

# Verification and Validation of HiFiLES: a High-Order Navier-Stokes unstructured solver on multi-GPU platforms

Manuel R. López-Morales <sup>\*</sup>, Abhishek Sheshadri,<sup>†</sup>,  
Kartikey Asthana<sup>†</sup>, Jonathan Bull<sup>‡</sup>, Jacob Crabbill<sup>§</sup>, Thomas Economou<sup>†</sup>, David Manosalvas<sup>†</sup>,  
Joshua Romero<sup>†</sup>, Jerry Watkins<sup>†</sup>, David Williams<sup>¶</sup>,  
Francisco Palacios<sup>||</sup>, and Antony Jameson<sup>\*\*</sup>

*Department of Aeronautics and Astronautics, Stanford University, Stanford, CA, 94305*

The goal of this paper is to show a detailed Verification and Validation of HiFiLES: a high-order Navier-Stokes solver developed at the Aerospace Computing Laboratory (ACL) at Stanford University. HiFiLES has been built on top of SD++ (Castonguay et al.) and achieves high-order spatial discretization with the Energy-Stable Flux Reconstruction (ESFR) scheme in unstructured grids (two and three dimensions). The high parallelizability of this scheme motivates the optimization of the solver's ability to run in a multi-GPU (Graphical Processing Unit) environment. We intend this paper to be the main reference for HiFiLES and serve as a sort of manual for researchers and engineers that would like to develop or implement high-order numerical schemes based on an Energy-Stable Flux Reconstruction (ESFR) scheme.

## I. Introduction

Over the last 20 years, much fundamental work has been done in developing high-order numerical methods for Computational Fluid Dynamics. Moreover, the need to improve and simplify these methods has attracted the interest of the applied mathematics and the engineering communities. Now these methods are beginning to prove themselves sufficiently robust, accurate, and efficient for use in real-world applications.

However, low-order numerical methods are still the standard in the aeronautical industry. There has been a sustained scientific and economical investment to develop this successful and robust technology for a long time. Currently, a state-of-the-art 2nd-order Finite Volume computational tool performs adequately well in a broad range of aeronautical engineering applications. For that reason, the introduction of brand-new high-order numerical schemes in the aeronautical industry is challenging, particularly in areas where the low-order numerical methods already provide the required robustness and accuracy (keeping in mind the limitations of the current turbulence model technology).

Thanks to new and emerging aircraft roles (very small or very big concepts, very high or very low altitude, quiet vehicles, low fuel consumption, etc.), revolutionary aircraft design concepts will appear in the near future and the need for high-fidelity simulation techniques to predict their performance is growing rapidly. Undoubtedly, high-order numerical methods are starting to find their place in the aeronautical industry.

Unsteady simulation, flapping wings, wake capturing, noise prediction, and Large Eddy Simulations are just a few examples of computations that require high-order numerical methods. In particular, high-order methods have a significant edge in applications that require accurate resolution of the smallest scales of the flow. Such situations include the generation and propagation of acoustic noise from the airframe, or at the limits of the flight envelope where unsteady vortex-dominated flows have a significant effect on the aircraft's performance. Utilizing a high-order representation enables the smallest scales to be represented with a greater degree of accuracy than standard second-order methods. Furthermore, high-order methods are inherently less dissipative, resulting in less unwanted interference

<sup>\*</sup>Ph.D. Candidate, Department of Aeronautics and Astronautics, Stanford University, AIAA Student Member; mlopez14@stanford.edu

<sup>†</sup>Ph.D. Candidates (authors in alphabetical order), Department of Aeronautics and Astronautics, AIAA Student Members

<sup>‡</sup>Postdoctoral Scholar, Department of Aeronautics and Astronautics, Stanford University, AIAA Member

<sup>§</sup>M.S. Student, Department of Aeronautics and Astronautics, Stanford University, AIAA Student Member

<sup>¶</sup>CFD Engineer, Flight Sciences Division, Boeing Commercial Airplanes, AIAA Member

<sup>||</sup>Engineering Research Associate, Department of Aeronautics and Astronautics, AIAA Senior Member

<sup>\*\*</sup>Thomas V. Jones Professor of Engineering, Department of Aeronautics and Astronautics, Stanford University, AIAA Fellow

with the correct development of the turbulent energy cascade.

Finally, the amount of computing effort to achieve a small error tolerance can be much smaller with high-order than second-order methods. Even real time simulations (one second of computational time, one second of real flight), could benefit from high-order algorithms which count on a more intensive inner element computation (ideal for vector machines and new computational platforms like GPUs, FPGAs, coprocessors, etc).

But, before claiming the future success of the high-order numerical methods in the industry, two main difficulties should be overcome: a) high-order numerical schemes must be as robust as state-of-the-art low-order numerical methods, b) the existing level of Verification and Validation (V&V) in high-order CFD codes should be similar to the typical level of their low-order counterparts.

During the last decade, the Aerospace Computing Laboratory (ACL) of the Department of Aeronautics and Astronautics at Stanford University has developed a series of high-order numerical schemes and computational tools, contributing massively to the demonstration of the viability of this technique. In this paper, a code called HiFiLES developed at ACL and built on top of SD++ (Castonguay et al.<sup>?</sup>) is described in detail with a particular emphasis on robustness in a broad range of applications and an industrial-like level of V&V. HiFiLES takes advantage of the synergies between applied mathematics, aerospace engineering, and computer science to achieve the ultimate goal of developing an advanced high-fidelity simulation environment.

Apart from the original characteristics of the SD++ code (described in Castonguay et al.<sup>?</sup>), HiFiLES includes some important physical models and computational methods such as: Large Eddy Simulation (LES) using explicit filters and advanced subgrid-scale (SGS) models, high-order stabilization techniques, shock detection and capturing for compressible flow calculations, convergence acceleration methodologies like p-multigrid, and local and dual time stepping.

During the development of this software several key decisions have been taken to guarantee a flexible and lasting infrastructure for industrial Computational Fluid Dynamics simulations:

- The selection of the Energy-Stable Flux Reconstruction (ESFR) scheme on unstructured grids. The flexibility of this method has been critical to guarantee a correct solution independently of the particular physical characteristics of the problem.
- High performance, materialized in a multi-GPU implementation which takes advantage of the ease of parallelization afforded by discontinuous solution representation. Furthermore, HiFiLES aims to guarantee compatibility with future vector machines and revolutionary hardware technologies.
- Code portability by using ANSI C++ and relying on widely-available, and well-supported mathematical libraries like Blas, LAPACK, CuBLAS and ParMetis.
- Object oriented structure to boost the re-usability and encapsulation of the code. This abstraction enables modifications without incorrectly affecting other portions of the code. Although some level of performance is traded for re-usability and encapsulation, the loss in performance is minor.

As mentioned before, the mathematical basis and computational implementation of HiFiLES were described in Castonguay et al.<sup>?</sup>. For that reason, the goal of this paper is to illustrate the level of robustness of HiFiLES in complex problems, via a detailed Verification and Validation study which is fundamental to increase the credibility of this technology in a competitive industrial framework.

In particular, to ensure that the implementation of the aforementioned features in HiFiLES is correct, the following verification tests are shown: checks of spatial and temporal order of accuracy using the Method of Manufactured Solutions (MMS) in 2D and 3D for viscous and inviscid flows, characterization of stable time-step limits, assessment of computational cost per degree of freedom for a given error tolerance, and measurement of weak and strong scalability in GPUs and CPUs.

After the Verification, a detailed Validation of the code is presented to illustrate that the solutions provided by HiFiLES are an accurate representation of the real world. Simulations of complex flows are validated against experimental or Direct Numerical Simulation (DNS) results for the following cases: laminar and turbulent flat plane, flow around a circular cylinder, subsonic flow attached in a NACA0012, SD7003 wing-section and airfoil at 4° angle of attack, and LES of the Taylor-Green Vortex.

The organization of this paper is as follows. Section II. provides a description of the governing equations. Section III. describes the mathematical numerical algorithms implemented in the code (with a particular emphasis in convergence acceleration techniques and stability). IV. and V. focus on the Verification and Validation of HiFiLES, and finally, the conclusions are summarized in section VI..

Finally, it is our intent for this paper to be the main reference for work that uses or enhances the capabilities of HiFiLES, and for it to serve as a sort of reference for researchers and engineers that would like to develop or implement High-order numerical schemes based on an Energy-Stable Flux Reconstruction (ESFR) scheme.

## II. Governing Equations

### A. Navier Stokes (NS) Equations

The main purpose of HiFiLES is to perform High-Fidelity Large-Eddy Simulations –hence the name. The form of the NS Equations most useful for understanding how the FR methodology can be used to solve them is the following

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F} = 0 \quad (1)$$

where  $\mathbf{F} = (F, G, H) = (F_I, G_I, H_I) - (F_V, G_V, H_V)$  and

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho e \end{pmatrix} \quad F_I = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ \rho ue + p \end{pmatrix} \quad G_I = \begin{pmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vw \\ \rho ve + p \end{pmatrix} \quad H_I = \begin{pmatrix} \rho w \\ \rho wu \\ \rho wv \\ \rho w^2 + p \\ \rho we + p \end{pmatrix} \quad (2)$$

$$F_V = \begin{pmatrix} 0 \\ \sigma_{xx} \\ \sigma_{xy} \\ \sigma_{xz} \\ u_i \sigma_{ix} - q_x \end{pmatrix} \quad G_V = \begin{pmatrix} 0 \\ \sigma_{yx} \\ \sigma_{yy} \\ \sigma_{yz} \\ u_i \sigma_{iy} - q_y \end{pmatrix} \quad F_V = \begin{pmatrix} 0 \\ \sigma_{zx} \\ \sigma_{zy} \\ \sigma_{zz} \\ u_i \sigma_{iz} - q_z \end{pmatrix} \quad (3)$$

As usual,  $\rho$  is density,  $u, v, w$  are the velocity components in the  $x, y, z$  directions, respectively, and  $e$  is total energy per unit mass. In HiFiLES, the pressure is determined from the ideal gas equation of state

$$p = (\gamma - 1)\rho \left( e - \frac{1}{2} (u^2 + v^2 + w^2) \right) \quad (4)$$

the viscous stresses are those of a Newtonian fluid

$$\sigma_{ij} = \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \mu \delta_{ij} \frac{\partial u_k}{\partial x_k} \quad (5)$$

and the heat fluxes are defined as

$$q_i = -k \frac{\partial T}{\partial x_i} \quad (6)$$

where

$$k = \frac{C_p \mu}{\text{Pr}}, \quad T = \frac{p}{R \rho} \quad (7)$$

$\text{Pr}$  is the Prandtl number,  $C_p$  is the specific heat at constant pressure and  $R$  is the gas constant. In the case of air,  $\gamma = 1.4$  and  $\text{Pr} = 0.72$ . The dynamic viscosity  $\mu$  in HiFiLES can be a constant or a function of temperature using Sutherland's law.

### B. Reynolds Averaged Navier-Stokes (RANS) equations

The compressible NS equations can be used to solve a variety of different flow physics problems but for turbulent flows, direct numerical simulation using these equations can become excessively expensive. For engineering applications, it is customary to perform a Favre averaging procedure to the NS equations to solve a turbulent mean quantity. This leads to a variety of terms which must be modeled in order to provide closure to the resulting RANS equations<sup>??</sup>. The conservative form of the equations is very similar to the NS equations with an added source term:

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F} = \mathbf{Q} \quad (8)$$

The conservative variables are now averaged quantities and the RANS equations must now be solved with an appropriate turbulence model.

### III. Numerics

#### A. Flux Reconstruction Method

What follows is an overview of the flux reconstruction (FR) framework. We start the discussion with the solution of the advection-diffusion equation in one dimension using the FR approach to illustrate the peculiarities of the method. We then proceed to briefly explain how conservation equations can be solved in multiple dimensions. The NS equations are a set of coupled conservation equations in multiple dimensions, so the extension of the FR methodology to them is straightforward. The detailed description of the algorithm used in HiFiLES is given by Castonguay et al.<sup>7</sup>.

##### 1. Solution of the Advection Equation in One Dimension using the FR Approach

Consider the one-dimensional conservation law

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = 0 \quad (9)$$

in domain  $\Omega$ , where  $x$  is the spatial coordinate,  $t$  is time,  $u$  –the *solution*– is a scalar function of  $x$  and  $t$ , and  $f$  –the *flux*– is a scalar function of  $u$ . Note that by letting  $f = f(u, \frac{\partial u}{\partial x})$ , Equation 9 becomes a model of the Navier-Stokes equations.

Let us partition the domain  $\Omega = [x_1, x_{N+1}]$  into  $N$  non-overlapping elements with interfaces at  $x_1 < x_2 < \dots < x_{N+1}$ . Then,

$$\Omega = \bigcup_{n=1}^N \Omega_n \quad (10)$$

and  $\Omega_n = [x_n, x_{n+1}]$  for  $n = 1, \dots, N$ .

To simplify the implementation, let us map each of the physical elements  $\Omega_n$  to a standard element  $\Omega_s = [-1, 1]$  with the function  $\Theta_n(\xi)$ , where

$$x = \Theta_n(\xi) = \left( \frac{1 - \xi}{2} \right) x_n + \left( \frac{1 + \xi}{2} \right) x_{n+1} \quad (11)$$

With this mapping, the evolution of  $u$  within each  $\Omega_n$  can be determined with the following transformed conservation equation

$$\frac{\partial \hat{u}}{\partial t} + \frac{1}{J_n} \frac{\partial \hat{f}}{\partial \xi} = 0 \quad (12)$$

where

$$\hat{u} = u(\Theta_n(\xi), t) \text{ in } \Omega_n \quad (13)$$

$$\hat{f} = f(\Theta_n(\xi), t) \text{ in } \Omega_n \quad (14)$$

$$J_n = \left. \frac{\partial x}{\partial \xi} \right|_{\Omega_n} \quad (15)$$

Now, introduce polynomials of degree  $p$ ,  $\hat{u}^\delta$  and  $\hat{f}^\delta$ , to approximate the exact values  $\hat{u}$ ,  $\hat{f}$ , respectively. We can write these polynomials as

$$\hat{u}^\delta = \sum_{i=1}^{N_s} \hat{u}_i^\delta l_i(\xi) \quad (16)$$

$$\hat{f}^\delta = \sum_{i=1}^{N_s} \hat{f}_i^\delta l_i(\xi) \quad (17)$$

where  $N_s$  is the number of solution points,  $\hat{u}_i^\delta$  is the current value of the solution approximation function at the  $i^{\text{th}}$  *solution point* in the reference element,  $\hat{f}_i^\delta$  is the current value of the flux approximation function at the  $i^{\text{th}}$  *flux point* in the reference element,  $l_i$  is the Lagrange polynomial equal to 1 at the  $i^{\text{th}}$  solution point and 0 at the others, and  $\delta$  denotes that the function is an approximation.

Note that the piecewise polynomials might not be continuous (or  $C^0$ ) across the interfaces. In the Flux Reconstruction approach, the flux used in the time advancement of the solution is made  $C^0$  by introducing flux correction functions.

This can be achieved by finding interface solution values at each element boundary and then correcting the solution. Let  $\hat{u}_L^{\delta I}$  and  $\hat{u}_R^{\delta I}$  be the interface solution values at left and right boundaries of some element, respectively.  $\hat{u}_L^{\delta I}$  and  $\hat{u}_R^{\delta I}$  can be found with a Riemann solver for Discontinuous-Galerkin (DG) methods<sup>7</sup>. Then, select solution correction functions  $g_L$  and  $g_R$  such that

$$g_L(-1) = 1, \quad g_L(1) = 0 \quad (18)$$

$$g_R(-1) = 0, \quad g_R(1) = 1 \quad (19)$$

and let

$$\hat{u}^C = \hat{u}^\delta + (\hat{u}_L^{\delta I} - \hat{u}_L^\delta)g_L + (\hat{u}_R^{\delta I} - \hat{u}_R^\delta)g_R \quad (20)$$

where superscript  $C$  denotes the function is corrected, and  $\hat{u}_L^\delta, \hat{u}_R^\delta$  represent the solution approximation evaluated at the left and right boundaries.

Using the values of  $\hat{u}_i^\delta$  and  $\frac{\partial \hat{u}_i^C}{\partial \xi}|_{\xi_i}$  we then find

$$\hat{f}_i^\delta = \hat{f} \left( \hat{u}_i^\delta, \frac{1}{J_n} \frac{\partial \hat{u}^C}{\partial \xi} \Big|_{\xi_i} \right) \text{ in element } \Omega_n$$

We can proceed in a similar fashion to correct the flux to obtain

$$\hat{f}^C = \hat{f}^\delta + (\hat{f}_L^{\delta I} - \hat{f}_L^\delta)h_L + (\hat{f}_R^{\delta I} - \hat{f}_R^\delta)h_R \quad (21)$$

where  $h_R$  and  $h_L$  are right and left flux correction functions satisfying the same boundary conditions as  $g_R$  and  $g_L$ , respectively, and  $\hat{f}_L^{\delta I}$  and  $\hat{f}_R^{\delta I}$  are the interface fluxes found via a Riemann solver. Note that if the flux corresponds to linear advection, correcting the solution and correcting the flux are equivalent steps.

The solution can then be advanced at each solution point. In semi-discrete form, this is

$$\frac{d\hat{u}_i^\delta}{dt} = -\frac{\partial \hat{f}^c}{\partial \xi}(\xi_i) \quad (22)$$

The FR scheme can be made provably stable for the linear advection-diffusion equation by selecting special types of correction functions<sup>7</sup>. In general, these correction functions are polynomials of degree  $p+1$  so both sides in Equation (22) are quantities related to polynomials of order  $p$  –for consistency<sup>7</sup>.

Vincent et al.<sup>7</sup> have shown that in the case of the 1-dimensional, linear advection equation, the Flux Reconstruction approach can be proven to be stable for a specific family of correction functions parameterized by a scalar called  $c$ . In addition, they showed that by selecting specific values of  $c$  it is possible to recover a particular nodal DG and Spectral Difference (SD) methods plus a FR scheme that was previously found to be stable by Huynh<sup>7</sup>.

## B. Unstructured Mesh Treatment

Extension to multiple dimensions requires formulating multi-dimensional interpolation functions and correction functions that satisfy boundary conditions equivalent to those in Equation (18) for each type of element.

Interpolation bases for quadrilaterals and hexahedra can be obtained via tensor products of the 1-dimensional interpolation basis. More concretely, in HiFiLES, we discretize the solution in 3-dimensions in the following way

$$\hat{u}^\delta(\xi, \eta, \zeta) = \sum_{i=1}^{p+1} \sum_{j=1}^{p+1} \sum_{k=1}^{p+1} \hat{u}_{i,j,k}^\delta l_i(\xi) l_j(\eta) l_k(\zeta) \quad (23)$$

where  $i, j, k$  index the solution points along the  $\xi, \eta, \zeta$  directions, respectively. The flux is discretized similarly.

The interpolation basis for triangles are described in detail by Castonguay et al.<sup>7</sup> and Williams et al.<sup>7</sup>. The formulation for tetrahedra is detailed by Williams et al.<sup>7</sup>.

The extension of interpolation polynomials to prisms is obtained via tensor products of the 1-dimensional basis with the triangular basis<sup>7</sup>.

In general, the boundary conditions for the correction functions in multiple dimensions can be formulated as

$$\mathbf{h}_i(\vec{\xi}_j) \cdot \mathbf{n}_j = \delta_{ij} \quad (24)$$

where  $\mathbf{h}_i$  is the vector of correction functions associated with interface point  $i$ ,  $\vec{\xi}_j$  is the location vector of the  $j^{\text{th}}$  interface point,  $\mathbf{n}_j$  is the outward unit normal at interface point  $j$ , and  $\delta_{ij}$  is the Kronecker delta. Interface points are located on the boundary of an element.

One of the challenges in the FR approach is finding correction functions that not only satisfy Equation (24) but also guarantee stability in the linear advection-diffusion case. Correction functions that guarantee such stability exist for 1-dimensional segments<sup>?</sup>, triangles<sup>??</sup>, and tetrahedra<sup>?</sup>. FR schemes with these correction functions comprise the ESFR family of schemes.

Although formal proofs of stability for the linear advection equation do not exist yet for quadrilaterals, hexahedra, and prisms, it has been observed that the tensor products of provably stable correction functions used in these elements maintain stability. In addition, as of now HiFiLES does not have an implementation for pyramidal elements, mostly because of the challenges involved in finding the respective correction functions that guarantee stability. Nevertheless, a suggested approach to find such correction functions has been presented by Jameson<sup>?</sup>.

### C. Large Eddy Simulation Models

In order to resolve all the scales of motion in a high Reynolds number turbulent flow, the computational mesh would have to be impractically fine. A practical solution is to employ the Large Eddy Simulation (LES) formulation, which only resolves the larger scales of motion and thus allows for the use of coarser meshes. The effect of the unresolved or subgrid-scale (SGS) dynamics on the solution is accounted for by an SGS model for the *subgrid-scale stress*  $\bar{\tau}_{sgs}$ , which is added to the viscous stress tensor  $\bar{\tau}$  given by (??):

$$\bar{\tau} = \mu \left( \nabla \vec{v} + \nabla \vec{v}^T - \frac{2}{3} \bar{I} (\nabla \cdot \vec{v}) \right) + \bar{\tau}_{sgs}. \quad (25)$$

The standard Smagorinsky model<sup>?</sup> is available in HiFiLES:

$$\bar{\tau}_{sgs} = 2\rho\nu_t \bar{S}^d, \quad (26)$$

$$\nu_t = C_S^2 \Delta^2 |\bar{S}^d|, \quad (27)$$

$$\bar{S}^d = \frac{1}{2} \left( \nabla \vec{v} + (\nabla \vec{v})^T - \frac{2}{3} \bar{I} (\nabla \cdot \vec{v}) \right), \quad (28)$$

where the Smagorinsky coefficient  $C_S = 0.1$  and the filter width  $\Delta$  is a measure of the size of an element. The quantity  $\nu_t$  is the eddy viscosity. In HiFiLES the filter width is given by (in 3D):

$$\Delta = \alpha(\text{vol})^{1/3}, \quad (29)$$

where  $\alpha \geq 1$  is a user-defined scaling factor and  $\text{vol}$  is the element volume. HiFiLES also includes the Wall-Adapting Local Eddy-Viscosity (WALE) model<sup>?</sup> and the Similarity model<sup>?</sup>. The Similarity model incorporates a filtering operator that locally smoothes the solution in order to extract small-scale information. Several filters are available in HiFiLES: a discrete Gaussian filter<sup>?</sup>, a high-order commuting Vasilyev-type filter<sup>??</sup> and a modal Vandermonde-type filter<sup>?</sup>. The modal filter is designed to operate on unstructured tetrahedral meshes. For details of these operators, see Lodato, Castonguay and Jameson<sup>?</sup> and Bull and Jameson<sup>?</sup>. Also available is the option to combine the similarity model with the Smagorinsky or WALE model to form a mixed SGS model. The WALE-similarity mixed (WSM) model, first proposed by Lodato et al.<sup>?</sup>, was used in simulations of the flow over a square cylinder (see Section F.). Finally, a Spectral Vanishing Viscosity (SVV) model is available. Instead of adding a model term, the SVV method applies one of the above mentioned filters directly to the solution in order to damp the small scales or high-order modes<sup>?</sup>.

### D. Computing Architecture and Scalability

The HiFiLES code has been designed to work on multi-CPU as well as multi-CPU-GPU platforms. The Flux Reconstruction method in its current form with explicit time-stepping has a great potential for parallelization. Since the solution points are not explicitly shared between elements, most of the computations are element-local enabling an

efficient use of shared memory on GPUs. Also, several computations are independent for each solution point and the highly parallelizable nature of GPUs becomes very useful. A detailed description of the parallelization of the FR method, along with scalability and performance analysis has been performed in<sup>7</sup>.

### E. Shock capturing and Stabilization Models

Currently, we have adapted Persson and Peraire's method<sup>7</sup> for shock detection and capturing. The method works well for inviscid flow cases and compressible flows up to a Mach number of 2 have been tried and tested. However the method requires fine-tuning of parameters for each problem and we are currently working on a new method which is more robust. Persson and Peraire have used this shock capturing tool as a stabilization method as well in their turbulence calculations and we plan to test this and present results for inviscid and viscous cases in the final draft. Here we show a few inviscid results for the case of  $M = 1.6$  where we illustrate the ability to detect shocks and add dissipation in a local manner in the form of artificial viscosity in order to capture the shock in a fine manner and avoid the loss of accuracy away from the shock.

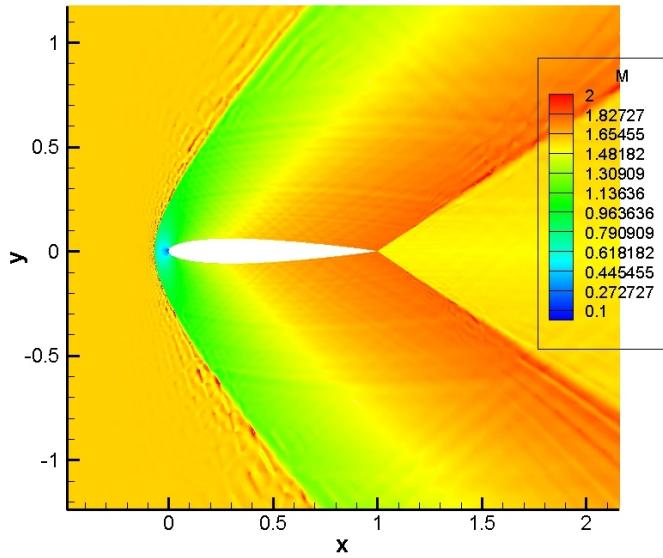


Figure 1: Mach contours for inviscid flow over Naca0012 at  $M = 1.6$  and  $\text{AoA} = 0^\circ$

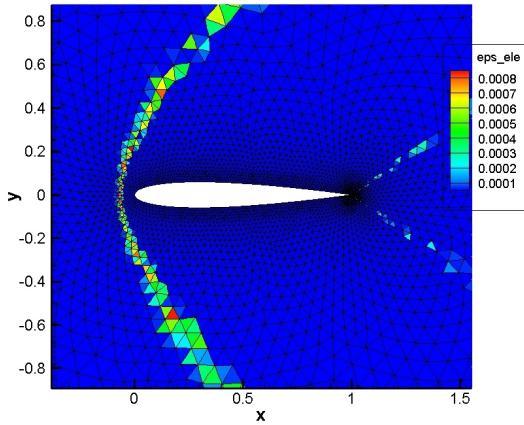


Figure 2: Element-wise AV co-efficients (case 1)

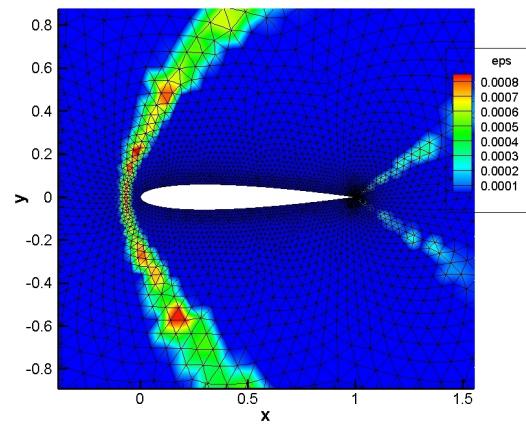


Figure 3: AV co-efficients with continuity enforcement

## F. Spalart-Allmaras (SA) Turbulence Model and Negative $\tilde{\nu}$ Modification

The one equation SA turbulence model is one of the more commonly used turbulence models used to solve attached and moderately separated aerodynamic flows<sup>7</sup>. The added equation directly solves for turbulent eddy viscosity via advection, diffusion, production and dissipation. A modified form of the equation can be written as<sup>7,8,9</sup>:

$$\frac{\partial}{\partial t}(\rho\tilde{\nu}) + \nabla \cdot (\rho\tilde{\nu}\mathbf{u}) = c_{b1}\tilde{S}\rho\nu\psi + \frac{1}{\sigma}[\nabla \cdot ((\mu + \mu\psi)\nabla\tilde{\nu}) + c_{b2}\rho\nabla\tilde{\nu} \cdot \nabla\tilde{\nu}] - c_{w1}\rho f_w \left(\frac{\nu\psi}{d}\right)^2 \quad (30)$$

$$\mu_t = \begin{cases} \rho\tilde{\nu}f_{v_1} & \text{if } \tilde{\nu} \geq 0 \\ 0 & \text{if } \tilde{\nu} < 0 \end{cases} \quad \text{where} \quad f_{v_1} = \frac{\left(\frac{\rho\tilde{\nu}}{\mu}\right)^3}{\left(\frac{\rho\tilde{\nu}}{\mu}\right)^3 + c_{v_1}^3} \quad (31)$$

$$\tilde{S} = \begin{cases} S + \bar{S} & \text{if } \bar{S} \geq -c_{v_2}S \\ S + \frac{S(c_{v_2}^2S + c_{v_3}\bar{S})}{(c_{v_3} - 2c_{v_2})S - \bar{S}} & \text{if } \bar{S} \leq -c_{v_2}S \end{cases} \quad (32)$$

$$S = \sqrt{\omega \cdot \omega} \quad \bar{S} = \frac{(\nu\psi)^2 f_{v_2}}{\kappa^2 d^2} \quad (33)$$

$$f_{v_2} = 1 - \frac{\psi}{1 + \psi f_{v_1}} \quad (34)$$

$$f_w = g \left[ \frac{1 + c_{w_3}^6}{g^6 + c_{w_3}^6} \right]^{1/6} \quad g = r + c_{w_2}(r^6 - r) \quad r = \frac{\nu\psi}{\tilde{S}\kappa^2 d^2} \quad (35)$$

The diffusion term,  $\nabla \cdot (\rho\tilde{\nu}\mathbf{u})$ , may become discontinuous in the first derivative leading to oscillations in high-order polynomials. This can lead to large negative values of the modified eddy viscosity term,  $\tilde{\nu}$ , significant enough to cause an unbounded solution. To prevent this, the following modification is introduced<sup>7</sup>.

$$\psi = \begin{cases} 0.05\log(1.0 + e^{(20.0\chi)}) & \text{if } \chi \leq 10.0 \\ \chi & \text{if } \chi > 10.0 \end{cases} \quad (36)$$

$$\chi = \frac{\tilde{\nu}}{\nu} \quad (37)$$

## IV. Verification

This section describes the check of spatial and temporal order of accuracy using the Method of Manufactured Solutions in 2D and 3D for viscous and inviscid flows, characterization of stable time-step limits, assessment of computational cost per degree of freedom for a given error tolerance, and measurement of weak and strong scalability in GPUs and CPUs.

## A. Method of Manufactured Solutions

$c, \kappa$	Grid	$L_2$ err.	$L_{2s}$ err.	$O(L_2)$	$O(L_{2s})$
$c_{dg}, \kappa_{dg}$	$N = 24$	8.53e-04	1.43e-01		
	$N = 32$	3.60e-04	7.95e-02	2.98	2.00
	$N = 48$	1.01e-04	3.30e-02	3.16	2.18
	$N = 64$	3.98e-05	1.72e-02	3.27	2.30
$c_+, \kappa_+$	$N = 24$	5.65e-03	7.23e-01		
	$N = 32$	2.34e-03	4.12e-01	3.07	1.99
	$N = 48$	6.41e-04	1.79e-01	3.19	2.09
	$N = 64$	2.48e-04	9.61e-02	3.30	2.21

Figure 4: Accuracy of ESFR schemes for flow generated by a time-dependent source term on triangular grids, for the case of  $p = 2$ . The inviscid and viscous numerical fluxes were computed using a Rusanov flux with  $\lambda = 1$  and a LDG flux with  $\tau = 0.1$  and  $\beta = \pm 0.5n$ .

$c, \kappa$	Grid	$L_2$ err.	$L_{2s}$ err.	$O(L_2)$	$O(L_{2s})$
$c_{dg}, \kappa_{dg}$	$N = 24$	1.27e-05	4.49e-03		
	$N = 32$	4.02e-06	1.86e-03	3.99	3.06
	$N = 48$	7.99e-07	5.35e-04	3.97	3.06
	$N = 64$	2.54e-07	2.26e-04	3.96	2.99
$c_+, \kappa_+$	$N = 24$	9.48e-05	1.71e-02		
	$N = 32$	2.86e-05	6.87e-03	4.17	3.16
	$N = 48$	5.27e-06	1.91e-03	4.17	3.15
	$N = 64$	1.58e-06	7.73e-04	4.18	3.15

Figure 5: Accuracy of ESFR schemes for flow generated by a time-dependent source term on triangular grids, for the case of  $p = 3$ . The inviscid and viscous numerical fluxes were computed using a Rusanov flux with  $\lambda = 1$  and a LDG flux with  $\tau = 0.1$  and  $\beta = \pm 0.5n$ .

	$p = 2$	$p = 3$
$c_{dg}, \kappa_{dg}$	2.67e-03	1.70e-03
$c_+, \kappa_+$	4.39e-03	2.35e-03

Figure 6: Explicit time-step limits ( $\Delta t_{max}$ ) of ESFR schemes for flow generated by a time-dependent source term on the triangular grid with  $\tilde{N} = 48$ , for the cases of  $p = 2$  and 3. The inviscid and viscous numerical fluxes were computed using a Rusanov flux with  $\lambda = 1$  and a LDG flux with  $\tau = 0.1$  and  $\beta = \pm 0.5n$ .

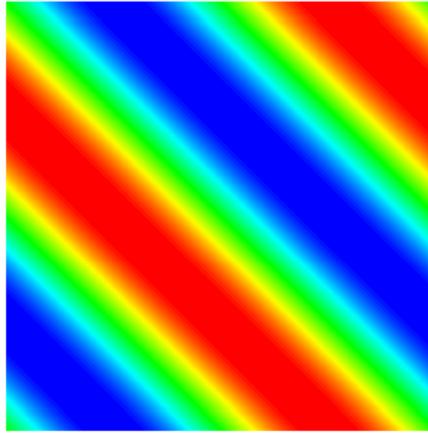


Figure 7: Contours of energy obtained using the ESFR scheme with  $c = c_+$  and  $\kappa = \kappa_+$  on the triangular grid with  $\tilde{N} = 32$  for the case of  $p = 3$ . The inviscid and viscous numerical fluxes were computed using a Rusanov flux with  $\lambda = 1$  and a LDG flux with  $\tau = 0.1$  and  $\beta = \pm 0.5n$ .

$c, \kappa$	Grid	$L_2$ err.	$L_{2s}$ err.	$O(L_2)$	$O(L_{2s})$
$c_{dg}, \kappa_{dg}$	$N = 24$	2.81e-03	4.98e-01		
	$N = 32$	1.20e-03	2.78e-01	2.95	2.03
	$N = 48$	3.58e-04	1.20e-01	2.99	2.07
	$N = 64$	1.48e-04	6.54e-02	3.07	2.11
$c_+, \kappa_+$	$N = 24$	1.05e-01	1.06e+01		
	$N = 32$	4.85e-02	6.45e+00	2.68	1.72
	$N = 48$	1.53e-02	3.02e+00	2.85	1.87
	$N = 64$	6.40e-03	1.69e+00	3.03	2.02

Figure 8: Accuracy of ESFR schemes for flow generated by a time-dependent source term on tetrahedral grids, for the case of  $p = 2$ . The inviscid and viscous numerical fluxes were computed using a Rusanov flux with  $\lambda = 1$  and a LDG flux with  $\tau = 0.1$  and  $\beta = \pm 0.5n$ .

$c, \kappa$	Grid	$L_2$ err.	$L_{2s}$ err.	$O(L_2)$	$O(L_{2s})$
$c_{dg}, \kappa_{dg}$	$N = 24$	8.78e-05	2.24e-02		
	$N = 32$	2.70e-05	9.11e-03	4.11	3.13
	$N = 48$	5.17e-06	2.57e-03	4.07	3.12
	$N = 64$	1.62e-06	1.06e-03	4.04	3.10
$c_+, \kappa_+$	$N = 24$	3.86e-03	4.90e-01		
	$N = 32$	1.29e-03	2.17e-01	3.80	2.84
	$N = 48$	2.62e-04	6.51e-02	3.94	2.96
	$N = 64$	8.14e-05	2.69e-02	4.06	3.08

Figure 9: Accuracy of ESFR schemes for flow generated by a time-dependent source term on tetrahedral grids, for the case of  $p = 3$ . The inviscid and viscous numerical fluxes were computed using a Rusanov flux with  $\lambda = 1$  and a LDG flux with  $\tau = 0.1$  and  $\beta = \pm 0.5n$ .

	$p = 2$	$p = 3$
$c_{dg}, \kappa_{dg}$	1.07e-03	7.35e-04
$c_+, \kappa_+$	1.81e-03	1.03e-03

Figure 10: Explicit time-step limits ( $\Delta t_{max}$ ) of ESFR schemes for flow generated by a time-dependent source term on the triangular grid with  $\tilde{N} = 48$ , for the cases of  $p = 2$  and  $3$ . The inviscid and viscous numerical fluxes were computed using a Rusanov flux with  $\lambda = 1$  and a LDG flux with  $\tau = 0.1$  and  $\beta = \pm 0.5n$ .

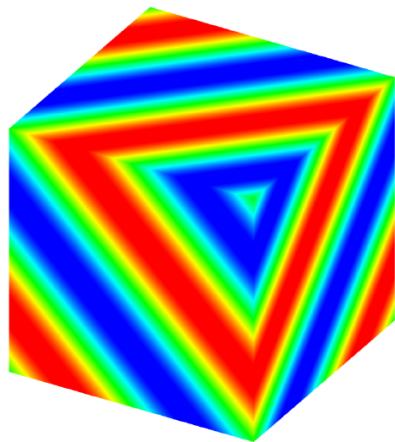


Figure 11: Contours of energy obtained using the ESFR scheme with  $c = c_+$  and  $\kappa = \kappa_+$  on the tetrahedral grid with  $\tilde{N} = 32$  for the case of  $p = 3$ . The inviscid and viscous numerical fluxes were computed using a Rusanov flux with  $\lambda = 1$  and a LDG flux with  $\tau = 0.1$  and  $\beta = \pm 0.5n$ .

## **V. Validation**

### A. Subsonic laminar flat-plate

Computations of the flow over a subsonic flat-plate have been performed and validated against the Blasius' solution for laminar boundary layer. The flow conditions are Mach number 0.5, angle of attack 0.0 deg and Reynolds number based on the plate length of  $1 \cdot 10^6$ . The governing equations are the 2D Navier-Stokes equations with constant ratio of specific heats of 1.4, Prandtl number of 0.72 and constant dynamic viscosity of  $1.827 \cdot 10^{-5} Pa \cdot s$ .

Height first cell & # of cells inside the boundary layer	Order 2	Order 3	Order 4	Order 5
Mesh a0 (140 = 14x10). 0.00075 / 2 cells	✗	✗	✗	✓
Mesh a1 (560 = 28x20). 0.000375 / 4 cells	✗	✗	✓	✓
Mesh a2 (2240 = 56x40). 0.0001875 / 8 cells	✗	✓	✓	✓
Mesh a3 (8960 = 112x80). 0.0000935 / 16 cells	✓	✓	✓	✓

Table 1: HiFiLES convergence using different grids and polynomial order

The objective of this study is to determine the minimum number of elements and the order of polynomial required to converge the flat-plate simulation using HiFiLES. In particular, 4 different numerical grids have been used in this study (2, 4, 8, 16 elements inside the boundary layer). The results are summarized in Table ??, the simulations require a minimum number of elements in the boundary layer to obtain a satisfactory converge otherwise, there are important jumps across the elements.

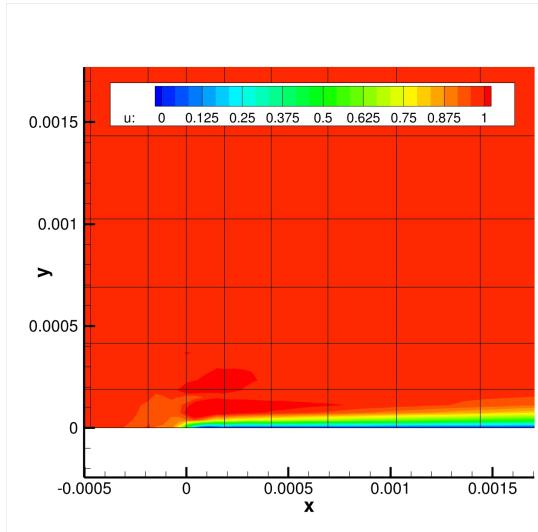


Figure 12: Detail of the flat-plate leading edge ( $x=0.0$ , mesh a2).

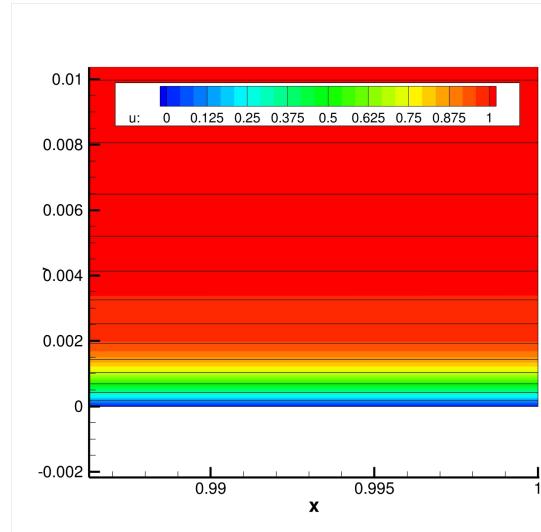


Figure 13: Flow solution at the end of the flat-plate ( $x=1.0$ , mesh a2).

The results has been compared with the Blasius' solution for laminar boundary layer with satisfactory results, and some details of the solutions are presented in Fig. 12 (leading edge), and Fig. 13 (end of the flat-plate). It is important to note that in this particular case (mesh a2) the flap-plate is captured using 8 elements, while in a second order solver it would be necessary of the order of 30 elements inside the boundary layer.

To finalize, it is critical to note that the absence of a local time stepping technique in HiFiLES increases the required number of iterations to obtain a converged solution. However, we have noticed an improvement of the rate of converge as we refine the grid (see Fig. 14), and this convergence rate is comparable to a second order numerical code (e.g. SU<sup>2</sup>) running using a similar numerical time integration (see Fig.15).

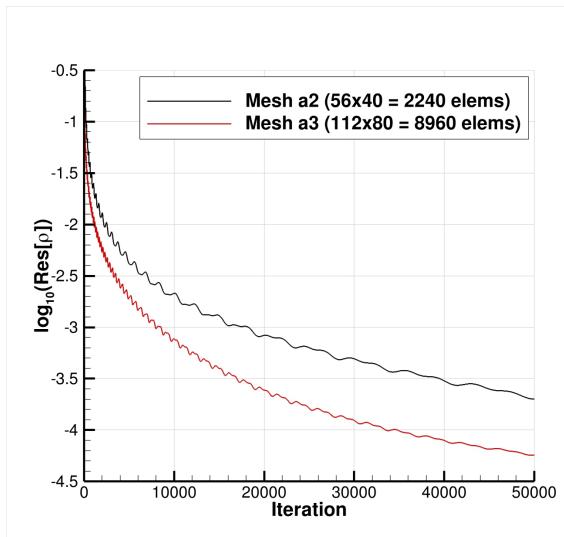


Figure 14: Convergence comparison ( $3^{rd}$  order, finest grids).

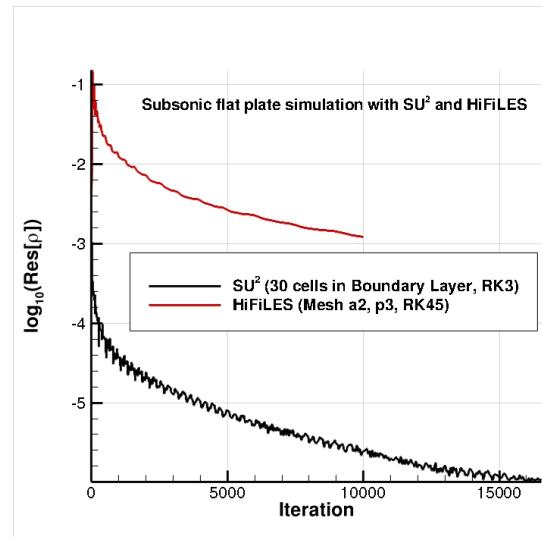


Figure 15: Comparison of HiFiLES with SU<sup>2</sup> using a similar time integration scheme.

## B. Circular Cylinder

$c, \kappa$	$M = 0.3$		$M = 0.5$	
	$\overline{C_D}$	$St$	$\overline{C_D}$	$St$
$c_{dg}, \kappa_{dg}$	1.3928	0.190	1.5669	0.190
$c_+, \kappa_+$	1.3923	0.190	1.5666	0.190

Figure 16: Values of the time-averaged drag coefficient and Strouhal number for the circular cylinder in flows with  $M = 0.3$  and  $M = 0.5$ . The flows were simulated using the VCJH schemes with  $p = 2, c = c_{dg}, \kappa = \kappa_{dg}$  and  $c = c_+, \kappa = \kappa_+$  in conjunction with the Rusanov flux with  $\lambda = 1$  and the LDG flux with  $\beta = 0.5n$  and  $\tau = 0.1$  on the unstructured triangular grid with  $N = 63472$  elements.

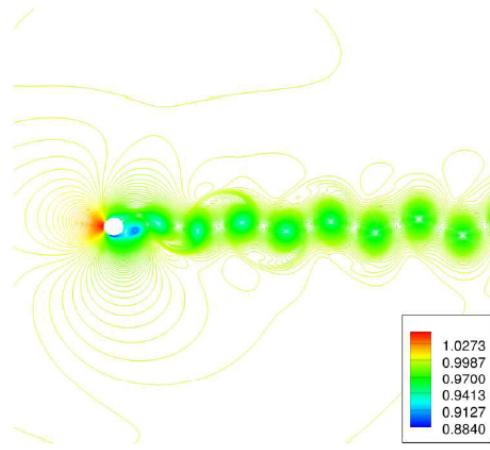


Figure 17: Density contour for the flow with  $M = 0.3$  around the circular cylinder.

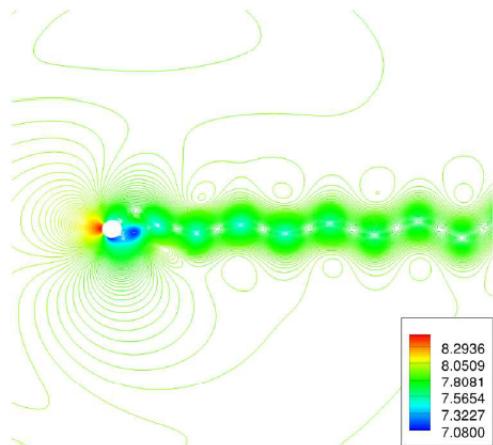


Figure 18: Pressure contour for the flow with  $M = 0.3$  around the circular cylinder.

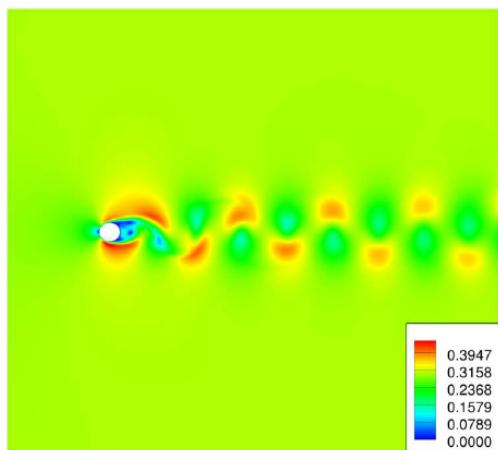


Figure 19: Mach contour for the flow with  $M = 0.3$  around the circular cylinder.

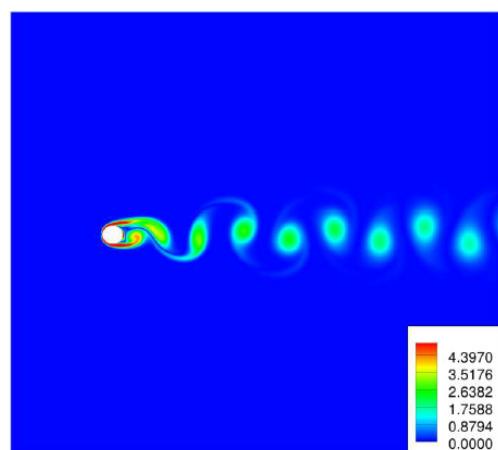


Figure 20: Vorticity contour for the flow with  $M = 0.3$  around the circular cylinder.

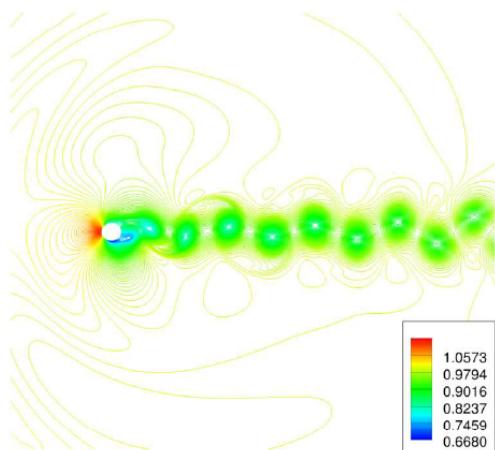


Figure 21: Density contour for the flow with  $M = 0.5$  around the circular cylinder.

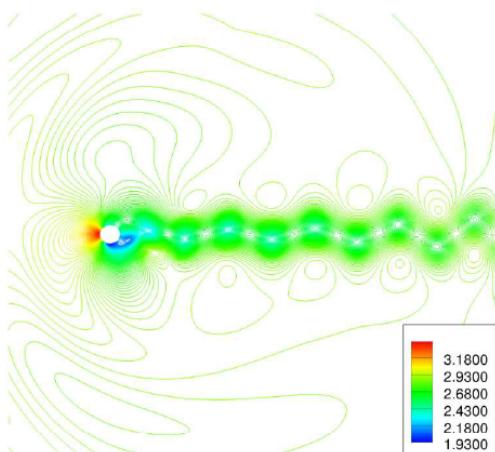


Figure 22: Pressure contour for the flow with  $M = 0.5$  around the circular cylinder.

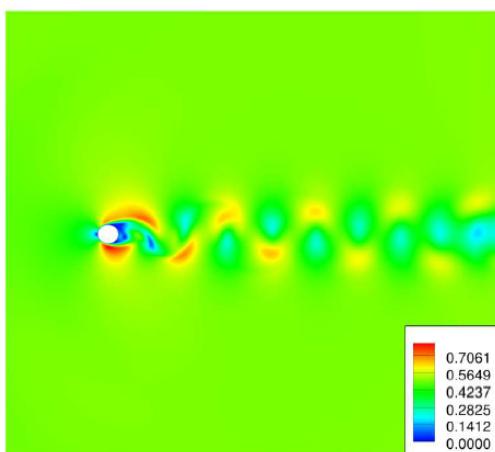


Figure 23: Mach contour for the flow with  $M = 0.5$  around the circular cylinder.

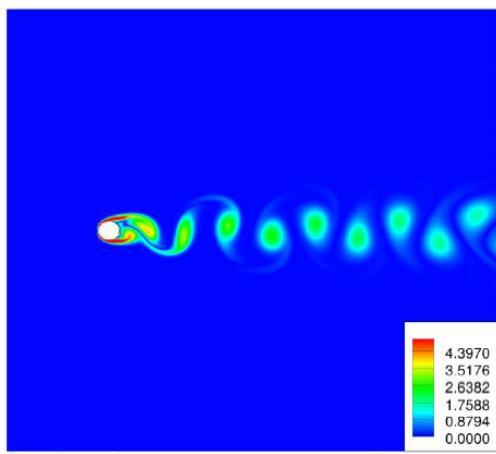


Figure 24: Vorticity contour for the flow with  $M = 0.5$  around the circular cylinder.

### C. SD7003 airfoil at 4° angle of attack

From Williams's thesis?

Source	$Re = 10K$	$\overline{C_L}$	$\overline{C_D}$	$Re = 22K$	$\overline{C_L}$	$\overline{C_D}$	$Re = 60K$	$\overline{C_L}$	$\overline{C_D}$
Uranga et al. [115]	0.3755	0.04978	0.6707	0.04510	0.5730	0.02097			
$c_{dg}, \kappa_{dg}$	0.3719	0.04940	0.6722	0.04295	0.5831	0.01975			
$c_+, \kappa_+$	0.3713	0.04935	0.6655	0.04275	0.5774	0.02005			

Figure 25: Time-averaged values of the lift and drag coefficients for the SD7003 airfoil flows with  $Re = 10000, 22000, 60000$

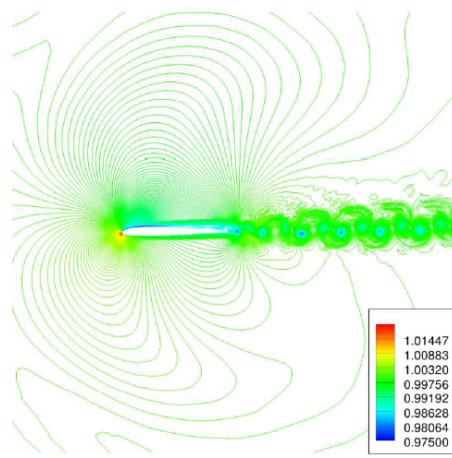


Figure 26: Density contour for the flow with  $Re = 10000$  around the SD7003 airfoil.

### D. SD7003 wing section at 4° angle of attack

From David's thesis.

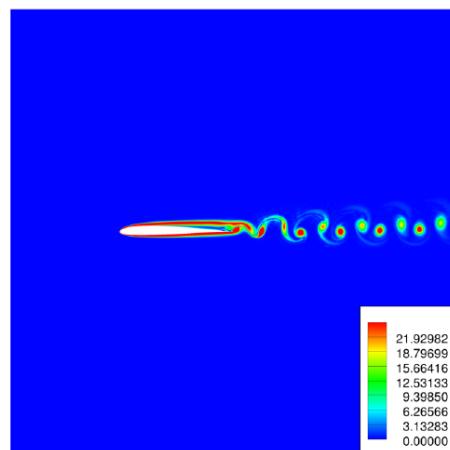


Figure 27: Vorticity contour for the flow with  $Re = 10000$  around the SD7003 airfoil.

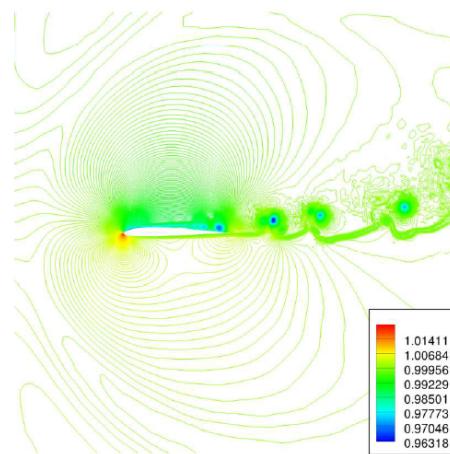


Figure 28: Density contour for the flow with  $Re = 22000$  around the SD7003 airfoil.

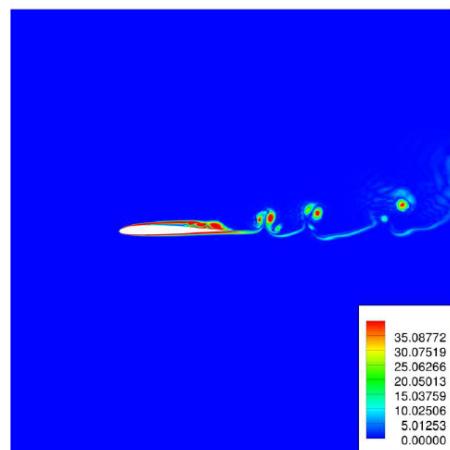


Figure 29: Vorticity contour for the flow with  $Re = 22000$  around the SD7003 airfoil.

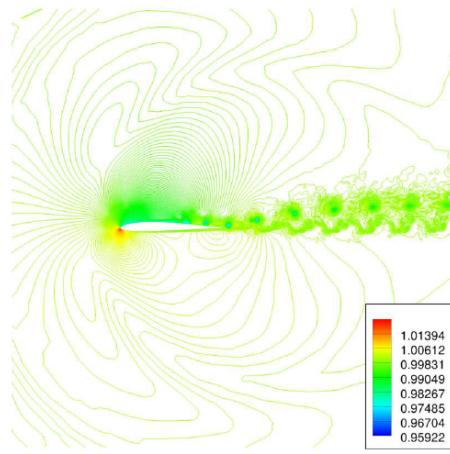


Figure 30: Density contour for the flow with  $Re = 60000$  around the SD7003 airfoil.

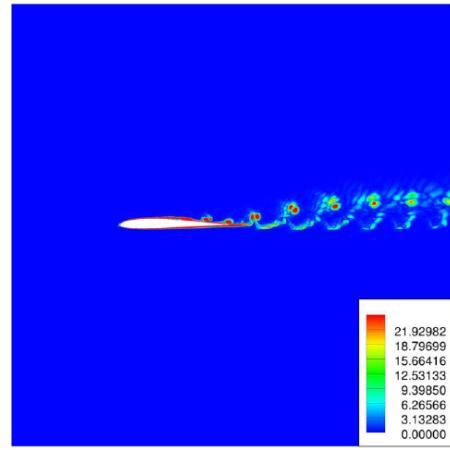


Figure 31: Vorticity contour for the flow with  $Re = 60000$  around the SD7003 airfoil.

Source	$Re = 10K$	
	$\overline{C_L}$	$\overline{C_D}$
Uranga et al. [115]	0.3743	0.04967
$c_{dg}, \kappa_{dg}$	0.3466	0.04908
$c_+, \kappa_+$	0.3454	0.04903

Figure 32: Time-averaged values of the lift and drag coefficients for the SD7003 wing-section in a flow with  $Re = 10000$

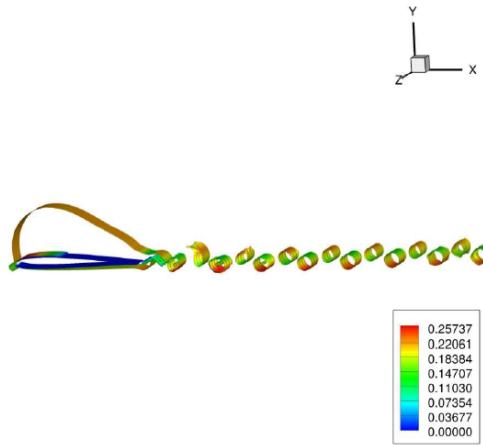


Figure 33: Density isosurfaces colored by Mach number for the flow with  $Re = 10000$  around the SD7003 wing-section.

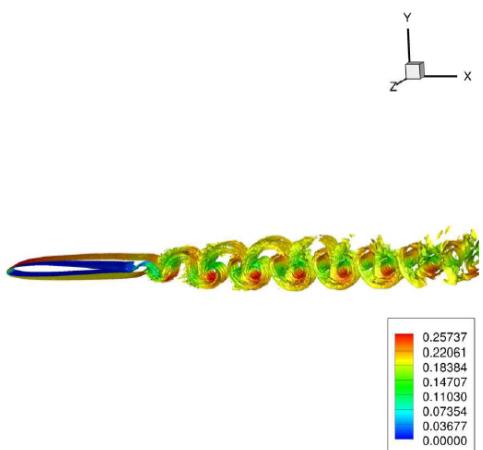


Figure 34: Vorticity isosurfaces colored by Mach number for the flow with  $Re = 10000$  around the SD7003 wing-section.

## E. Taylor-Green Vortex at $Re = 1,600$

The Taylor-Green Vortex (TGV) is a simple test of the resolution of the small scales of a turbulent flow by a numerical method. The compressible TGV at  $Re = 1600$  was one of the benchmark problems in the 1st and 2nd International Workshops on High-Order CFD Methods<sup>7</sup>. A reference solution was computed by Debonis<sup>7</sup> using a high-order dispersion relation-preserving (DRP) scheme on a mesh of  $512^3$  elements. The results presented here were obtained by Bull and Jameson using FR to recover the fourth-order-accurate DG and SD schemes in HiFiLES<sup>7,8</sup>. We also compare our results to those of Beck and Gassner<sup>9</sup>, who used a fourth-order filtered DG method on a mesh of  $64^3$  elements. From a simple initial condition in a triply-periodic box of dimensions  $[0 : 2\pi]^3$ , interactions between vortices cause the flow to develop in a prescribed manner into a mass of elongated vortices across a range of scales. The initial condition is specified as

$$u(t_0) = u_0 \sin(x/L) \cos(y/L) \cos(z/L), \quad (38)$$

$$v(t_0) = -u_0 \cos(x/L) \sin(y/L) \cos(z/L), \quad (39)$$

$$w(t_0) = 0, \quad (40)$$

$$p(t_0) = p_0 + \frac{\rho_0 V_0^2}{16} \left[ \cos\left(\frac{2x}{L}\right) + \cos\left(\frac{2y}{L}\right) \right] \left[ \cos\left(\frac{2z}{L}\right) + 2 \right], \quad (41)$$

where  $L = 1$ ,  $u_0 = 1$ ,  $\rho_0 = 1$  and  $p_0 = 100$ . The Mach number is set to 0.08 (consistent with the initial pressure  $p_0$ ) and the initial temperature is 300K.

Figs. 35 (a) and (b) show the volume-averaged kinetic energy  $\langle k \rangle$  on (a) hexahedral meshes of  $16^3$ ,  $32^3$  and  $64^3$  elements and (b) tetrahedral meshes (formed by splitting the hexahedral meshes). The reference solution, labelled as ‘DRP-512’ is plotted for comparison. Figs. 35 (c) and (d) show the kinetic energy dissipation rate, given by  $\epsilon = -d\langle k \rangle / dt$  versus the reference solution and the results of Beck and Gassner<sup>9</sup>, labelled as ‘Beck-DG-64x4’. On the finest hexahedral and tetrahedral meshes the kinetic energy and dissipation rate predictions match the reference solution, demonstrating that the high-order numerical scheme is able to resolve the important flow dynamics on a relatively coarse mesh. As a qualitative measure of the resolution of the turbulent flow structures, Figure 36 shows isosurfaces of the  $q$  criterion at four times during the simulation. The evolution of complex small scale structures is evident.

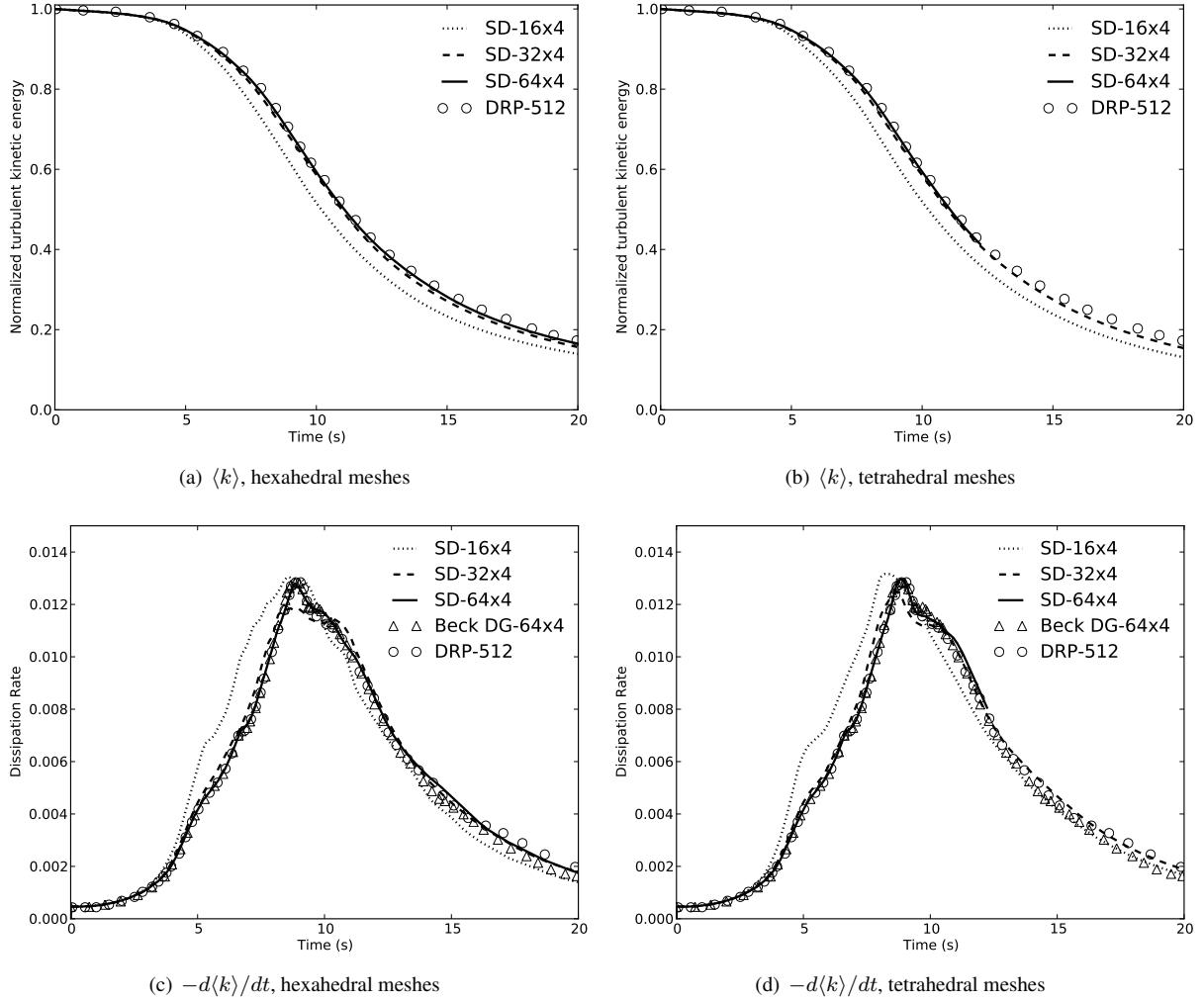


Figure 35: Taylor-Green vortex results on hexahedral and tetrahedral meshes from Bull and Jameson<sup>7</sup>. (a, b) Evolution of average kinetic energy  $\langle k \rangle$ ; (c, d) dissipation rate  $-d\langle k \rangle/dt$ . ‘SD- $M \times N$ ’ refers to  $M^3$  mesh,  $N$ th-order accurate SD scheme. (---) 4th-order DG on  $64^3$  mesh<sup>7</sup>; (○) DNS<sup>7</sup>.

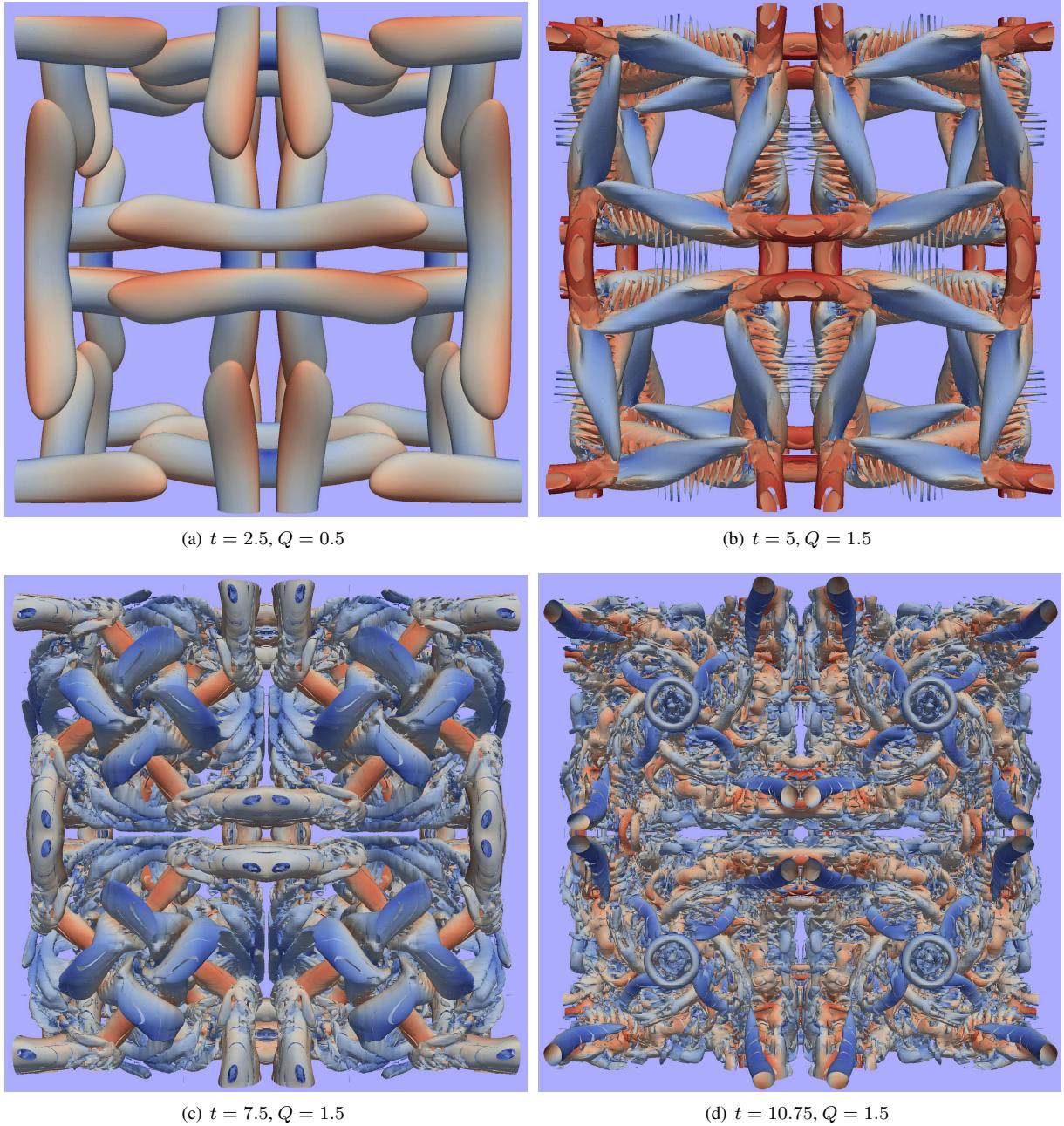


Figure 36: TGV solution on the fine mesh using fourth order accurate DG method, showing isosurfaces of  $q$  criterion colored by velocity magnitude at time  $t = 2.5$  to  $10.75$  seconds.

## F. LES of Flow Over a Square Cylinder at $Re = 21,400$

Using the FR method to recover the fourth-order accurate SD scheme, the flow over a square cylinder of side  $D$  in a domain of  $21D \times 12D \times 3.2D$  at  $Re = 21,400$  and Mach 0.3 was simulated, for which LDV experimental data is available<sup>??</sup>. A tetrahedral mesh of 87,178 elements was generated giving a total of 1.74M degrees of freedom (DoF) since there are 20 solution points per element at fourth order accuracy. Time discretization was by the fourth-order five-stage explicit RK scheme. A total time of 250 seconds was simulated and time-averaged quantities were calculated over the last 100 seconds (approx. 5 flow-through periods). The WSM model (see Section C.) based on the modal Vandermonde filter<sup>?</sup> was used with the Breuer-Rodi three-layer wall model<sup>?</sup> within 0.2D of the wall. The computation took around 60 hours on 7 GPUs in the lab's own cluster. Figure 37 shows the computational mesh including all the DoF. Figure 38 shows an isosurface of the  $q$ -criterion colored by velocity magnitude, illustrating the structures present in the turbulent boundary layer and wake. Figures 39 (a, b) show the normalized mean streamwise and vertical velocity components  $\langle u \rangle / u_B$  and  $\langle v \rangle / u_B$  respectively along several vertical lines in the wake. Figures 39 (c, d) show the normalized mean Reynolds stress components  $\langle u' u' \rangle / u_B^2$  and  $\langle u' v' \rangle / u_B^2$  along the same lines. For comparison, high-order LES results computed by Lodato and Jameson<sup>?</sup> using the SD method and the WSM model on a hexahedral mesh of 2.3M DoF are plotted. Mean velocities are accurately predicted although the accuracy is reduced near the cylinder owing to the coarse tetrahedral resolution in the boundary layer. The Reynolds stresses are less accurately predicted than the mean velocities but are broadly correct. These results highlight the advantages of using HiFiLES for LES of turbulent flows: the ability to obtain good results on coarse meshes and the ability to use unstructured tetrahedral meshes.

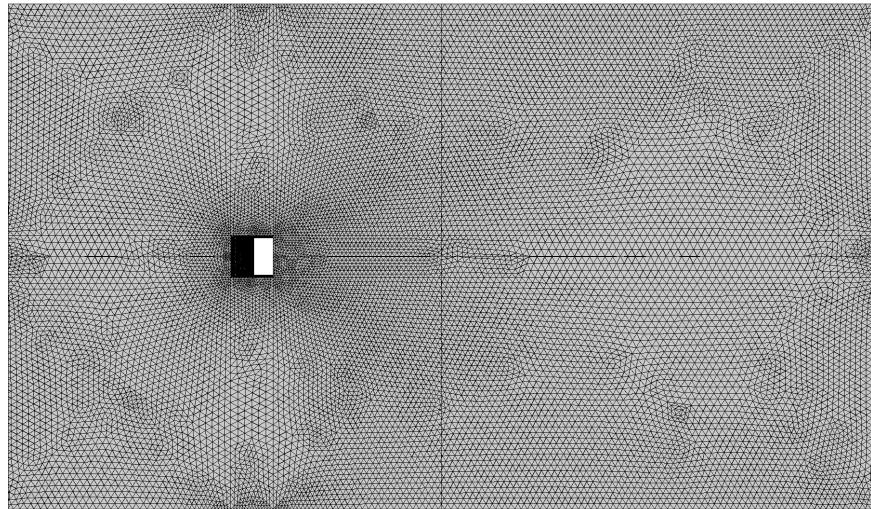


Figure 37: Tetrahedral mesh used for LES of the square cylinder showing all degrees of freedom



Figure 38: Isosurface of the  $q$ -criterion colored by velocity magnitude showing the wake behind the square cylinder

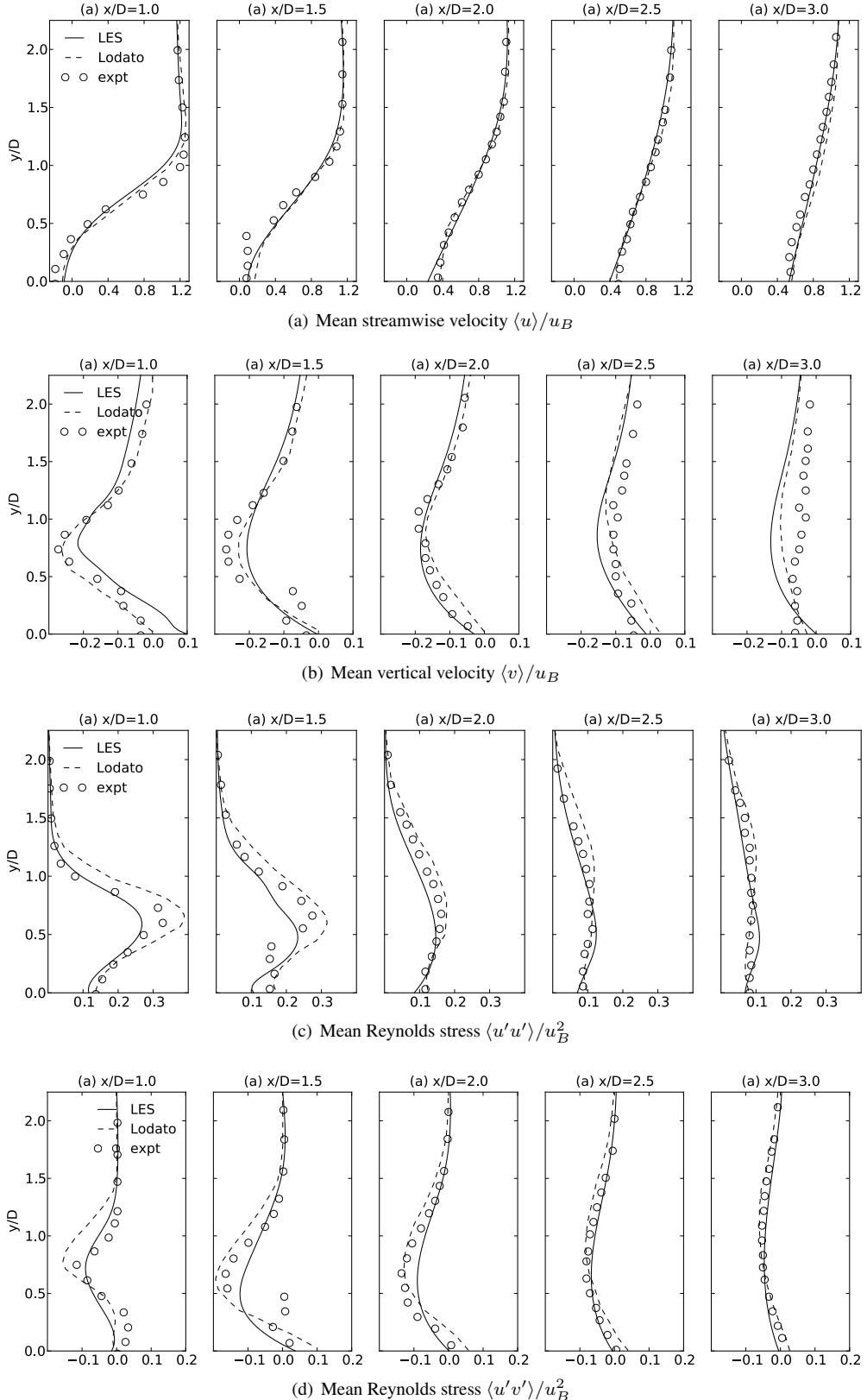


Figure 39: (a) Mean streamwise and vertical velocity and mean Reynolds stresses along vertical lines in the wake. (—) current results, (---) 4th order SD+WSM on hexahedral mesh by Lodato and Jameson<sup>7</sup>, (○) LDV experiments by Lyn et al.<sup>??</sup>.

#### **G. NACA 0012 airfoil at 0° angle of attack, Re = 6 million, Ma = 0.15**

In this section, the NACA 0012 airfoil is used to study the accuracy of the SA turbulence model coupled with FR. The NACA 0012 is commonly used as a validation case for all turbulence models and a large database of results are available at the NASA Turbulence Modeling Resource website. A 6,539 element quad/triangle mixed mesh is used with a NACA 0012 airfoil of chord length 1.0 and a farfield boundary 20 chord lengths away.

## **VI. Conclusion**

Conclusions here...

## **Acknowledgements**

The authors would like to acknowledge the support for this work provided by the Stanford Graduate Fellowships, National Science Foundation Graduate Fellowships, and the Air Force Office of Scientific Research.