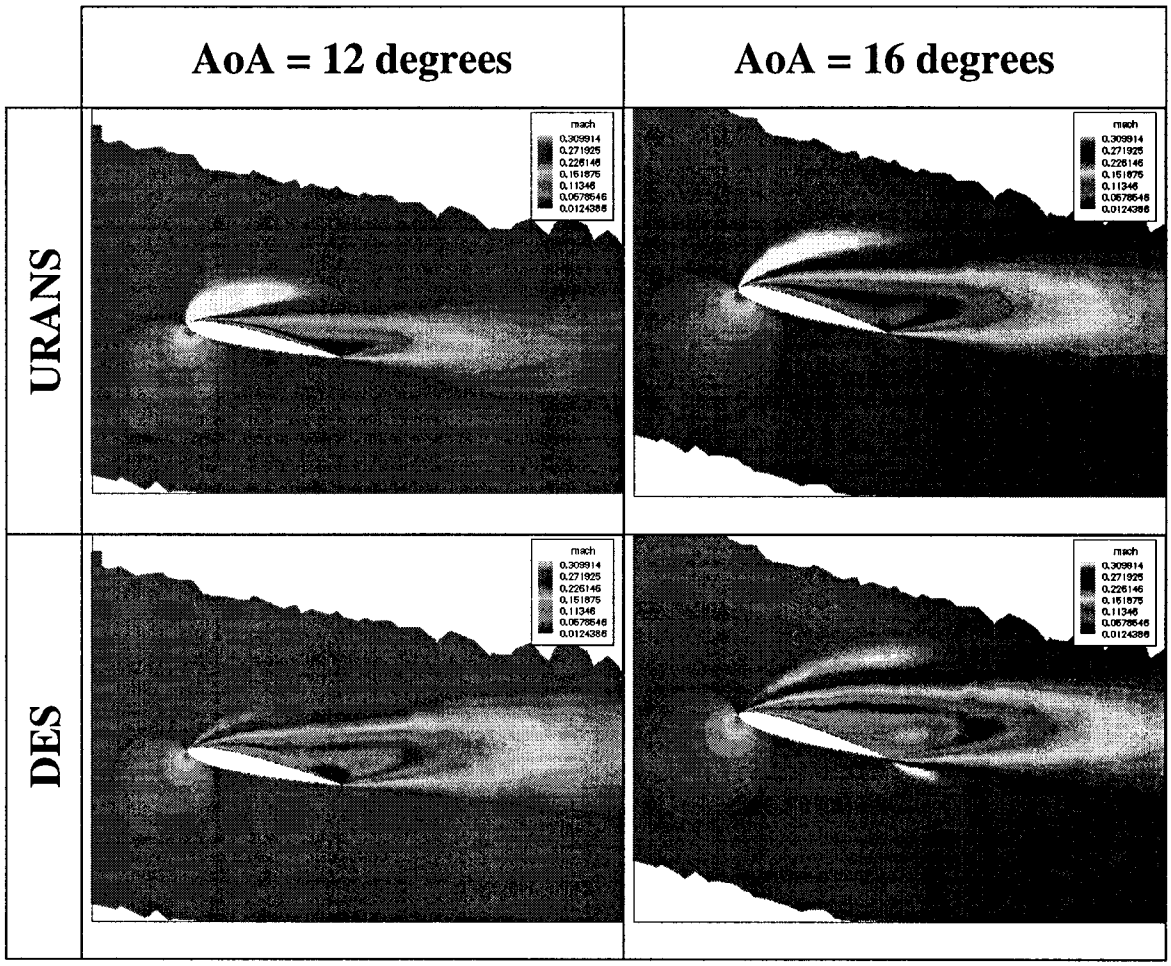

Figure 64. Sample Mach contours computed using URANS and DES at 12 and 16 degrees angle of attack. Mach number=0.25, Re=10^5.

The averaged lift and drag coefficient curves with respect to angle of attack are shown in Figures 65 and 66. The DES simulation predicts a more severe stall than the URANS results, i.e. lower post-stall lift and higher drag. This is consistent with the DES results obtained by Shur et al. [10], who computed an even more severe stall than the one

obtained in this research. Both computations, URANS and DES, stall at the same incidence angle of 11 degrees as compared to the 12 degrees computed by Shur et al. All these results must be put in perspective taking into account the crucial effect on separation of transition, which is neglected here by assuming fully turbulent flow over the whole surface of the wing. With that in mind, it is concluded that comparison between URANS and DES with experimental data is not sufficiently close to favor agreement for one method over the other. DES is expected to show improvements at higher angles of attack for massively separated flow conditions, but it relies on URANS to predict separation and near-stall regimes. This dependency on RANS to predict transition and separation is one of the major limitations of DES which needs to be addressed in future investigations.

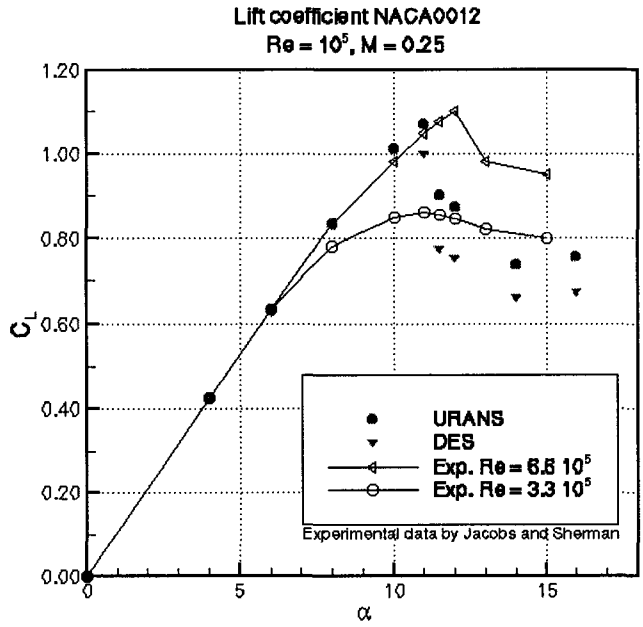

Figure 65. Comparison of Computed Lift Coefficient versus Angle of Attack for URANS and DES versus Experimental Data at two Different Reynolds Numbers

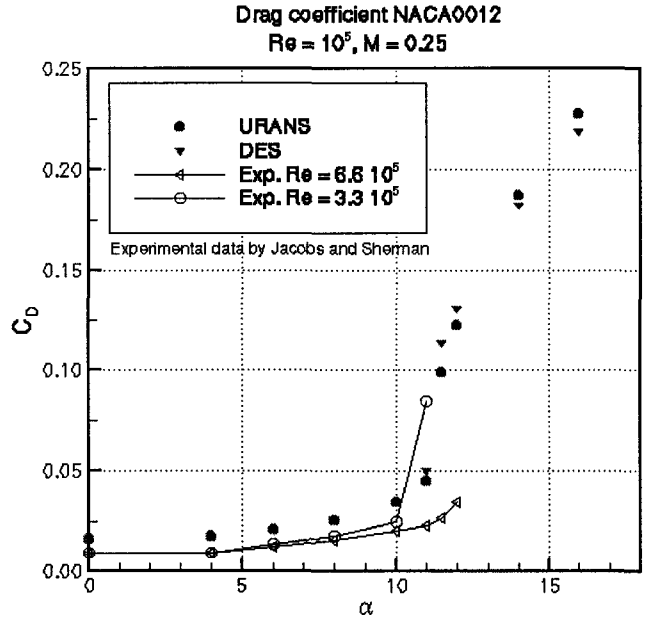

Figure 66. Comparison of Computed Drag Coefficient versus Angle of Attack for URANS and DES versus Experimental Data at two Different Reynolds Numbers

Results for Post-Stall Tests (AoA=60 degrees)

After concluding that DES and RANS predicted similar separation and little differences in near-stall results, it was decided to test DES at higher angles of attack since in massively separated conditions DES should produce more valid results than RANS. Previous studies [10] had shown that DES was able to accurately predict C_L and C_D at 60 degrees incidence while RANS over-predicted this case by 50%.

Initial runs of the wing case at 60 degrees angle of attack were performed using the same mesh used for the near-stall tests producing surprisingly poor results, very close to the results obtained by previous URANS calculations. The time history of the lift and drag coefficients for an angle of attack of 60 degrees is shown in Figure 67. The results appear closer to URANS results than to the expected DES values obtained by Shur et al. and to the experimental values of 0.90 for the lift coefficient and 1.625 for the drag coefficient.

Figure 68 shows a snapshot of the Mach contours obtained using DES for an angle of attack of 60 degrees incidence.

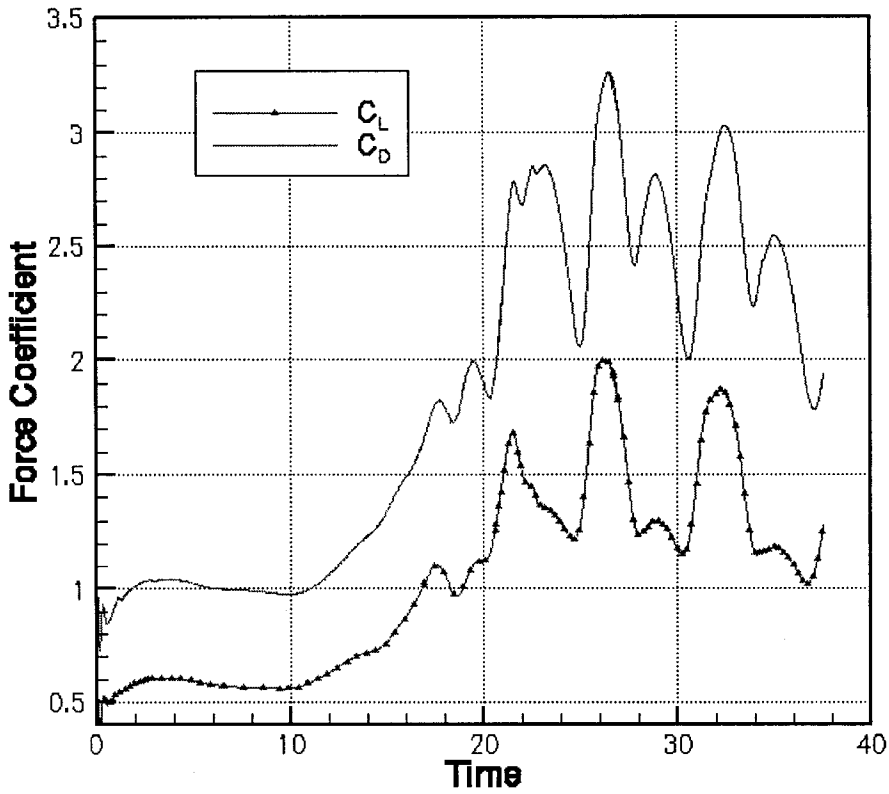

Figure 67. Lift and drag coefficient time history for 60 degrees of AoA. Computation performed on unstructured mesh of 1,231,667 nodes. Mach number = 0.25, Re = 10^5.

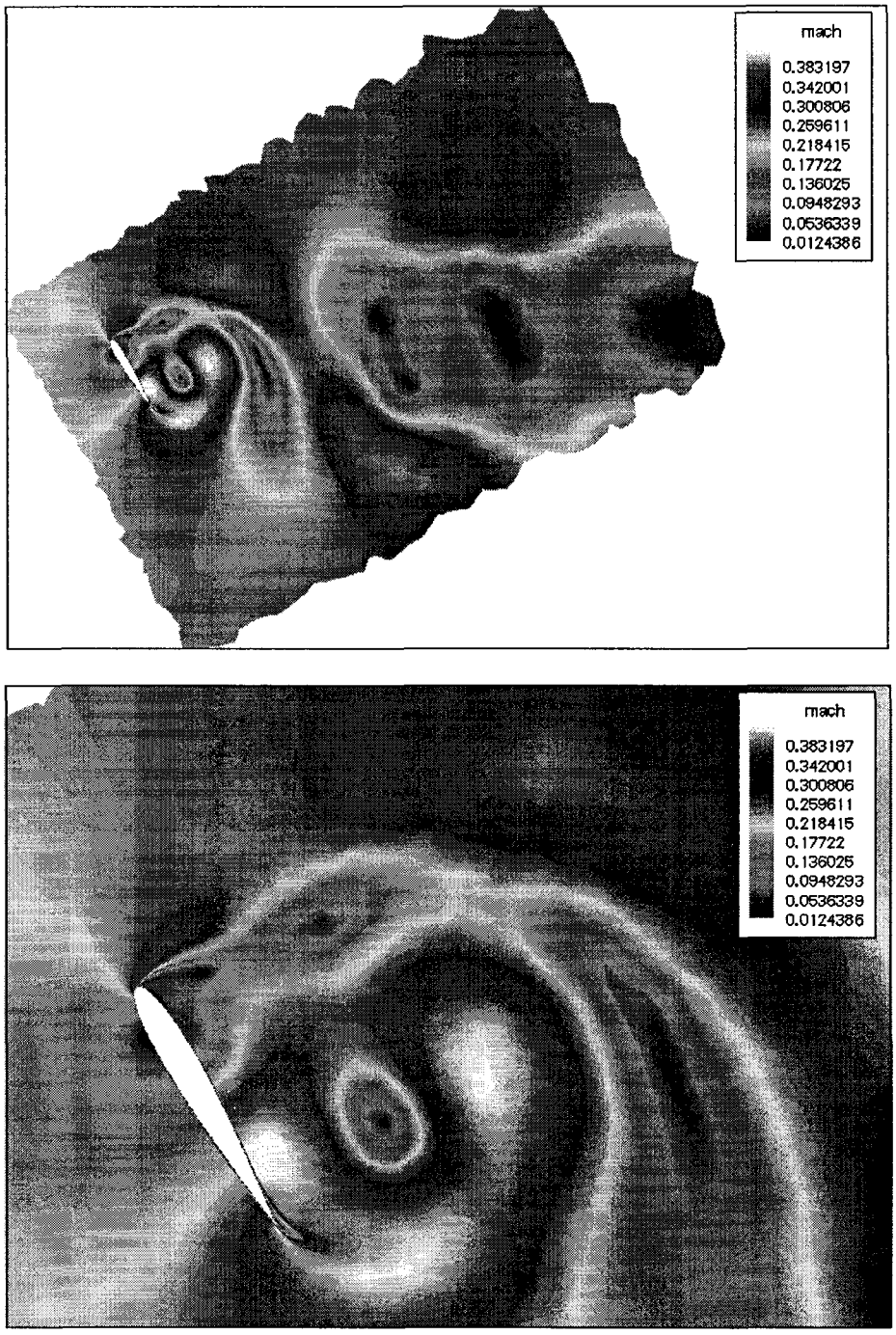

Figure 68. Sample Mach contours of the flow around a NACA 0012 at 60 degrees AoA. Computation performed on unstructured mesh of 2,107,026 nodes. Mach number = 0.25, Re = 10^5.

The averaged computed C_L and C_D are summarized and compared to the DES results obtained by Shur et al. [10], the URANS results, and the experimental results in Table 12.

	C_L	C_D
Experimental	0.90	1.625
Shur et al. (DES)	1.000	1.625
URANS	1.300	2.250
DES (original mesh)	1.520	2.540

Table 12. Comparison of computed lift and drag coefficients obtained for AoA=60 degrees, Mach number = 0.25 and Re = 10^5 with experimental values and previous computational results.

Conclusions

DES was tested in a case of aeronautical interest for flow around a NACA 0012 wing. The tests cases were grouped around the stall regime (9 test cases from 0 to 16 degrees angle of attack) and for the massively stalled regime (60 degrees angle of attack).

The objective of the near-stall tests was to compare the stall prediction capabilities of traditional RANS and DES methods. The results did not show significant differences between the computed RANS and DES results. This is quite reasonable if one considers that before stall DES acts merely as a RANS method, and in addition, DES relies on RANS to predict separation and transition is not modeled. This was already identified as a major limitation of DES by Spalart [6] and needs to be addressed in future investigations.

The second set of tests tried to reproduce the very promising results obtained by Shur et al. [10], which accurately computed the lift, and drag coefficients for a NACA 0012 wing profile at 60 degrees angle of attack using DES, while RANS over-predicted these values by up to 50%. The DES runs, computed in this investigation, over-predicted the results similarly to the URANS calculations, showing no improvement between DES and RANS. Examining the mesh it was concluded that the mesh was not dense enough in the region of interest. This test highlighted the importance of mesh resolution in the DES calculation. A coarse mesh will prevent DES from displaying its full capability and a RANS solution will be recovered. Mesh quality is crucial for a successful DES run. This raises the issue of self-adaptive meshing techniques to refine the mesh in the regions where it is necessary, automatically solving the problems associated with inadequate mesh resolution. This is a line of research that needs to be investigated that will be addressed in Chapter IX.

CHAPTER IX

FUTURE WORK AND RECOMMENDATIONS

In this chapter four lines of future research that can increase the capabilities of DES are presented. These potential fields of investigation include the following: adaptive artificial dissipation, adaptive meshing, higher order methods in spatial and time discretization and hybrid RANS/LES methods using different RANS turbulence models.

Adaptive artificial dissipation

As was stated in the previous chapters, the motivation for the DES technique was to find a general approach to solve flows characterized by large regions of separation in which a very wide range of flow scales are present and the traditional RANS approach fails. The lack of generality of RANS models restricts their ability to predict the effects of large scales contained in these flows, which are associated with the boundary conditions for each case. This motivated techniques such as LES, which solves the large scales and models the small scales, or DNS, which resolves the entire flow at all scales. The complications in both cases are obvious when the scales that must be resolved are too small and require very fine grids and very small time steps. In this scenario, DES finds its place as a technique that combines the best characteristics of the RANS and the LES approaches. DES acts as a RANS technique when close to a solid boundary and as an LES method for solving the flow away of the boundary layer. Notice that DES is of special interest when solving flows with large regions containing different scales, as is

the case for massively separated flows, where DES can take advantage of its LES nature. DES is a technique which is designed for resolving turbulent flows with a fairly good level of detail and it is important to consider and ensure that the artificial dissipation associated with the numerical scheme is not diffusing the eddies that the DES approach is trying to capture. Note that excess dissipation would not produce meaningless solutions, but would prevent the resolution of the flow to the scales the grid resolution can allow. This issue has been present through all the research, especially in Chapter VI, where the effect of artificial dissipation was studied and calibrated using decaying homogeneous turbulence in a periodic domain, and in Chapter VII, where the flow around a sphere was studied for different levels of artificial dissipation.

In this investigation, the artificial dissipation was studied to assess its optimum level that, without risking the stability of the scheme, would not smooth out the instabilities that DES was expected to capture. However, a different approach was presented by Strelets et al. [40]. In this work, it was shown how an excessive level of dissipation fails to take full advantage of the grid resolution by destroying the energy cascade before the Sub-Grid-Scales (SGS) eddy viscosity can dissipate the small scales. Although the discussion in this investigation centered on upwind schemes, the effect is the same in the case of centered schemes with added artificial dissipation, as is the case for the code used in this research. The solution presented by Strelets et al. [40] is to use a hybrid central/upwind approximation of the inviscid fluxes in the governing equations. The scheme has to adjust from a central scheme to a more dissipative upwind scheme in the irrotational regions to guarantee the stability in the coarse grids usually used in such

regions. The baseline is to have a blending function, that generates this self-adaptive scheme which evolves from an upwind scheme in the irrotational regions to a centered scheme in the rotational regions, where the eddy viscosity can ensure the stability of the solution.

In this research, a similar approach was investigated and is presented here as a future line of investigation to be explored. The code used in this research uses a second order central approximation for the inviscid and viscous fluxes with added artificial dissipation. The goal is to tune the artificial dissipation depending on the mesh density and the flow characteristics (vorticity levels) and this was attempted based on two main ideas:

- A reference level of total dissipation is established and the artificial dissipation is adjusted at each iteration to ensure that this reference level is attained but not exceeded. After analyzing the results of Chapter VI, it was observed that the stability of the scheme can be ensured using only the artificial dissipation. It can be observed in Figures 21 and 22 in Chapter VI, that using the nominal value for the artificial dissipation scaling factor, the total dissipation is mainly artificial and is enough to ensure stability, and in fact is excessively dissipative in terms of flow resolution. Therefore, it is quite easy to envision a simple logic that forces the sum of the physical and artificial dissipation to be equal to the levels of dissipation obtained by the artificial viscosity on its own for a determined value

of the scaling factor. The mathematical expression of the adaptive logic explained above is the following:

$$Total\ dissipation = \left\{ \begin{matrix} Artificial \\ Dissipation \end{matrix} \right\}_{\alpha} + \left\{ \begin{matrix} Physical \\ Dissipation \end{matrix} \right\} = \left\{ \begin{matrix} Artificial \\ Dissipation \end{matrix} \right\}_{\substack{reference \\ value\ of\ the \\ scaling\ factor}}$$

where α is the calculated artificial dissipation scaling factor required to ensure that the total dissipation is equal to the artificial dissipation obtained with a reference value of the scaling factor.

As the dissipation is computed along edges, the adaptive logic is applied at each iteration for each node along all the edges intersecting the node. Moreover, the dissipation is computed for four different equations corresponding to the conservation of momentum and energy. Therefore, for each node there will be several computed values for the artificial dissipation scaling factor to balance the dissipation along each edge containing that node. The final artificial dissipation scaling factor value that will be stored as the optimized adaptive value of each node will be the average of the values obtained for each equation (1 continuity +3 momentum + 1 energy) and for each edge containing that node.

- Second, the reference level can be adjusted in each time iteration as a function of the averaged cell Reynolds number based on the eddy viscosity. The averaged cell Reynolds number will be computed averaging the cell Reynolds number over

all the cells in the mesh. This Reynolds number will be used to determine the reference value of the scaling factor in the right hand side of the previous equation.

Results for the decaying homogeneous turbulence in a box using adaptive artificial dissipation

Preliminary results were obtained for cases of decaying homogeneous turbulence in a periodic domain in which the artificial dissipation was adjusted to force the total dissipation, artificial plus physical, to be equal to the artificial dissipation produced for a scaling factor value of 3.5 and 4.0 (Figure 71).

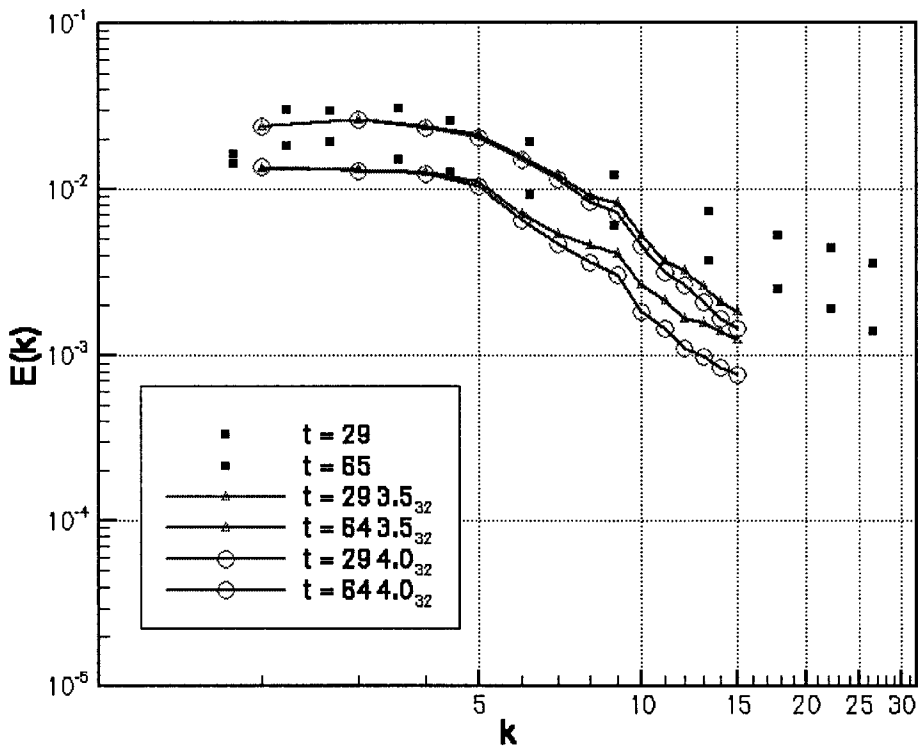

Figure 69. Preliminary results for the case of decaying homogenous turbulence in a box for adaptive artificial dissipation. Computation performed in an unstructured mesh of (32x32x32) nodes.

The results obtained did not significantly improve the results obtained for non-adaptive artificial dissipation with a scaling factor of 5.0, as can be observed comparing Figure 71 and Figure 24. This indicates that this constant value of the scaling factor of the artificial dissipation is very well optimized and better results could not be obtained even with adaptive schemes for the artificial dissipation. However, the adaptive artificial dissipation logic was initially envisioned for cases in which very different flows were present simultaneously (highly turbulent regions and irrotational regions). It is in those cases where an adaptive artificial dissipation can be very useful since it will turn itself on and off automatically to ensure stability in irrotational regions or let DES capture the small eddies depending on the flow characteristics. Obviously, the decaying homogeneous turbulence in a periodic domain is not the best case for this purpose since there are no different flow characteristics present in the domain. It would be interesting to test this approach for cases such as the flow around a sphere or the stall regime of a wing at 60 degrees angle of attack.

Some other tests were performed by trying to adapt the artificial dissipation using the averaged Reynolds number based on the eddy viscosity as explained above. No improvements with respect to non-adapted cases were obtained and further studies need to be conducted along that line of inquiry. Again, note that all these cases were performed using the decaying homogeneous turbulence in a periodic domain case, which is not a well-suited case for these purposes. This test was the preferred choice for speed and simplicity.

In summary, adaptive artificial dissipation seems a reasonable step to improve DES performance that should not include excessive complexity in the code. Previous work on this area has reported interesting results [40] and further work is recommended.

Mesh refinement and adaptive meshing

As stated in previous chapters, an adequate mesh, fine enough to capture the different scales present in the flow, is necessary to allow DES to perform correctly. If DES is not applied on an adequate mesh, it will not be able to resolve a detailed flow solution and a RANS-like solution will be obtained instead.

For the case of the flow over a NACA 0012 airfoil, initial post-stall runs conducted at 60 degrees angle of attack were performed using the same mesh that had been used for the near-stall tests. Preliminary results showed that the computed values of C_L and C_D were not as accurate as expected based on previous results presented by Shur et al [10]. It was concluded that the mesh was not fine enough in the region of interest; that is the region on the upper side of the wing where DES is expected to capture most of the vorticity associated with the massively separated flow (Figure 59). Consequently, DES was not able to display its full capability and no significant improvements were obtained as compared to RANS calculations.

Based on this preliminary result, the original mesh was modified by adding nodes in the region of interest to enable DES with a fine mesh to capture the small eddies

present in the flow. A code was programmed which inserted nodes in all tetrahedra that had all four nodes inside the pre-defined region. Obviously, the region chosen to inject the nodes was the volume on the upper side of the wing where the wake was expected to evolve at 60 degrees angle of attack. The node insertion was limited to tetrahedra inside the domain, avoiding the insertion of nodes close to the boundaries (wing surface and walls). This simplified the process since no modifications were included in the boundary conditions or the surface mesh on the wing surface.

The final result was a mesh of 2,107,026 nodes with at least 875,359 nodes (number of nodes inserted) in the region of interest. A cut of the new mesh at z=1 is shown and compared to the original mesh in Figures 72 and 73.

Problems related to the size of the mesh files and the capabilities of the machines available appeared in the pre-processing of the mesh and it was impossible to correctly produce coarser multigrid levels. Not having multigrid to accelerate convergence will dramatically impact the convergence rates and will make an unsteady calculation unfeasible due to time limitations. However, a steady calculation (unsteady with very big time steps) was used to determine an approximation of the averaged value of the result and to determine any improvements in the result as compared to URANS and previous DES runs.

A steady run of 10,000 sub-iterations was produced as an alternative to obtain an averaged value of C_L and C_D. The evolution of the aerodynamic coefficients through the

10,000 iterations is shown in Figure 74 and Figure 75 illustrates the sequence of the residual. Note, that this is not a time history sequence of the lift and drag coefficient but an evolution through the sub-iterations of a steady run, which is not time accurate. Nevertheless, it provides an estimate of the averaged value of the C_L and C_D that, without being conclusive, shows promising results of what DES is capable of when provided with an adequate mesh.

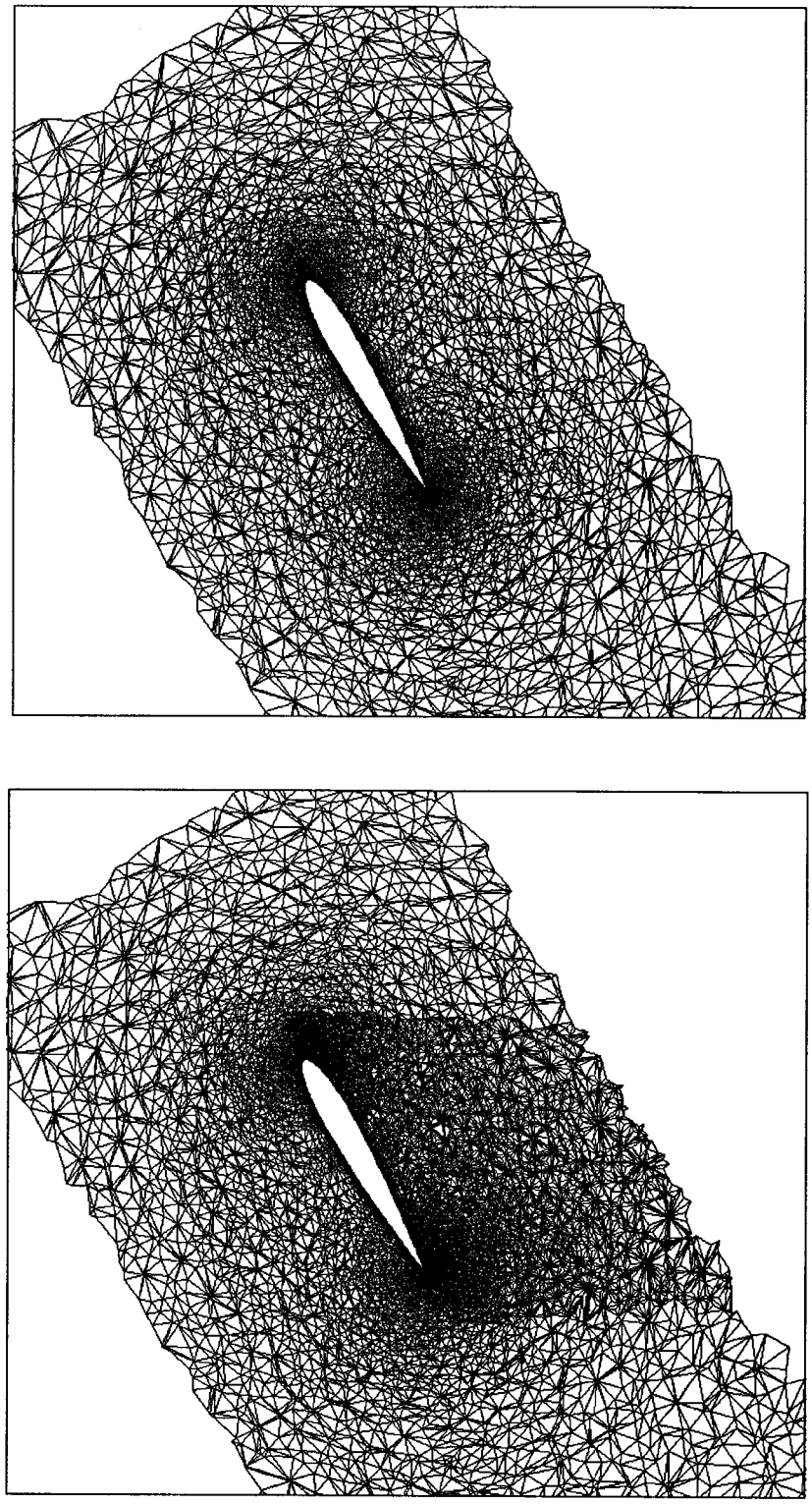

Figure 70. Near-field view of original meshes and improved mesh two-dimensional cut at z=0.

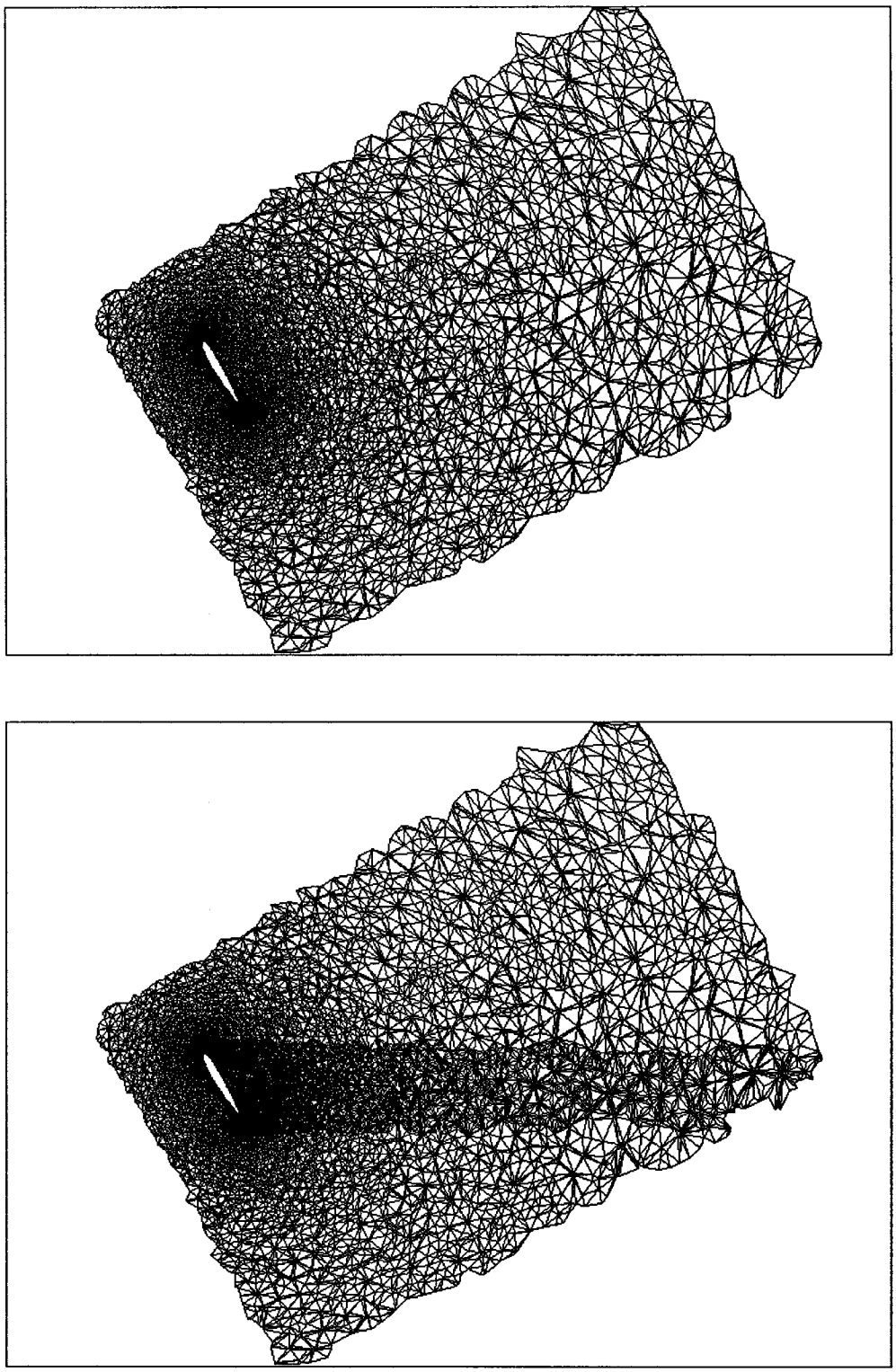

Figure 71. Far-field view of original meshes and improved mesh two-dimensional cut at z=0.

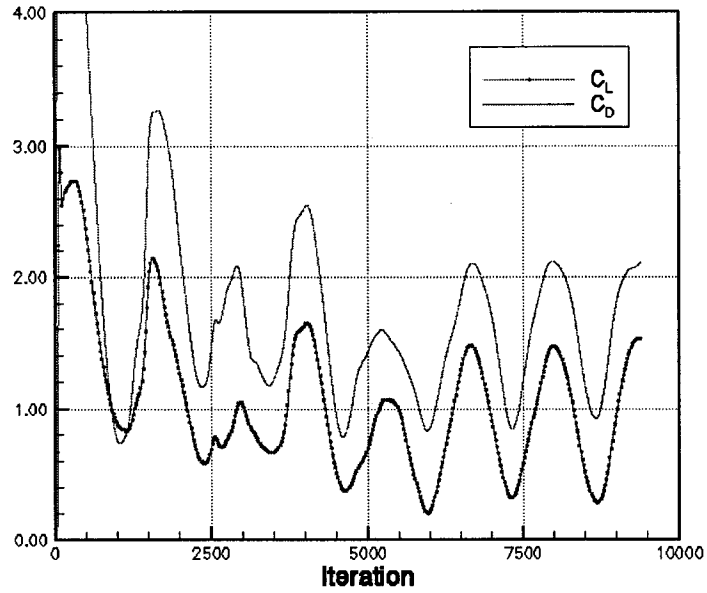

Figure 72. Iteration sequence of the lift and drag coefficient for a steady DES calculation of a NACA 0012 at 60 degrees AoA. Computation performed on unstructured mesh of 2,107,026 nodes. Mach number = 0.25, Re = 10^5.

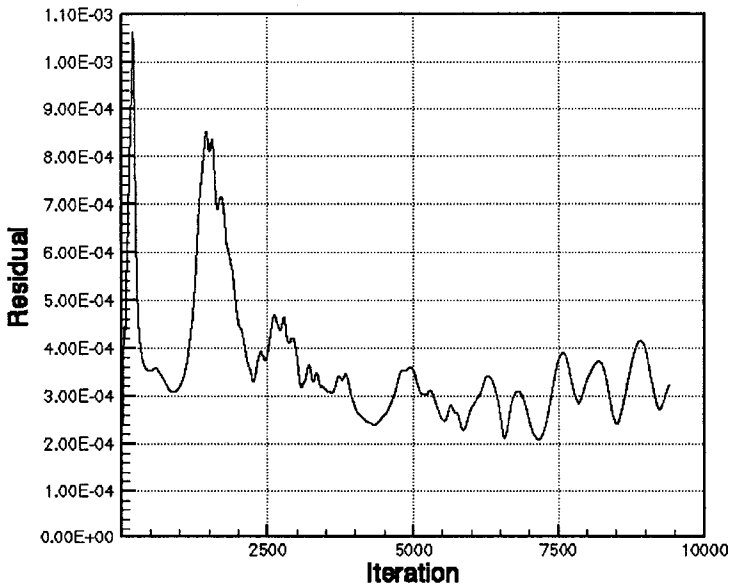

Figure 73. Iteration sequence of the residual for a steady DES calculation for a NACA 0012 at 60 degrees AoA. Computation performed on unstructured mesh of 2,107,026 nodes. Mach number = 0.25, Re = 10^5.

The averaged C_L and C_D computed are summarized and compared to the DES results initially obtained using the coarser initial mesh, the DES results obtained by Shur et al. [10], the URANS results and the experimental results, in Table 13. The steady DES run is averaged between iterations 6000 and 9400 (Figure 74).

	C_L	C_D
Experimental	0.90	1.625
Shur et al. (DES)	1.000	1.625
URANS	1.300	2.250
DES (original mesh)	1.520	2.540
DES (improved mesh) (Steady run)	0.920	1.590

Table 13. Comparison of computed lift and drag coefficients obtained for AoA=60 degrees, Mach number = 0.25 and Re = 10^5 with experimental values and previous computational results.

The results obtained with the refined mesh agree more closely to the experimental values and to the results computed by Shur et al [10]. Note that these results are not time accurate and cannot conclusively be compared to the other unsteady results. However, this case emphasizes how important the mesh quality is for DES to be able to capture the flow features, and raises the subject of self–adaptive meshing that would inject nodes in the regions of interest based on some flow characteristic such as vorticity.

Higher order methods for spatial and time discretization

Higher order methods for the spatial and time discretizations would improve the accuracy of the scheme. Furthermore, higher order methods of the spatial discretization would reduce the effects of the artificial dissipation and would improve the solutions obtained by DES as it has been proved by previous investigations [11, 12].

Work has been done by G. Jothiprasad, D. J. Mavriplis and D. A. Caughey to extend the solver used in this investigation to higher order methods [41]. This work has shown how the number of required time steps can be reduced and temporal accuracy can be increased through the use of high order accurate implicit Runge-Kutta schemes.

Hybrid RANS/LES methods using different RANS turbulence models

DES is the first technique that combines RANS and LES in a hybrid approach to combine their strengths in the flow regimes where they are more capable. However, hybrid RANS/LES methods do not necessarily have to be limited to the one equation Spalart-Allmaras turbulence model. Other turbulence models can be tested and used combined with LES to explore its advantages and disadvantages versus the Spalart-Allmaras model. Similarly other LES Sub-Grid-Scale models should also be tested.

In this line of research some work has been done by N. J. Georgiadis, J. I. D. Alexander and E. Reshotko [42].

CHAPTER X

CONCLUSIONS

The Detached Eddy Simulation (DES) technique was successfully implemented in a second-order accurate unstructured mesh steady-state solver. Initial efforts focused on making the steady solver capable of time accurate calculations. An implicit second-order accurate scheme was employed and the non-linear equations at each time-step were solved using a steady-state unstructured agglomeration multigrid solver. The unsteady solver was satisfactorily tested for the flow over a circular cylinder. The correct vortex shedding frequency was computed as compared to experimental results and the second-order accurate convergence behavior was observed as the time-step was reduced. Moreover, the pressure distribution as well as the skin friction distribution were accurately computed.

DES was implemented and the effect of the artificial dissipation assessed using the test case of decaying homogenous turbulence in a periodic domain. The objective was to investigate the optimum level of artificial dissipation required to ensure the stability of a second-order accurate central difference scheme, and at the same time, to minimize damping of most of the physical instabilities present in the flow that DES is expected to capture. Tests were conducted using two different meshes, different C_{DES} values (C_{DES} represents a model constant) and different cell types (prism and hexahedral). It was concluded that the most consistent results were obtained using one fourth of the nominal value (the value generally employed for steady calculations in

RANS mode) of the artificial dissipation scaling factor and a C_{DES} value of 0.65. This C_{DES} value confirmed the results obtained in previous investigations.

Using the results obtained for the isotropic decaying turbulence in a periodic domain, the flow around a sphere was computed at a Reynolds number of 10^4 and a Mach number of 0.2. The solution obtained using DES exhibited frequency contents in the drag coefficient time history in reasonable agreement with experimental measurements. The RANS solution did not capture any of these instabilities. It was necessary to force transition at 90 degrees azimuthal to obtain the correct pressure and skin friction distribution over the surface of the sphere.

Finally, a test case of aeronautical interest such as the flow over a NACA 0012 wing was used to test the DES approach. Computations of the lift and drag coefficient near the stall angle of attack (12 degrees incidence) did not show significant differences between the computed RANS and DES results. A new test was designed at 60 degrees angle of attack to investigate DES in a massive separated regime, beyond the scope of application of RANS. The results obtained were surprisingly poor, very close to the results obtained by previous URANS calculations and over-predicting experimental values by 50%. Since the mesh used to compute these results was suspected to be too coarse, a refined mesh was generated by injecting 800,000 nodes in the region of interest. Problems with the mesh and time constraints did not allow for a time-accurate solution but averaged steady runs delivered promising results for the lift and drag coefficient, very close to DES computed results obtained in previous investigations.

DES has been implemented, tested and concluded to be a quite simple to implement turbulence modeling technique, that is practical based on the current computational resources and is capable of resolving massively separated flows to scales beyond URANS capabilities. It expands CFD to flows out of the feasible domain of URANS that could only be explored using LES or DNS. However, considering that it will require several decades before LES and DNS mature enough to be suitable techniques for engineering problems, DES presents itself as a good solution keeping a good balance between the obtained results and the cost to obtain them, in time and in computational resources. In conclusion, DES appears highly promising and opens up interesting lines of research for future investigations dealing with massively separated flows, which in the past would have been limited by the use of very expensive techniques such as LES or DNS.

REFERENCES

1. D. J. Mavriplis, D. W. Levy, "Transonic drag prediction using an unstructured multigrid solver", AIAA paper, 2002-0838, 2002.
2. D. W. Levy, T. Zickur, J. Vassberg, S. Agrawal, R. A. Wahls, S. Pirzadeh, M. J. Hemsch, "Summary of data from the first AIAA CFD drag prediction workshop", AIAA paper, 2002-0841, January 2002.
3. E. N. Tinoco, "An Assessment of the CFD Prediction of Drag and Other Longitudinal Characteristics", AIAA paper, 2001-1002, 2001.
4. J. V. Sanchez, "An Optimally Converging Multigrid Algorithm for the Three-Dimensional Euler Equations", Master Thesis, George Washington University, 1998.
5. V. Venkatakrishnan, "A perspective on unstructured grid flow solvers", NASA CR 195025, ICASE Report No. 95-3, 1995.
6. Edited by R. Rubinstein, C. L. Rumsey, M. D. Salas, J. L. Thomas, "Turbulence Modelling Workshop", ICASE Interim Report No. 37, NASA CR-2001-210841, 2001.
7. R. D. Joslin, "Discussion of DNS: Past, Present and Future", NASA Technical Report, August 4-8, 1997.
8. P. R. Spalart, "Strategies for turbulence modelling and simulations", Fourth International Symposium on Engineering Turbulence modelling and Measurements, May 24-26, 1999, Corsica.

9. 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" Advances in DNS/LES, 1st AFOSR Int. Conf. On DNS/LES, Greyden Press, Columbus Oh, Aug 4-8, 1997

10. M. Shur, P. R. Spalart, M. Strelets and A. Travin, "Detached-eddy simulation of an airfoil at high angle of attack", 1999

11. A. Travin, M. Shur, M. Strelets and P. Spalart, "Detached-Eddy Simulation past a Circular Cylinder", Journal of Flow, Turbulence and Combustion, 63, 269-291, 1999.

12. G. S. Constantinescu, K. D. Squires, "LES and DES Investigations of Turbulent Flow over a Sphere", AIAA paper 2000-0540, 2000.

13. J. R. Forsythe, "Numerical Computation of Turbulent Separated Supersonic Flowfields", Doctoral Dissertation, Wichita State University, 2000.

14. J. R. Forsythe, K. D. Squires, K. E. Wurtzler, P. R. Spalart, "Detached-Eddy Simulation of Fighter Aircraft at High Alpha", AIAA paper, 2002-0591, January 2002.

15. K. D. Squires, J. R. Forsythe, S. A. Morton, D. C. Blake, M. Serrano, K. E. Wurtzler, W. Z. Strang, R. F. Tomaro, P. R. Spalart, "Analysis of Full Aircraft with Massive Separation using Detached-Eddy Simulation".

16. K. D. Squires, J. R. Forsythe, S. A. Morton, W. Z. Strang, K. E. Wurtzler, R. F. Tomaro, M. J. Grismer, P. R. Spalart, "Progress on Detached-Eddy Simulation of Massively Separated Flows", AIAA paper, 2002-1021, January 2002.

17. S. A. Morton, J. R. Forsythe, K. D, A. Mitchell, D. Hajek, "DES and RANS

Simulation of Delta Wing Vortical Flow", AIAA paper, 2002-0587, 2002.

18. S. A. Morton, J. R. Forsythe, K. D. Squires, K. E. Wurtzler, "Assessment of Unstructured Grids for Detached-Eddy Simulation of High Reynolds Number Separated Flows". http://www.eas.asv.edu/squires/publications/grid.pdf
19. J. O. Hinze, *Turbulence,* Second Edition, Mc Draw-Hill, Inc., 1987.
20. Schlicting, H., ``Boundary Layer Theory'', Seventh Edition, McGraw Hill, New York, NY, 1979.
21. Venkatakrishnan, V., Mavriplis, D. J., ``Computation of Unsteady Flows over Complex Geometries in Relative Motion'', 1st AFOSR Conference On Dynamic Motion CFD, Rutgers University, New Brunswick, NJ, June 3-5, 1996.
22. Venkatakrishnan, V., Mavriplis, D. J., ``Implicit Method for the Computation of Unsteady Flows on Unstructured Grids'', Journal of Computational Physics, March 1996.
23. P. R. Spalart, Young-Person's Guide to Detached-Eddy Simulation Grids, NASA/CR-2001-211032, July 2001, pp. 23
24. D. C. Wilcox, *Turbulence Modelling for CFD*, Second Edition, DCW Industries Inc., 1998.
25. D. J. Mavriplis, "Aerodynamics Drag Prediction using Unstructured Mesh Solvers".
26. C. Hirsch, *Numerical Computation of Internal and External Flows*, Volume 1&2, John Wiley & Sons, 1995.
27. B. Hendrickson and R. Leland, "The Chaco User's Guide: Version 2.0" Tech. Rep. SAND94-2692, Sandia National Laboratories, Albuquerque, NM, July

1995.

28. D. J. Mavriplis, "Multigrid Strategies for Viscous Flow Solvers on Anisotropic Unstructured Meshes", NASA CR-1998-206910, ICASE Report No. 98-6, 1998.
29. D. J. Mavriplis, "On convergence acceleration techniques for unstructured meshes", NASA CR-208732, ICASE Report No. 98-44, 1998.
30. D. J. Mavriplis, A. Jameson, L. Martinelli, "Multigrid Solution of the Navier-Stokes Equations on Triangular Meshes", NACA/CR 181786, ICASE Report No. 89-11, 1989.
31. C. Norberg, "Flow Around a Circular Cylinder: Aspects of Fluctuating Lift", Journal of Fluids and Structures, 15, 459-469, 2001.
32. M .M. Zdravkovich, *Flow Around Circular Cylinders, Vol 1: Fundamentals*, Oxford University Press, July 1997.
33. C. Wieselsberger, "New Data of the Laws of Fluid Resistance", NACA Techincal Notes, March 1922.
34. G. Comte-Bellot, S. Corrsin, "Simple Eulerian correlation of fill-and narrow-band velocity signals in grid-generated, 'isotropic' turbulence", J. Fluid Mechanics, vol. 48, part 2, pp. 273-337, 1971.
35. L. F. G., Simmons and C. Salter, "Experimental investigation and analysis of the velocity variations in turbulent flow", Proc. Roy. Soc., 145, 212-234, 1934.
36. S. W. Churchill, *Viscous Flows: The Practical Use of Theory,* Butterworth, Boston, 1988.
37. (http://geolab.larc.nasa.gov/GridTool)
38. D. J. Mavriplis, S. Pirzadeh, "Large-Scale Parallel Unstructured Mesh

Computations for 3D High-Lift Analysis", AIAA 99-0537, January 1999.

39. E. Jacobs, A. Sherman, "Airfoil section characteristics as affected by variations of the Reynolds number", NACA Report No. 586, 1937.
40. M. Strelets, "Detached Eddy Simulation of Massively Separated Flows", Fourth International Symposium on Engineering Turbulence Modelling and Measurements, May 24-26, 1999, Corsica.
41. G. Jothiprasad, D. J. Mavriplis and D. A. Caughey, "Higher Order Methods for the Unsteady Navier-Stokes Equations on Unstructured Meshes".
42. N. J. Georgiadis, J. I. D. Alexander and E. Reshotko, "Development of a Hybrid RANS/LES Method for Compressible Mixing Layer Simulations".

VITA

Juan Antonio Pelaez was born on March 3, 1974 in Madrid, where he received his secondary education. He entered Saint Louis University Madrid Campus in 1992, transferring to Parks College of Saint Louis University in 1994, where he received a Bachelor of Science in Aerospace Engineering in May 1996. He worked as a Graduate Teaching Assistant at Parks College of Saint Louis University from August 1996 to May 1998 when he graduated from a Master of Science in Aerospace Engineering. In August 1998 he started working as Research Assistant at Old Dominion University and by 1999 he started to conduct his research at ICASE (Institute for Computer Applications in Science and Engineering) located at NASA Langley Research Center. In March of 2001 he accepted a position at INDRA and moved back to Madrid, Spain, where he works as an engineer in flight simulation projects. He conducted the last two years of his dissertation research part time and he received his Doctoral degree in Aerospace Engineering in May of 2003.