KTH Computer Science
and Communication

An adaptive finite element method for the compressible Euler equations

MURTAZO NAZAROV

Licentiate Thesis
Stockholm, Sweden 2009
http://www.kth.se/csc/publikationer/Licavhandlingar

TRITA-CSC-A 2009:13
ISSN-1653-5723 KTH School of Computer Science and Communication
ISRN-KTH/CSC/A–09/13-SE SE-100 44 Stockholm
ISBN 978-91-7415-365-1 SWEDEN

Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framlägges till offentlig granskning för avläggande av teknologie licentiatexamen i numerisk analys önsdagen den 10 juni 2009 klockan 14.15 i sal D42, Lindstedtsvägen 5, Kungliga Tekniska högskolan, Stockholm.

Tryck: Universitetsservice US AB

To the memory of my sister...
Ба хотирам Ҳаиста

Abstract

This work develops a stabilized finite element method for the compressible Euler equations and proves an a posteriori error estimate for the approximated solution. The equations are approximated by the cG(1)cG(1) finite element method with continuous piecewise linear functions in space and time. cG(1)cG(1) gives a second order accuracy in space, and corresponds to a Crank-Nicholson type of discretization in time, resulting in second order accuracy in space, without a stabilization term.

The method is stabilized by componentwise weighted least squares stabilization of the convection terms, and residual based shock capturing. This choice of stabilization gives a symmetric stabilization matrix in the discrete system. The method is successfully implemented for a number of benchmark problems in 1D, 2D and 3D. We observe that cG(1)cG(1) with the above choice of stabilization is robust and converges to an accurate solution with residual based adaptive mesh refinement.

We then extend the General Galerkin framework from incompressible to compressible flow, with duality based a posteriori error estimation of some quantity of interest. The quantities of interest can be stresses, strains, drag and lift forces, surface forces or a mean value of some quantity. In this work we prove a duality based a posteriori error estimate for the compressible equations, as an extension of the earlier work for incompressible flow [25].

The implementation and analysis are validated in computational tests both with respect to the stabilization and the duality based adaptation.

Preface

This thesis consists of an introduction and two papers.

Paper I.

Johan Hoffman, Johan Jansson and Murtazo Nazarov. A General Galerkin Finite Element Method for the Compressible Euler Equations, *submitted to SIAM Journal of Scientific Computing*, 2008.

The author of this thesis contributed to the ideas on the numerical method and implemented the computational code. The author wrote the sections 5, 6, 7 and partly the sections 1 and 2 of the manuscript. The author presented a part of the paper on implementation of algorithms at FEniCS '08 workshop, LSU, USA, 2008. Also he presented the full paper at ECCOMAS 2008 conference, Venice, Italy.

Paper II.

Murtazo Nazarov and Johan Hoffman. An Adaptive Finite Element Method for the Compressible Euler Equations, *to be submitted*, 2009.

The author of this thesis contributed to the ideas, wrote the manuscript and performed the implementations. The author presented a part of this work at the FEF '09 conference, Tokyo, Japan.

Acknowledgement

I would like to say thanks to my advisor Johan Hoffman for his guidance, help and encouragement. Your deep knowledge in computational fluid dynamics and mathematics and devoted times for throughout discussions are great assets in accomplishing this thesis. Thank you for giving me an opportunity to work on such interesting subject and get familiar with the world of finite elements. You always listened to and gave me valuable ideas, taught me how to find good strategies on understanding the subject and dealing with a problem.

I would like also to thank Professor Jesper Oppelstrup and Marco Kupiainen for interesting discussions we had on conservation laws and compressible flows. Interesting ideas and encouragements in finite elements and in computational thermodynamics received from Professor Claes Johnson is gratefully acknowledged. These discussions served as a base for understanding the wonderful world of flows and thermodynamics. My special thanks go to Johan Jansson for various discussions on the FEniCS project and Unicorn's development, and also my colleagues in the Computational Technology Laboratory (CTL) and in the Numerical Analysis group.

Professors Muhamadiev Ergash Mirzoevich and Naimov Alijon from Vologda State University of Technology, Russia are gratefully acknowledged. Ergash Mirzoevich, I still miss your seminars, which I have been attending during my undergraduate education. Your seminars were the best source of knowledge and inspiration to mathematics in my life.

The special thanks goes to Nargis and Aziz, for thier understanding, support and encouragment throughout the whole period of this work.

Here, my great thanks goes to my family, my sister Maryam, brothers Muzaffar, Mustafo and Mustakim, my mother Muhabbat and father Zoidjon for your continuous love, support, encouragement and support in my education. You mean everything for me. This words are only for you: Ман хеле хушбахтам, ки тарбияи Шумо дадаҷон ва модарҷони азиз насиби ман гашт! Меҳри Шумо муқоисанашаванда аст!

Contents

List of Figures

Chapter 1

Background

1.1 Introduction

The compressible Euler equations is a system of conservation laws that consists of *conservation of mass, momentum and energy*. These equations describe inviscid flow in fluid dynamics, however they are also applicable as an approximation of flows with small *viscosity*. In the coming chapters we discuss phenomena that arises when the viscosity of the fluid decreases, with shock formation and *turbulence*. Turbulent flow is not well suited for analysis, instead we must use numerical approaches.

In general problems in science and industry involve complex geometries and shapes. Simulation of flow around cars, airplanes, rackets, wings, birds, supersonic jets are real examples which can be required from the industry. Therefore, the numerical method should be designed to capture complex geometries. For this reason, we use *finite element methods* because of stability, generality, and applicability to real applications.

When differential equations are solved numerically, one has to ensure that the method is *efficient* and *reliable*. The efficiency corresponds to solving a problem with minimal computational cost, and reliability guaranties that the global error can be estimated from the computed solution. This is also one main reason for using finite element methods, since it is mathematically well motivated with a framework for a posteriori error analysis. Overall, the objective of the thesis is to design and implement a robust stabilized method and to prove a posteriori error estimates for the compressible Euler equations.

The structure of the thesis is the following: First we give a brief background about conservation laws and the compressible Euler equations in Chapter

1. Then in Chapter 2 we continue our discussion on numerical methods for conservation laws, were our focus is the Euler equations. Chapter 3 gives information about duality based error estimation for nonlinear problems and in particular the main theorem for the compressible Euler equations. We discuss different approaches for computational methods for turbulent flows in Chapter 4. Then we end the thesis by a short conclusion.

For a complete discussion of the stabilized finite element method and error analysis we refer to the attached papers.

This work is a part of the development of a unified continuum mechanic's solver, Unicorn, which is based on an adaptive finite element method. The vision of *Unicorn* [21] is to develop one unified continuum mechanics solver including compressible and incompressible flows, fluid-structure and multi-phase problems. Unicorn is part of the *FEniCS applications* and based on components of the *FEniCS project* [10].

1.2 Conservation Laws

The *Initial Value Problem* (IVP) for the n-dimensional *conservation law* is defined as follows:

$$\begin{aligned} \dot{u} + \nabla \cdot f(u) = 0, \quad x \in \mathbb{R}^n, t > 0, \\ u(x,0) = u_0(x), \quad x \in \mathbb{R}^n, \end{aligned} \tag{1.1}$$

where $u = u(x,t) : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^m$ is a m - dimensional vector of conserved variables, $f(u) : \mathbb{R}^m \to \mathbb{R}^m$ is convectional flux, a given smooth function of u, and u_0 is initial data which is also given. The first term in the equation, $\dot{u}$, is a time derivative of the unknown solution u.

We assume that the system (1.1) is a *hyperbolic* system of partial differential equations, meaning that the flux Jacobian $f'(u)$ is diagonalizable and has real eigenvalues.

The *Boundary Value Problem* (BVP) is defined as follows:

$$\begin{aligned} \dot{u} + \nabla \cdot f(u) = 0, \quad x \in \Omega, t > 0, \\ u(x,0) = u_0(x), \quad x \in \Omega, \end{aligned} \tag{1.2}$$

where $\Omega \subset \mathbb{R}^n$ is a fixed open domain and Γ its boundary. The boundary conditions for the BVP (1.2) are based on the characteristics of the flux Jacobian system $f'(u)$. We will discuss the choice of boundary conditions later.

1.2.1 Weak Solution

Conservation laws allow for discontinuous solutions, e.g. in the form of shocks, which may be interpreted as weak solutions.

Assume that $\phi(x,t)$ is a smooth test function with compact support. Multiplying this test function to the conservation law (1.1) and integration by part gives

$$\int_{\mathbb{R}^n\times\mathbb{R}} (u\dot{\phi} + f(u)\cdot\nabla\phi)\mathrm{d}t\mathrm{d}x + \int_{\mathbb{R}^n} u(x,0)\phi(x,0)\mathrm{d}x = 0. \tag{1.3}$$

A function $u(x,t)$ is said to be a *weak solution* if it satisfies equation (1.3) for all smooth test functions with compact support.

1.2.2 Viscosity Solution

In general there is no sharp jump or sharp discontinuity in the physical problem. Instead there is a small diffusive layer over which the solution changes from one value to another. Therefore, we may replace the above conservation law with a regularized equation

$$\dot{u} + \nabla\cdot f(u) = \epsilon\Delta u, \tag{1.4}$$

where ϵ is a small viscosity. The limit when $\epsilon \to 0$ is referred to as a viscosity solution to the conservation law.

Analyzing equations with viscous regularization makes analysis easier in most cases. However, in a limit where ϵ goes to zero, problems may become more complicated and analysis difficult. In this paper we study *the compressible Euler equations*, that play a fundamental role in fluid dynamics describing *inviscid flow*. The Euler equations are obtained from *the Navier-Stokes equations*, when the physical viscosity in the system goes to zero.

For the Navier-Stokes equations, the existence and smoothness [1] of a strong (classical) solution is still not known. A French mathematician, Leray, showed the existence of weak solutions of the incompressible Navier-Stokes equations, but despite of that, the existence of weak solutions for the Euler equations is still unknown.

[1] This is one of the seven Millennium prizes which are stated by the Clay Mathematics Institute. The prize is US $1,000,000 for a solution or a counter-example.

1.2.3 The Riemann Problem

A conservation law together with a piecewise constant initial data, which has a single discontinuity, is called a *Riemann problem*. For the scalar conservation law it is defined as:

$$\dot{u} + \frac{\partial}{\partial x} f(u) = 0, \quad u(x,0) = \begin{cases} u_l, & x < 0, \\ u_r, & x \geq 0. \end{cases} \tag{1.5}$$

The initial data has a discontinuity point at $x = 0$ and depending on the values u_l and u_r the problem (1.5) has different solutions, such as s*hocks*, *rarefactions wave* and *contact discontinuities*.

1.2.4 Rankine Hugoniot Condition

If $u_l > u_r$, there is a unique weak solution to the Riemann problem (1.5), which is calculated as

$$u(x,t) = \begin{cases} u_l, & x < st, \\ u_r, & x \geq st. \end{cases} \tag{1.6}$$

where the constant s is called the *shock speed*. The discontinuity in the solution propagates with the shock speed.

Furthermore, the discontinuous function (1.6) is called a weak solution to the Riemann problem (1.5) if it satisfies the *Rankine-Huginiot condition*

$$f(u_l) - f(u_r) = s(u_l - u_r), \tag{1.7}$$

or for problem (1.5) we can write

$$s = \frac{f(u_l) - f(u_r)}{u_l - u_r}. \tag{1.8}$$

1.2.5 Example

To understand a family of solutions of the Riemann problem, consider the following conservation law

$$\dot{u} + \frac{\partial}{\partial x}\left(\frac{u^3}{3}\right) = 0, \quad u(x,0) = \begin{cases} 0, & x < 0, \\ 1, & 0 \leq x \leq 1, \\ 0, & x \geq 1. \end{cases} \tag{1.9}$$

For this problem the shock speed according the Rankine-Hugoniot condition is $s = \frac{1}{3}$. One can easily show that the *characteristics*, $\frac{dx}{dt} = f'(u(x,t))$,

for the equation (1.9) are straight lines, $x = \xi t + x_0$, $\xi = u_0(x)$, and a solution of (1.9) is constant along the characteristics with initial data $u(x, 0)$. Then the solution can be found as

$$u(x,t) = \begin{cases} 0, & x < 0, \\ \sqrt{\dfrac{x}{t}}, & 0 \le x < t, \\ 1, & t \le x < \dfrac{1}{3}t + 1, \\ 0, & x \ge \dfrac{1}{3}t + 1. \end{cases} \tag{1.10}$$

From Figure 1.1 one can see that the characteristic lines do not cross in $x \le 0$. Between $x > 0 \cup x < t$ a rarefaction wave develops in the solution. The characteristic lines from the region $t \le x \cup 0 < x < 1$ crosses the lines from the region $x \ge 1$. The crossed lines is called a *shock line* and solutions on this line develops with the shock speed $s = \frac{1}{3}$. Figure 1.2 describes the solution $u(x, t)$ at different times $t = 0, 0.5, 1, 1.5$.

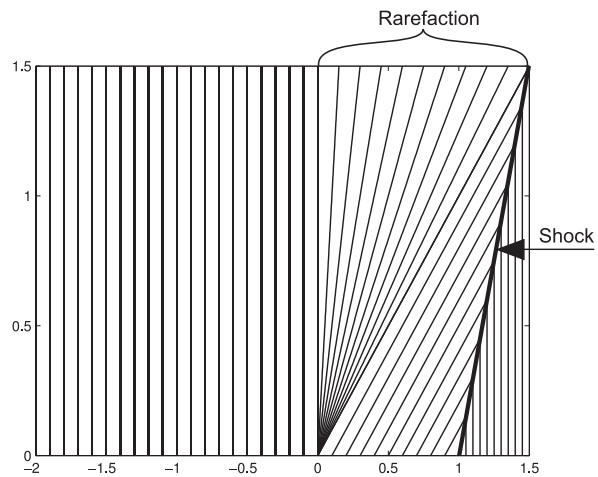

Figure 1.1. The characteristic lines for the equation (1.9).

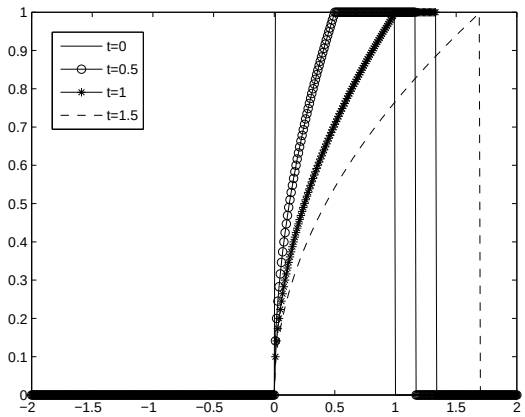

Figure 1.2. Solution of the equation (1.9) at for different times.

1.3 The Compressible Euler Equations

Now, we study one of the most important conservation laws in gas dynamics, the *compressible Euler equations*. The compressible Euler equations play a fundamental role in understanding and simulating compressible gases and fluids and are widely used in science and industry.

Similar to scalar conservation laws, as described above, the compressible Euler equations has a complex behavior, containing rarefaction and shock waves, jumps and contact discontinuities. To capture this complex behavior one need to design a numerical method carefully.

The compressible Euler equations express conservation of mass, momentum and total energy for an inviscid fluid enclosed in a fixed (open) domain Ω in three-dimensional space $\mathbb{R}^3$ with boundary Γ over a time interval $[0, \hat{t}\,]$ with initial time zero and final time $\hat{t}$.

In principle the conservation of mass states that the net mass flowing into some control volume must be equal to the rate of increase of total mass in the volume. The conservation of momentum comes directly from Newton's second law, which states that the time rate of change of momentum in a material region is equal to the sum of the forces on that region, or that the sum of forces is equal to mass times acceleration. And finally the conservation of energy states that the energy in an isolated system remains constant.

We seek the *density* ρ, *momentum* $m = \rho u$, with $u = (u_1, u_2, u_3)$ the *velocity*, and the *total energy* e as functions of $(x,t) \in \Omega \cup \Gamma \times [0, \hat{t}\,]$, where $x = (x_1, x_2, x_3)$ denotes the coordinates in $\mathbb{R}^3$ and u_i is the velocity in the x_i-direction. The Euler equations for $\hat{u} \equiv (\rho, m, e)$ read with $Q = \Omega \times I$ and $I = (0, \hat{t}\,]$:

$$\begin{aligned} \dot{\rho} + \nabla \cdot (\rho u) &= 0 \quad \text{in } Q, \\ \dot{m}_i + \nabla \cdot (m_i u) + p_{,i} &= f_i \quad \text{in } Q, \quad i = 1,2,3, \\ \dot{e} + \nabla \cdot (eu + pu) &= 0 \quad \text{in } Q, \\ u \cdot n &= 0 \quad \text{on } \Gamma \times I, \\ \hat{u}(\cdot, 0) &= \hat{u}^0 \quad \text{in } \Omega, \end{aligned} \tag{1.11}$$

where $p = p(x,t)$ is the *pressure* of the fluid, $p_{,i} = \partial p / \partial x_i$ is the partial derivative with respect to x_i, the dot indicates differentiation with respect to time, n denotes the outward unit normal to the boundary Γ and $f = (f_1, f_2, f_3)$ is a given *volume force* (like gravity) acting on the fluid, and $\hat{u}^0 = \hat{u}^0(x)$ represents initial conditions. Further, the total energy $e = k + \theta$, where $k = \rho |u|^2/2$ is the *kinetic energy*, with $|u|^2 \equiv u_1^2 + u_2^2 + u_3^2$, and $\theta = \rho T$ is the *internal energy* with T the *temperature* scaled so that $c_v = 1$, where c_v is the heat capacity under constant volume.

The boundary condition $u \cdot n = 0$ is a *slip boundary condition* requiring the normal velocity $u \cdot n$ to vanish corresponding to an impenetrable boundary with zero friction.

The number of unknowns including the pressure is six but there are only five equations in the Euler system (1.11), and so we close the system with the *state equation* of a *perfect gas*;

$$p = (\gamma - 1)\rho T, \tag{1.12}$$

expressing the pressure p as a function of density ρ and temperature T, where $\gamma = c_p$ is the *adiabatic index* with c_p the heat capacity under constant pressure, and $(\gamma - 1)$ is the *gas constant*.

For a perfect gas, the *speed of sound* c is given by $c^2 = \gamma(\gamma - 1)T$, and the *Mach number* is defined as $M = |u|/c$, with u the velocity of the gas.

Note that the incompressible Euler equations are obtained by assuming that the density is constant, where the conservation of mass or continuity equation becomes simply $\nabla \cdot u = 0$.

Chapter 2

Stabilized Finite Element Methods for Conservation Laws

2.1 Numerical Methods for Conservation Laws and the Compressible Euler Equations

In a review paper Woodward and Colella summarized three main approaches to the representation of discontinuities and shocks [47]: *artificial viscosity techniques*, *linear hybridization* and *Godunov's approach*.

Godunov's approach is based on a Riemann solver and solves the equations exactly at the interior discontinuity interfaces or approximates it using polynomials. In the original Godunov's scheme [11], the solution is approximated by piecewise constant functions, resulting in first order accuracy in space, together with the exact local Riemann solution at the cell interfaces. The piecewise constant functions can be replaced with higher order functions, leading to higher order schemes such as *Monotone Upstream-centered Schemes for Conservation Laws* (MUSCL). In general, it is hard to use this approach for complex geometries and unstructured grids. Obtaining higher order accurate schemes for unstructured meshes in 3D is not straightforward.

Linear hybridization methods are usually designed in the following way: The regions with a smooth solution is approximated with a high order scheme, while in non smooth regions the scheme is replaced by a lower order scheme which captures step gradients and discontinuities. Finite difference schemes are usually used within this framework. These methods are similar to the artificial viscosity approach, where a small amount of artificial viscosity is added to the scheme in regions near discontinuities. Both need to find the

right regions to give a special treatment.

Artificial viscosity was suggested by Neumann and Richtmyer [43] and has been much developed over the last decades. It regularizes conservation laws by adding extra terms and then modified equations are solved, see section 1.2.2. Large amount of viscosity can smeared out the important characteristics of the solution. For conservation laws, in particular the compressible Euler equations, one has to choose the artificial viscosity in such a way that it reduces oscillations near shocks and discontinuities, but is small in the smooth part of the domain. One well known approach is to locate shocks by *wavelet coefficients*, proposed by Harten [13]. Wavelet coefficients are used to locate the shock and discontinuity regions, where a small viscosity is then added.

Later in this chapter we discuss residual based streamline diffusion stabilization and shock capturing methods for finite elements.

2.2 An Overview of Numerical Methods

In this section we discuss some numerical techniques which are used for solving the compressible Euler equations. Among them, the well-known finite difference, finite volume and finite element methods.

2.2.1 Finite Difference and Finite Volume methods

Finite difference methods are popular with respect to efficiency for certain computational domains. If the computational domain is structured or equidistant, there is no need to keep information about the entire mesh. The solution is computed in each mesh point without solving a linear system of equations. This makes the program notably fast and very efficient. Finite difference methods are based on Taylor expansions.

Another method, which is similar to finite difference methods, is the *finite volume method*. The method is based on a volume integral of the partial differential equation where the integrals containing divergences are converted to surface integrals, which are approximated as a flux between volumes. For this method the computational mesh can be both structured and unstructured, and it allows for complex geometry. It is hard to construct a high order accurate scheme with this method, especially in the case of unstructured meshes.

2.2.2 Finite Element Methods

Finite element methods (FEM) are know for their *reliability* and *generality*. FEM is based on the Galerkin method with polynomial basis functions, where

the equations are written in *weak form*, by multiplying the equation to a *test function*. FEM is well suited for mathematical analysis, opening for a rigorous error analysis. This is very important feature, since approximating a solution does not guarantee that it is the correct solution.

FEM can be used in any geometry; it is easy to make a scheme very high order accurate by increasing the order of the basis function. *Discontinuous Galerkin methods* can be seen as a generalization of finite volume methods.

A disadvantage of the method is the computational efficiency. The method is typically relatively slow on a fixed mesh. However, rigorous error estimation and flexible adaptive mesh refinement can make the method remarkably efficient.

There are finite element methods, using non polynomial basis functions. The most popular one is the *spectral element method*, which is extensively used in fluid dynamics in Direct Numerical Simulations (DNS). Spectral element methods are combinations of spectral methods and finite element methods with special choices of basis functions. Despite of high orders of accuracy, the methods only works for relatively simple geometries.

2.3 A General Galerkin finite element method

The standard Galerkin finite element method is not stable for convection dominated problems. Instead a mesh-dependent consistent numerical stabilization is added. The development of stabilization methods started in the late 70's. Claes Johnson adopted the name *streamline diffusion methods* (SD) and with his coworkers published a series of papers on time dependent problems including advection-diffusion systems, and the Navier-Stokes equations.

The first paper with systematic analysis both mathematically and numerically by Johnson and Nävert appeared already in 1981 in the conference book [36]. Nävert defended his PhD dissertation in 1982 also on stabilization of finite element methods [42]. The work by Johnson, Hansbo and Szepessy [40, 35, 38, 39, 12], was mainly focused on stability properties, and accuracy of SD for conservation laws including the compressible Euler equations, and the incompressible Navier-Stokes.

In parallel, Hughes and his co-workers were doing similar research on stabilization of finite element methods. Hughes together with Brooks published the journal paper [5], which summarized the existing stabilized methods. Here they introduced the Streamline Upwind Petrov-Galerkin (SUPG) methods in 1979, see [28], and later Hughes and Tezduyar and their co-workers published a number of papers in this subject, see [29, 46]. For more details about stabi-

lized methods for compressible flow we refer to [30].

Here we introduce a General Galerkin or cG(1)cG(1) finite element method which we refer to as G2. G2 is a simplified stabilized finite element method with individual stabilization of each equation for conservation of mass, momentum and energy, with streamline diffusion least squares stabilization of only the convection term of the corresponding residual, and with componentwise residual based shock-capturing stabilization for each equation. This makes the computation of the stabilization terms cheaper compared to other methods, since there are fewer terms.

2.3.1 cG(1)cG(1) for the Euler Equations

We start by introducing the following notation: The scalar product over the finite element mesh $\mathcal{T}_h$ is defined as

$$(v, w) = \sum_{K \in \mathcal{T}_h} \int_K v \cdot w \, dx.$$

Let $0 = t_0 < t_1 < ... < t_N = \hat{t}$ be a sequence of discrete time steps with time intervals $I_n = (t_{n-1}, t_n]$ and timestep $\Delta t = t_n - t_{n-1}$. Let $\hat{u}_h = (\rho_h, m_h, e_h)$ be continuous piecewise linear in space and time. The time derivatives of density, momentum and energy, and the mean values in the interval I_n are denoted by

$$\dot{\rho}_n = \frac{\rho_n - \rho_{n-1}}{\Delta t_n}, \quad \dot{m}_n = \frac{m_n - m_{n-1}}{\Delta t_n}, \quad \dot{e}_n = \frac{e_n - e_{n-1}}{\Delta t_n},$$

$$\bar{\rho}_n = \frac{1}{2}(\rho_n + \rho_{n-1}), \quad \bar{m}_n = \frac{1}{2}(m_n + m_{n-1}), \quad \bar{e}_n = \frac{1}{2}(e_n + e_{n-1}),$$

where $\rho_n = \rho_h(t_n)$, $p_n = p_h(t_n)$, $m_n = m_h(t_n)$ with $m_n = m_{n_j}$, $m_h = m_{h_j}$ for $j = 1, 2, 3$.

Let $W_h \subset H^1(\Omega)$ be a finite element space consisting of continuous piecewise linear functions on a fixed mesh $\mathcal{T}_h = \{K\}$ of mesh size $h(x) < 1$, with elements K. Here $H^1(\Omega)$ denotes the standard Hilbert space of functions that are square integrable together with their first order derivatives. Let $W_{h,0}$ be the space of vector functions in W_h^3 satisfying the slip boundary condition, that is $W_{h,0} = \{w \in W_h^3 : w \cdot n = 0 \text{ on } \Gamma\}$.

The cG(1)cG(1) method for the compressible Euler equations, here without source terms for simplicity, reads: For $n = 1, ..., N$, find $\hat{u}_h^n = (\rho_n, m_n, e_n) \equiv$

$(\rho_h(t_n), m_h(t_n), e_h(t_n))$ with $\hat{u}_h^n \in V_h \equiv W_h \times W_{h,0} \times W_h$, such that

$$\begin{aligned} R_\rho(\rho_h; v_n^\rho) + SD_\rho(\rho_h; v_n^\rho) &= 0, && (2.1)\\ R_m(m_h; v_n^m) + SD_m(m_h; v_n^m) &= 0, && (2.2)\\ R_e(e_h, v_n^e) + SD_e(e_h; v_n^e) &= 0, && (2.3) \end{aligned}$$

for all test functions $\hat{v}_h^n = (v_n^\rho, v_n^m, v_n^e) \in V_h \equiv W_h \times W_{h,0} \times W_h$, where $SD_\rho(\rho_h; v_n^\rho)$, $SD_m(m_h; v_n^m)$ and $SD_e(e_h; v_n^e)$ are stabilization terms, defined by

$$\begin{aligned} SD_\rho(\rho_h; v_n^\rho) &= (\delta u_n \cdot \nabla \bar{\rho}_n, u_n \cdot \nabla v_n^\rho) + (\hat{\nu}_\rho \nabla \bar{\rho}_n, \nabla v_n^\rho), && (2.4)\\ SD_m(m_h; v_n^m) &= (\delta u_n \cdot \nabla \bar{m}_n, u_n \cdot \nabla v_n^m) + (\hat{\nu}_m \nabla \bar{m}_n, \nabla v_n^m), && (2.5)\\ SD_e(e_h; v_n^e) &= (\delta u_n \cdot \nabla \bar{e}_n, u_n \cdot \nabla v_n^e) + (\hat{\nu}_e \nabla \bar{e}_n, \nabla v_n^e), && (2.6) \end{aligned}$$

and

$$\begin{aligned} R_\rho(\rho_h; v_n^\rho) &= (\dot{\rho}_n, v_n^\rho) - (u_n \bar{\rho}_n, \nabla v_n^\rho), && (2.7)\\ R_{m_j}(m_{h_j}; v_n^{m_j}) &= (\dot{m}_{n_j}, v_n^{m_j}) - (u_n \bar{m}_{n_j}, \nabla v_n^{m_j}) - (p_n, \nabla \cdot v_n^{m_j}), && (2.8)\\ R_e(e_h; v_n^e) &= (\dot{e}_n, v_n^e) - (u_n \bar{e}_n, \nabla v_n^e) + (\nabla \cdot (u_n p_n), v_n^e), && (2.9) \end{aligned}$$

where we let u_n, p_n and T_n be finite element functions in $W_{h,0}$, W_h and W_h respectively, which are defined by their nodal values given by

$$\begin{aligned} u_n(N_i) &= \bar{m}_n(N_i)/\bar{\rho}_n(N_i),\\ p_n(N_i) &= (\gamma - 1)\bar{\rho}_n(N_i) T_n(N_i),\\ T_n(N_i) &= \bar{e}_n(N_i)/\bar{\rho}_n(N_i) - |u_n(N_i)|^2/2 \end{aligned}$$

for all nodes N_i in the mesh $\mathcal{T}_h$.

We use the stabilization parameters $\delta = C_\delta(\Delta t_n^{-2} + |u_n|^2 h^{-2})^{-1/2}$ and $\hat{\nu}_\alpha = \max\left(C_\alpha |R_\alpha(\hat{u})| h^2, C_h h^{3/2}\right)$ $(\alpha = \rho, m_j, e)$, with constants C_δ, C_α and $C_h \sim \frac{U}{\sqrt{L}}$, where U and L are characteristic velocity and length scales, and we define the normalized strong residuals by

$$\begin{aligned} R_\rho(\hat{u}_h) &= (\dot{\rho}_n + \nabla \cdot (\bar{\rho}_n u_n))/|\bar{\rho}_n|, && (2.10)\\ R_{m_j}(\hat{u}_h) &= ((\dot{m}_n)_j + \nabla \cdot ((\bar{m}_n)_j u_n) + \nabla p_n)/(|(\bar{m}_n)_j| + \epsilon), \quad j = 1, 2, 3, && (2.11)\\ R_e(\hat{u}_h) &= (\dot{e}_n + \nabla \cdot (\bar{e}_n u_n + p_n u_n))/|\bar{e}_n|, && (2.12) \end{aligned}$$

for $t \in I_n$, and with $\epsilon > 0$ a small safety factor. The time step Δt_n is given by a CFL-condition, with typically $\Delta t_n \sim \min_{E_j \in \mathcal{T}_h} (h/|u_n|)_{E_j}$, for all elements E_j in the mesh $\mathcal{T}_h$.

We note that we here stabilize each equation of (1.11) individually, rather than stabilizing the complete system (1.11) using a stabilization matrix, which is a common approach, see e.g. [46]. Our approach is cheap and simple to implement, nevertheless it captures the important phenomena of compressible flow.

2.3.2 Numerical Examples

For complete analysis and numerical studies for well-known benchmarks, we refer to Paper I in this thesis.

To illustrate the G2 method for the compressible Euler equations we present the following 2D implosion benchmark problem. This benchmark problem was presented in [31] and was studied using a number of finite difference schemes in [44]. The problem is the following: a box in 2D of size $(x, y) \in [-0.3, 0.3] \times [-0.3, 0.3]$ has a small rhomb inside of size $(x, y) \in [\pm 0.15, 0] \times [0, \pm 0.15]$, see figure 2.1. Gas with low density and pressure $(\rho_{inside}, p_{inside}) = (0.125, 0.14)$, with higher density and pressure gas outside of the rhomb, $(\rho_{outside}, p_{outside}) = (1, 1)$. The initial velocity is zero, and the gas constant is $\gamma = 1.4$. Reflecting boundary conditions are used for all boundaries, corresponding to slip boundary conditions in G2.

Following [44] we also did the computation in the upper right quadrant, $(x, y) \in [0, 0.3] \times [0, 0.3]$, with a mesh of 400×400 vertices.

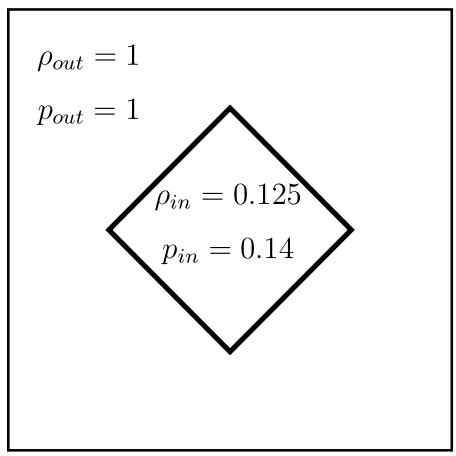

Figure 2.1. The initial data for implosion problem.

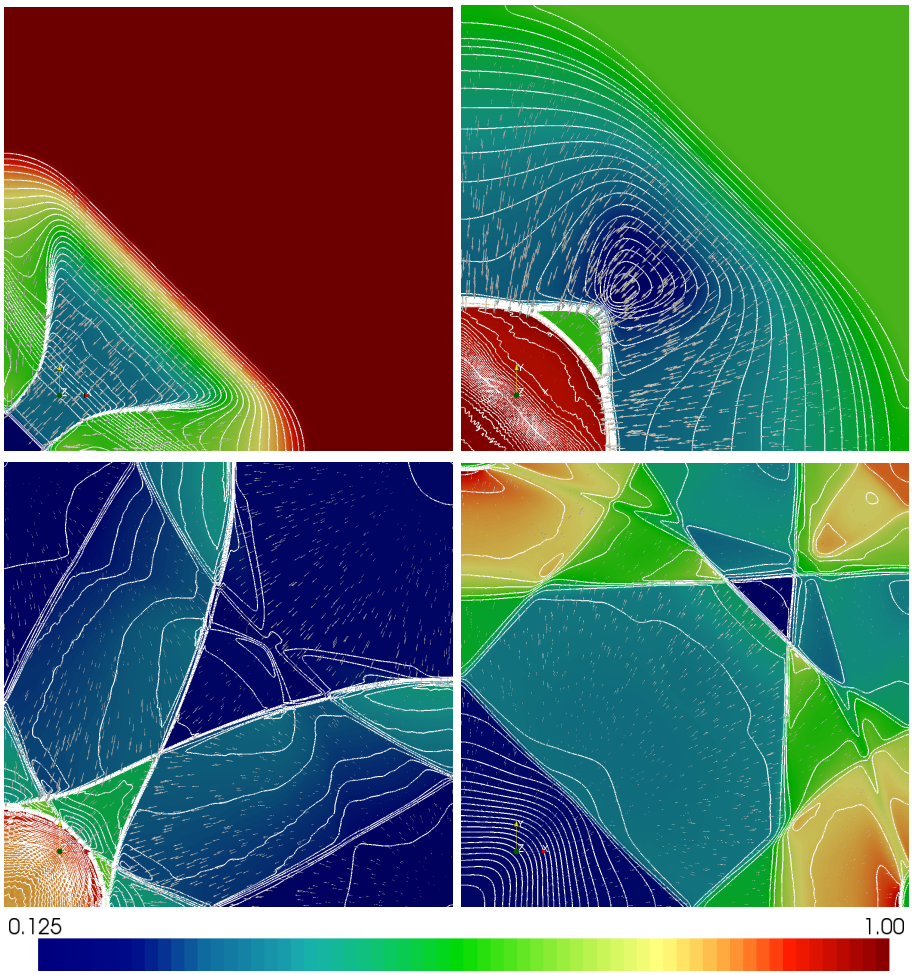

Figure 2.2. Result for the implosion problem at different times: In all figures we plot pressure color bars for pressure, velocity arrows and density contours: upper-left: $t = 0.025$, 36 density contours from the interval $[0.124923, 1.00005]$; upper-right: $t = 0.075$, 36 density contours from the interval $[0.563953, 1.26687]$; lower-left: $t = 0.7$, 36 density contours from the interval $[0.602772, 1.19334]$; lower-right: $t = 1.4$, 36 density contours from the interval $[0.43217, 1.05981]$.

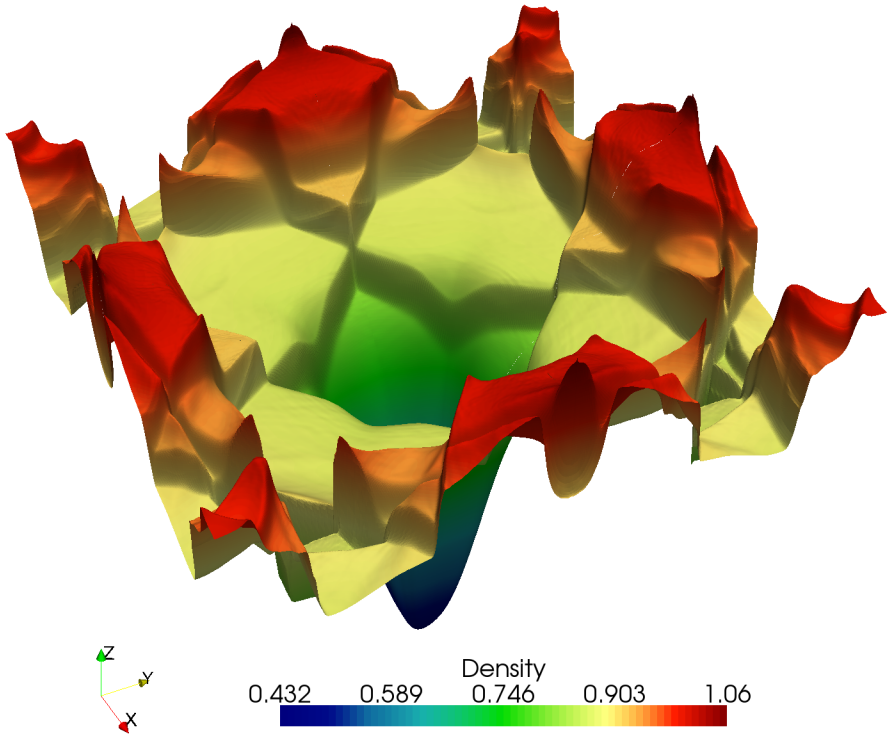

Figure 2.3. Density at time $t = 1.4$, on the square $(x, y) \in [-0.3, 0.3] \times [0.3, 0.3]$

At time $t = 0$ the internal "walls" of the rhomb are removed. Since the density and pressure are low inside the small rhomb, gas starts to move from the rest of domain to the rhomb, resulting in shock, rarefaction and contact discontinuity waves. Consequently, these waves cross each other and reflect when they hit the boundary.

High order finite difference schemes are used in [44], and the presented solutions appear to be sharper than the G2 solution defined above. However, this is compensated for by using adaptive mesh refinement, which is illustrated with numerical results in the following chapters.

Chapter 3

A posteriori error estimation and adaptive finite element methods

As mentioned in the above chapters, a main feature of finite element methods is a rigorous framework for a posteriori error estimation.

A posteriori error estimation is based on the computed finite element solution, while *a priori error estimation* is based on the exact solution.

In this chapter we give a brief overview of error estimation in the finite element framework.

3.1 Error analysis for finite element methods

Let $V_h \subset H^1(\Omega)$ be a finite element space, $H^1(\Omega)$ be the Hilbert space with function in $L_2(\Omega)$ - the space of Lebesgue square integrable functions, with also first order derivatives in $L_2(\Omega)$ and u, U be exact and approximate solutions. An a priori error estimate shows that for a class of problems a finite element method produces the best approximation U of the solution u in the finite element space V_h, see e.g. [9, 32].

In a posteriori error analysis, only knowledge of the computed solution is required. This means that the solution is computed, and then the error is estimated using this solution. Once the bound of the error is estimated in the appropriate norm, it is possible to improve the approximation in order to reduce the error which makes the method efficient and reliable. The idea of using duality arguments in a posteriori error estimation was first studied by Babuška and Miller for elliptic model problems [1, 2]. A more general framework for a systematic approach to a posteriori error estimation was

developed by Eriksson and Johnson, Becker and Rannacher, with co-workers, [9, 7, 8, 3, 4].

In most applications we are interested in obtaining an estimate of the error in some *quantity of interest* or *output*, rather than estimating the error itself. This approach minimizes a region in the computational domain, which must be refined to obtain the accuracy in the quantity of interest. The quantity of interest can be fluid forces such as drag and lift, stresses and fluxes. For more information, see e.g. [32].

3.2 A posteriori error estimation for the compressible Euler equations

For systems of conservation laws, a posteriori error analysis was earlier investigated by Johnson, Szepessy and Hansbo [37, 39, 33, 34], and stationary compressible Euler equations in 2D were studied by Hartmann, Huston and Süli, and Larson and Barth using adaptive discontinuous Galerkin methods [14, 26, 27, 41]. Burman studied a posteriori error estimations for the time dependent compressible Euler equations in 2D [6].

In this thesis we present a posteriori error estimates for the compressible Euler equations in three space dimensions, where we formulate the Euler equations (1.11) in terms of density, velocity and pressure. These variables are independent of each other, resulting in a simplified derivation of the a posteriori error estimate. For details we refer to the second paper of the thesis, where we show the derivation of the dual problem for the compressible Euler equations and we prove an a posteriori error estimates.

To understand the essence of a posteriori error analysis for nonlinear problems, including the compressible Euler equations, consider the following nonlinear equation in a finite n-dimensional space:

$$f(u) = b, \tag{3.1}$$

where $f : \mathbb{R}^n \mapsto \mathbb{R}^n$, $u \in \mathbb{R}^n$.

3.2.1 Linearization

The *residual* of a computed approximate solution $U \in \mathbb{R}^n$ is defined as

$$R(U) = f(U) - b, \tag{3.2}$$

or we can write

$$-R(U) = f(u) - f(U). \tag{3.3}$$

To derive a relation between the residual $R(U)$ and the error $e = u - U$,we insert $u = U + e$ in (3.2) and Taylor expand about U to get

$$\begin{aligned} f(U+e) &= b \quad \Rightarrow \\ f(U) + f'(U)e + \mathcal{O}(e^2) &= b \quad \Rightarrow \\ f'(U)e + \mathcal{O}(e^2) &= -(f(U) - b) \quad \Rightarrow \\ f'(U)e + \mathcal{O}(e^2) &= -R(U) \end{aligned} \tag{3.4}$$

where $f'(U)$ is a Jacobian matrix. We drop the remaining terms which is in order $\mathcal{O}(e^2)$, to write the relation between residual and error as

$$f'(U)e \approx -R(U). \tag{3.5}$$

3.2.2 Adjoint Problem

The adjoint problem is used to determine the effect on a quantity of interest of the accumulation of errors.

With the notation $A(e) = f'(U)e$ we use the definition: an operator A^* is adjoint to the linear operator A if

$$(A(e), \phi) = (e, A^*(\phi)), \quad \forall e, \phi \in \mathbb{R}^n. \tag{3.6}$$

The function ϕ in the last equality is also called a *Generalized Green's function*, see [32]. Using the adjoint operator A^*, we define the following *dual problem* for the variable ϕ: find $\phi \in \mathbb{R}^n$ such that

$$A^*(\phi) = \psi, \tag{3.7}$$

where $\psi \in \mathbb{R}^n$ defines *a quantity of interest* $M(u)$ of the problem by $M(u) = (u, \psi) = u \cdot \psi$. The quantity $M(u)$ can be an error at some point or the error in an average over some subset of the computational domain or some norm of the error. In this thesis, we choose the drag force of the compressible flow around an object as the quantity of interest.

3.2.3 Error Estimation

The following lemma is an error representation formula:

Lemma 3.2.1. *The error $M(u) - M(U) = M(e)$ has the following relation to the residual $R(U)$, and dual solution ϕ*

$$M(e) \approx (-R(U), \phi). \tag{3.8}$$

Proof. Using the definition of the adjoint operator, (3.6), and the dual problem (3.7), we have the following error representation for the quantity of interest:

$$M(u) - M(U) = M(e) = (e, \psi) = (e, A^*(\phi)) \approx (-R(U), \phi). \tag{3.9}$$

□

Now, assume $\hat{U} = (\rho_h, u_{h_i}, p_h)$ is a G2 solution and $\hat{u} = (\rho, u_i, p)$ is an exact solution, and $\hat{\phi} = (\phi_\rho, \phi_{u_i}, \phi_p)$ is the exact solution of the linearized dual problem of the Euler equations.

Then the following theorem is *an a posteriori error estimate for the compressible Euler equations.*

Theorem 3.2.2. *The error, $M(\hat{u}) - M(\hat{U})$ satisfies the following inequality*

$$|M(\hat{u}) - M(\hat{U})| \leq \sum_{n=1}^{N} \Delta t \max_{I_n} \left[C_n h \sum_{K \in \mathcal{T}_n} |D\hat{\phi}|_K |R(\hat{U})|_K + \sum_{K \in \mathcal{T}_n} |SD(\hat{U}; \pi_h \hat{\phi})| \right] \tag{3.10}$$

where Δt - time stepsize, h - meshsize, $\mathcal{T}_h = \{K\}$ - a fixed mesh with mesh size $h(x) < 1$ and elements K, $SD(\hat{U}; \pi_h \phi)$ - stabilization terms in the G2 method.

The a posteriori error estimate is expressed in terms of computed solutions of the primal and dual problems.

3.2.4 An adaptive algorithm

Once the error estimate is obtained, the next step is to use the error indicator (3.10) to improve the numerical approximation. In the simplest *adaptive algorithms* the mesh, $\mathcal{T}_h$, remains constant until the final time t. For the fixed mesh, the primal problem is solved forward and the dual problem is solved backward in time. Then the mesh is refined/coarsened according to the error indicator. Then the whole cycle is repeated until some criteria is satisfied:

Algorithm 3.2.3. *Given some tolerance* TOL *and initial coarse mesh $\mathcal{T}_h^0$. Starting with time $k = 0$ do the following adaptive loop:*

1. Compute the primal solution of the compressible Euler equations, $\hat{u}_h \in V_h$, on the current mesh $\mathcal{T}_h^k$.

2. Compute the dual solution, $\hat{\phi}_h \in V_h$, on the same mesh.

3. Compute the error indicator defined in (3.10), if $|M(u) - M(U)| <$ TOL*, then* STOP*.*

4. Refine elements in $\mathcal{T}_h^k$ with the largest error indicator to get mesh $\mathcal{T}_h^{k+1}$.
5. Set $k = k + 1$ and go to 1.

3.2.5 Numerical Examples

To illustrate the above adaptive algorithm, we consider compressible supersonic flow, for $M = 1.4$ around a cylinder, with diameter $d = 0.0254$, in two dimensions. In the simulation we use dimensionless variables: Let L be a length, then the density ρ^*, sound speed c^* and the temperature T^* are normalized by their free stream values, $\rho^*_\infty, c^*_\infty, T^*_\infty$. The time t is normalized by L/c^*_∞, pressure by $p = \frac{p^*}{\rho^*_\infty c^2_\infty}$ and velocity by $u = \frac{u^*}{c_\infty}$.

For the inlet all variables of the solution are given, at the outlet boundary conditions based on characteristics are used, and a slip boundary condition is used in the rest of the computational domain. Since the characteristics of the Euler equation goes outward at the outlet, homogeneous Neumann or "do nothing" boundary condition is applied at the outlet. The quantity of interest in this example is a drag force acting on the cylinder.

The problem is solved forward in time until a final time $t = 1$, and then the corresponding dual solution is computed backward in time. The mesh is refined according to the error indicator described above, and then the same steps are repeated.

We plot the solution from the finest mesh which is obtained after seven adaptive refinements. Figure 3.1 shows the solution of the dual problem at the finest mesh at time $t = 0$ together with the sonic contour $M = 1$ at time $t = 1$. Note, that this is not a stationary solution to the problem at this time. The shock waves develop along the sonic contour. One see that the dual solution indicates the upstream region, close to the supersonic region, and the downstream region where the wake will develop, to be important for the drag value. However, the residual of the primal equations is not large close to the inlet, it is usually large along the shock waves and wake. From the dual solution and the primal residual the error indicator is computed in each cell, K, of the triangulation $\mathcal{T}_h$. Figure 3.3 shows the finest mesh together with the $M = 1$ contour. We can see that the adaptive refinement does not follow only the shock region, it refines also the regions where the dual solution is large.

Finally, Figure 3.3 we plot the error indicator vs the number of cells. It shows, that the adaptive algorithm works well and alsready after seven iterations the error in drag smaller than TOL$= 10^{-3}$.

we plot the L_2-norm of the error indicator versus the number of cells. It shows, that the adaptive algorithm works well and already after seven

iterations the L_2-norm of the relative error to the value of drag gets smaller than $\mathrm{TOL} = 10^{-3}$.

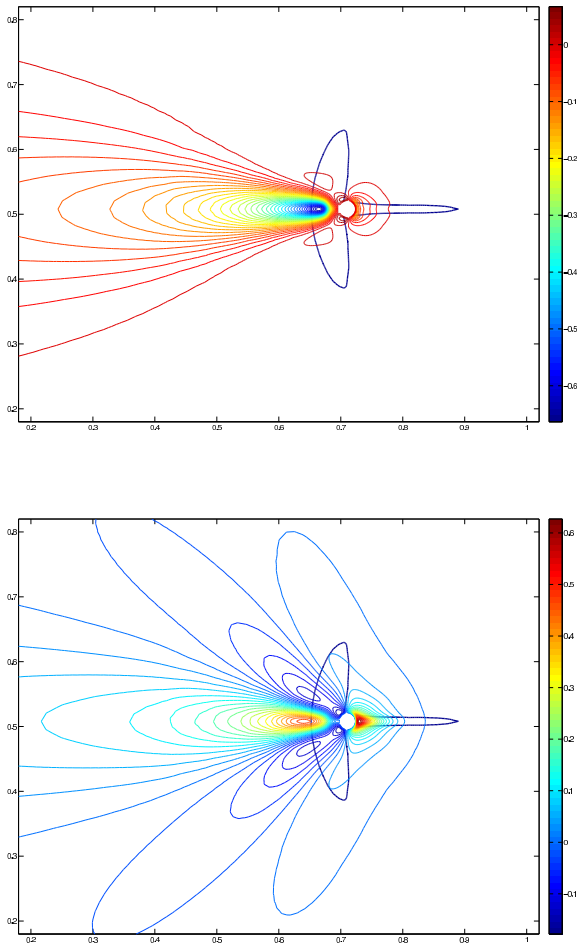

Figure 3.1. The dual density (upper) and pressure (lower) at time $t = 1$ after seven adaptive refinements with respect to drag force together with the $M = 1$ contour.

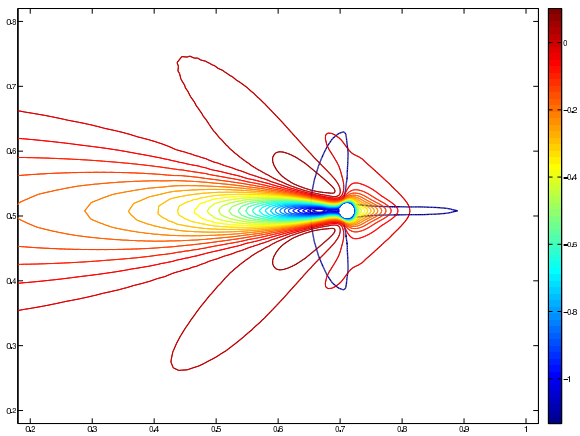

Figure 3.2. The dual velocity at time $t = 1$ after seven adaptive refinements with respect to drag force together with $M = 1$ contour.

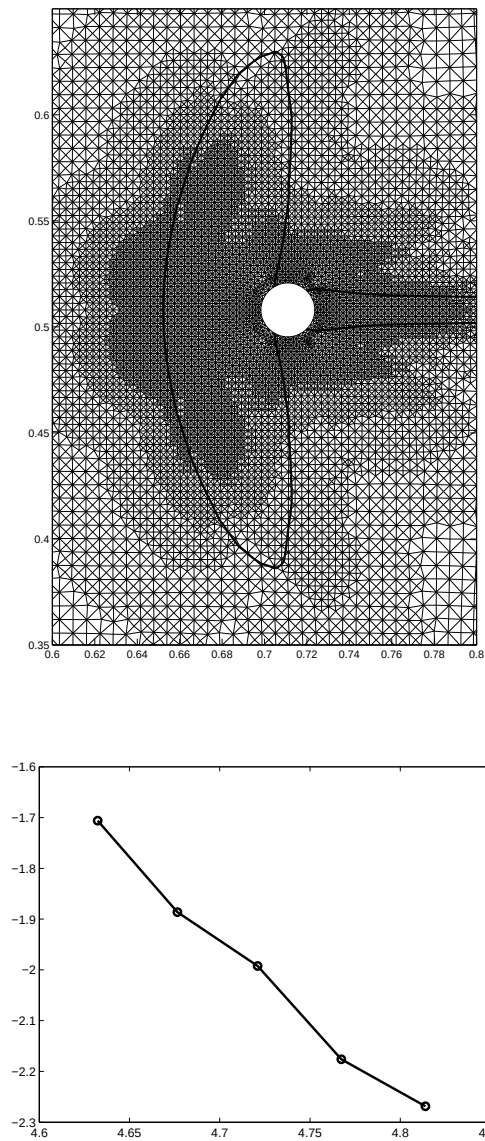

Figure 3.3. The mesh (*above*) after seven adaptive refinement with 53046 cells and 26807 nodes, and plot of $\log_{10}(\#cells)$ vs $\log_{10}(error\ indicator)$ (*below*).

Chapter 4

Adaptive DNS/LES turbulence modeling

As is mentioned above, when the viscosity of the flow gets very small, the flow becomes turbulent. In fluid mechanics the transition to turbulence from laminar flow is characterized by the *Reynolds number*, Re. The Reynolds number is a dimensionless parameter which is calculated as $Re = \frac{LU}{\nu}$, where L is a characteristic length, U is velocity and ν is the viscosity of the fluid. For small Reynolds numbers the Navier-Stokes equations can be solved, but for high Reynolds number when the flow is turbulent full computational resolution is impossible. The flow usually becomes turbulent for Reynolds numbers $Re \sim 10^3$.

4.1 Turbulence modeling for flow problems

Numerical methods for computing turbulent flows can be divided into three main approaches: ***D**irect **N**umerical **S**imulation* (*DNS*), ***R**eynolds **A**veraged **N**avier **S**tokes* (*RANS*) and ***L**arge **E**ddy **S**imulations* (*LES*). Here we give a short overview of these approaches.

DNS uses a numerical method to solve the Euler/Navier-Stokes equation directly by resolving all scales in space and time. For industrial applications such as aerodynamics, the Reynolds number of the fluid is of order 10^6 and larger, where full resolution of the small scales need more than $Re^3 = 10^{18}$ grid points, which makes the computation impossible with the computer power we have available today. Therefore, the traditional approach to get around DNS is to use turbulence modeling. For that the original equations are replaced by modified equations, which are obtained by averaging or filtering the original equations.

In RANS the Navier-Stokes equations are replaced by time averaged flow equations, where the filter corresponds to a global average. In LES the average is local in space and time. LES needs a more refined mesh than RANS, but much coarser than DNS. By the LES method one can resolve and predict instantaneous turbulent flow structures. For more information about this method we refer to a book by Sagault [45].

4.1.1 Adaptive DNS/LES

The G2 method described above can be an alternative approach to treat turbulence. With respect to a quantity of interest, G2 chooses the mesh automatically based on a posteriori error estimation and resolves the flow features which has large influence on the quantity of interest while other scales remain unresolved with the stabilization acting as a numerical turbulence model. Therefore, adaptive G2 can be characterized as a DNS/LES method, since a part of the flow is resolved according to the quantity of interest as DNS and the rest of the flow is left unresolved in a LES. To compute mean value output, G2 may e.g. use LES in the wake and DNS to capture separation and shocks. Rigorous error estimation in G2 significantly minimizes the computational cost of the simulation, for turbulent flow. The dual problem shows sensitivity with respect to the output which is taken into account automatically in the algorithm.

Adaptive DNS/LES was developed and successfully applied to incompressible flow by Hoffman and Johnson [15, 20, 24, 16, 22, 19, 17, 18, 23]. For a detailed analysis of G2 we refer to [25].

4.2 Adaptive DNS/LES for compressible Euler equations

In this thesis we study the extension of Adaptive DNS/LES to turbulent compressible flow. For low Mach number, the flow is characterized as incompressible since the effect of compressibility is negligible. However, when the Mach number increases the flow gets very complex: including turbulent wakes, bow shocks, rarefaction waves, discontinuities, shock wake interaction, expansion and compression of supersonics regions, and transition to subsonic/supersonic flow. To resolve all these features by the numerical method is impossible for inviscid or slightly viscous flows. We can see in the above presented results that G2 focus computational resources to regions that have high influence on the quantity of the interest in the computation, thereby minimizing the computational cost.

The detailed discussion of the G2 method for compressible flow can be found in the second paper of this thesis. This paper is the first step towards extending Adaptive DNS/LES to turbulent compressible flow.

Overall, G2 may open new possibilities for computational compressible turbulence.

Chapter 5

Conclusion and future direction

In this thesis we have presented a stabilized cG(1)cG(1) finite element method for the compressible Euler equations with componentwise least squares stabilization and residual based shock capturing. This method is robust, easy to implement and conserves all quantities of the conservation law, see Paper I.

Later we extended the Adaptive DNS/LES method to compressible flow, by proving duality based a posteriori error estimates, and presenting numerical results, see Paper II.

As a future direction, we are going to extend the a posteriori error analysis including Adaptive DNS/LES to viscous compressible flow and to multiphase flow, and also study the method for problems of compressible turbulence.

Chapter 6

Summary of Papers

6.1 Paper I: A General Galerkin Finite Element Method for the Compressible Euler Equations

In this paper we present a General Galerkin method for the compressible Euler equations. The General Galerkin method is a stabilized finite element method based on linear approximation in space and time. The stabilization is componentwise in the form of a weighted least squares stabilization of the convection terms together with residual based shock-capturing, resulting in a symmetric stabilization matrix in the discrete system. The G2 method presented here is a generalization of the cG(1)cG(1) method for incompressible flow [25].

The method is cheap and simple to implement and conserves total mass, momentum and energy exactly. We prove an a posteriori energy estimate for the method, and we illustrate the performance of the method in a number of numerical tests in 2d and 3d.

6.2 Paper II: An Adaptive Finite Element Method for the Compressible Euler Equations

In this paper we present an adaptive General Galerkin method for the compressible Euler equations. The mesh adaptation is based on a dual problem to the Euler equations. We give a derivation of the linearized dual problem which is solved numerically by a cG(1)cG(1) method. In this paper we prove an a posteriori error estimate for the compressible Euler equations and we give an adaptive algorithm based on this error estimate.

The primal problem is solved by the G2 method described in Paper I, and the dual problem is solved with the same approach but with a simplified stabilization. We give a number of computational results to illustrate the method.

Bibliography

[1] I. Babuška, *Feedback, adaptivity and a posteriori estimates in finite elements: Aims, theory, and experience*, Accuracy, Estimates and Adaptive Refinements in Finite Element Computation (I. Babuška, O. C. Zienkiewicz, J. Gago, and E. R. de A. Oliveira eds.), Wiley, New York, 1986.

[2] I. Babuška and A. D. Miller, *The post-processing approach in the finite element method, i: Calculation of displacements, stresses and other higher derivatives of the dispacements.*, Int. J. Numer. Meth. Eng., 20 (1984), pp. 1085–1109.

[3] R. Becker and R. Rannacher, *A feed-back approach to error control in adaptive finite element methods: Basic analysis and examples*, East-West J. Numer. Math., 4 (1996), pp. 237–264.

[4] ——, *A posteriori error estimation in finite element methods*, Acta Numer., 10 (2001), pp. 1–103.

[5] A. Brooks and T. Hughes, *Streamline upwind/petrov- galerkin formulations for convection dominated flows with particular emphasis on the incompressible navier-stokes equations*, 32 (1982), pp. 199–259.

[6] E. Burman, *Adaptive finite element methods for compressible flow*, Computer methods in applied mechanics and engineering, 190 (2000), pp. 1137–1162.

[7] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson, *Introduction to adaptive methods for differential equations*, Acta Numer., 4 (1995), pp. 105–158.

[8] ——, *Computational Differential Equations*, Cambridge University Press New York, 1996.

[9] K. Eriksson and C. Johnson, *An adaptive finite element method for linear elliptic problems*, Math. Comp., 50 (1988), pp. 361–383.

[10] FEniCS, *Fenics project*, `http://www.fenics.org`, (2003).

[11] S. K. Godunov, *A difference scheme for numerical solution of discontinuous solution of hydrodynamic equations*, vol. 47, Math. Sbornik (1959), translated US Joint Publ. Res. Service, JPRS 7226, 1969, pp. 271–306.

[12] P. Hansbo, *Explicit streamline diffusion finite element methods for the compressible euler equations in conservation variables.*, Journal of Computational Physics, 109 (1993), pp. 274–288.

[13] A. Harten, *Adaptive multiresolution schemes for shock computations*, J. Comput. Phys, 115 (1994), pp. 319–338.

[14] R. Hartmann, *Adaptive FE Methods for Conservation Equations*, in Hyperbolic Problems: theory, numerics, applications: eighth international conference in Magdeburg, February, March 2000, H. Freistühler and G. Warnecke, eds., vol. 141 of International series of numerical mathematics, Birkhäuser, Basel, 2001, pp. 495–503.

[15] J. Hoffman, *On duality based a posteriori error estimation in various norms and linear functionals for les*, SIAM J. Sci. Comput., 26(1) (2004), pp. 178–195.

[16] ——, *Computation of mean drag for bluff body problems using adaptive dns/les*, SIAM J. Sci. Comput., 27(1) (2005), pp. 184–207.

[17] ——, *Adaptive simulation of the sub-critical flow past a sphere*, J. Fluid Mech., 568 (2006), pp. 77–88.

[18] ——, *Computation of turbulent flow past bluff bodies using adaptive general galerkin methods: drag crisis and turbulent euler solutions*, Comp. Mech., 38 (2006), pp. 390–402.

[19] ——, *Adaptive turbulence computation based on weak solutions and weak uniqueness, quality and reliability of large-eddy simulations*, Eds. J. Meyers, B.J. Geurts and P. Sagaut, (2008), pp. 21–35.

[20] ——, *Efficient computation of mean drag for the subcritical flow past a circular cylinder using general galerkin g2*, International Journal of Numerical Methods in Fluids, 59(11) (2009), pp. 1241–1258.

[21] J. Hoffman, J. Jansson, M. Nazarov, and M. Stöckli., *Unicorn project*, `http://www.fenics.org/wiki/Unicorn`, (2008).

[22] J. Hoffman and C. Johnson, *Adaptive DNS/LES: a New Agenda in CFD*, Finite Element Methods: 1970s and beyond, CIMNE, 2004.

[23] ——, *A new approach to computational turbulence modeling*, Compit. Meth. Appl. Engrg, 195 (2006), pp. 2865–2880.

[24] ——, *Stability of the dual navier-stokes equations and efficient computation of mean output in turbulent flow using adaptive dns/les*, Comput. Methods Appl. Mech. Engrg., 195 (2006), pp. 1709–1721.

[25] J. Hoffman and C. Johnson, *Computational Turbulent Incompressible Flow*, Springer, 2007.

[26] P. Houston and R. Hartmann, *Goal–oriented a posteriori error estimation for compressible fluid flows*, in Numerical Mathematics and Advanced Applications, F. Brezzi, A. Buffa, S. Corsaro, and A. Murli, eds., Springer, 2003, pp. 775–784.

[27] P. Houston, R. Hartmann, and A. Süli, *Adaptive discontinuous Galerkin finite element methods for compressible fluid flows.*, in Numerical methods for Fluid Dynamics VII, ICFD, M. Baines, ed., 2001, pp. 347–353.

[28] T. Hughes and A. Brooks, *A multi-dimensional upwind scheme with no crosswind diffusion, in: Finite element methods for convection dominated flows*, T.J.R. Hughes (ed.), ASME Monograph AMD-34, 1979, pp. 19–35.

[29] T. Hughes and T. Tezduyar, *Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible euler equations*, Computer methods in applied mechanics and engineering, 45 (1984), pp. 217–284.

[30] T. J. R. Hughes, G. Scovazzi, and T. E. Tezduyar, *Stabilized methods for compressible flows*, Journal of Scientific Computing, (2008).

[31] W. Hui, P. Li, and Z. Li, *A unified coordinate system for solving the two-dimensional euler equations*, J. Comput. Phys., 153 (1999), pp. 596–637.

[32] D. J.Estep, *A short course on duality, adjoint operators, green's functions, and a posteriori error analysis*, Department of Mathematics, Colorado State University, Agust 6, 2004.

[33] C. Johnson, *Adaptive Finite Element Methods for Conservation Laws*, Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, Springer Lecture Notes in Mathematics, Springer Verlag, 1997.

[34] ——, *Adaptive Finite Element Methods for Conservation Laws*, Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, Springer Lecture Notes in Mathematics, Springer Verlag, 1998, pp. 269–323.

[35] C. Johnson and J.Saranen, *Streamline diffusion methods for the incompressible euler and navier-stokes equations*, 47 (1986), pp. 1–18.

[36] C. Johnson and U. Nävert, *An analysis of some finite element methods for advection diffusion problems*, Analytical and Numerical Approaches to Asymptotic Problems in Analysis. Axelsson, S. et all (eds.), North-Holland, 1981.

[37] C. Johnson and A. Szepessy, *Adaptive finite element methods for conservation laws based on a posteriori error estimates.*, vol. 48, Commun. Pure. Appl. Math.

[38] ——, *Shock-capturing streamline diffusion finite element methods for nonlinear conservation laws*, Recent Developments in Computatioonal Fluid Dynamics, edited by T. J. R. Hughes and T. Tezduyar, AMD-Vol 95, 1988.

[39] C. Johnson, A. Szepessy, and P. Hansbo, *On the convergence of shock-capturing streamline diffusion finite element methods for hyperbolic conservation laws*, Mathematics of Computation, 54(189) (1990), pp. 107–129.

[40] N. U. Johnson, C. and J. Pitkäranta, *Finite element methods for linear hyperbolic problems*, 45 (1984), pp. 285–312.

[41] M. Larson and T. Barth, *A posteriori error estimation for discontinuous galerkin approximations of hyperbolic systems*, In Discontinuous Galerkin Methods, B. Cockburn, G. Karniadakis, and C.-W. Shu, editors, Lecture Notes in Comput. Sci. Engrg., 11, Springer, Berlin, (2000), pp. 363–368.

[42] U. Nävert, *A finite element method for convection-diffusion problems*, PhD Thesis, Chalmers TU, Goteborg, 1982.

[43] J. V. Neumann and R. Richtmyer, *A method for the numerical calculation of hydrodynamic shocks.*, J. Appl. Phys., 21 (1950), pp. 232–237.

[44] B. W. Richard Liska, *Comparison of several difference schemes on 1d and 2d test problems for the euler equations*, SIAM, Journal of Scientific Computing, 25 (2003), pp. 995–1017.

[45] P. Sagaut, *Large Eddy Simulation for Incompressible Flows*, Springer-Verlag, Berlin, Heidelberg, New York, 2001.

[46] T. E. Tezduyar and M. Senga, *Stabilization and shock-capturing parameters in supg formulation of compressible flows*, Computer Methods in Applied Mechanics and Engineering, 195 (2006), pp. 1621–1632.

[47] P. Woodward and P. Colella, *The numerical simulation of two-dimensional fluid flow with stong shocks.*, Journal of Computational Physics, 54 (1984), pp. 115–173.

Paper I

A General Galerkin Finite Element Method for the Compressible Euler Equations

Johan Hoffman, Johan Jansson and Murtazo Nazarov
submitted to SIAM Journal of Scientific Computing, 2008.

A GENERAL GALERKIN FINITE ELEMENT METHOD FOR THE COMPRESSIBLE EULER EQUATIONS

JOHAN HOFFMAN, JOHAN JANSSON AND MURTAZO NAZAROV

Abstract. In this paper we present a General Galerkin (G2) method for the compressible Euler equations. The G2 method presented in this paper is a finite element method with linear approximation in space and time, with componentwise stabilization in the form of a weighted least squares stabilization of the convection terms together with residual based shock-capturing, resulting in a symmetric stabilization matrix in the discrete system. The method is cheap and simple to implement and conserves total mass, momentum and energy exactly. We prove an a posteriori energy estimate for the method, and we illustrate the performance of the method in a number of numerical tests in 2d and 3d.

Key words. General Galerkin method, stabilized finite element method, Euler equations, compressible flow

AMS subject classifications. 65M60

1. Introduction. In this note we consider the compressible Euler equations for which we present a stabilized finite element method based on linear approximation in space and time, which we refer to as a *General Galerkin (G2) method*. The G2 method presented here is a generalization of the cG(1)cG(1) method for incompressible flow [7], and we will use cG(1)cG(1) to denote also this method. cG(1)cG(1) has been shown to be efficient for incompressible flow on fixed and moving meshes, see e.g. [3], and has also been shown to be very robust and efficient for turbulent flow computations when used within an adaptive algorithm, see e.g. [8, 4, 5, 6].

Many different stabilized finite element methods for the compressible Euler equations have been proposed, see e.g. [10, 15, 12, 13]. What distinguishes the method in this paper from previous work is that we use individual stabilization of each equation for conservation of mass, momentum and energy, with streamline diffusion least squares stabilization of only the convection term of the corresponding residual, and with componentwise shock-capturing stabilization for each equation. Our formulation thus contains fewer terms than, for example, SUPG or GLS formulations [15], and the computation of the stabilization parameters is very cheap.

We prove that the method conserves total mass, momentum and energy exactly, and we prove a local energy estimate for the method which may be interpreted as representing the 2nd Law of thermodynamics, with the terms from numerical stabilization irreversibly transforming kinetic energy into internal

energy in a dissipative process. These dissipative terms are small in smooth parts of the flow, but large in shocks and turbulence. The energy estimate is of a posteriori character, assuming $\|h^{1/2}R_\alpha\|$ to be bounded independently of h, with $\|\cdot\|$ a L_2-norm, h is the mesh size and R_α the residuals for mass and momentum conservation, an assumption that can be checked a posteriori from the computed solution. With $R_\alpha \sim h^{-1}$ in shocks of width $\sim h$ we have that $\|hR_\alpha\| \sim h^{1/2}$, and from our previous experience with cG(1)cG(1) for incompressible turbulent flow [7] we expect that $\|hR_\alpha\| \sim h^{1/2}$ in turbulent parts of the flow, and thus the assumption is reasonable, since in smooth parts of the flow we expect that $\|hR_\alpha\|$ decays faster than $h^{1/2}$ with decreasing h. In numerical examples, we find that $\|hR_\alpha\|$ is decreasing with h as expected.

The streamline diffusion stabilization corresponds to a perturbation of order h to the equations, which potentially may be very harmful for the accuracy of the method. We may compare to standard artificial viscosity of order h, which is overly dissipative and smears internal shear layers. The streamline diffusion of cG(1)cG(1) is mainly active in the streamline direction, thus with minimal consequence for shear layers in the cross-flow direction: with the streamline derivative of order ~ 1 in shear layers, to compare with standard artificial viscosity with cross-flow derivatives of order $\sim h^{-1/2}$. In shocks the streamline diffusion is typically dominated by the residual based shock capturing terms.

1.1. Relation to previous work. Woodward and Colella summarized three main approaches to the representation of discontinuities and shocks [16]: *artificial viscosity technique*, *linear hybridization* and *Godunov's approach*. Godunov's approach is based on a Riemann solver and solves the equations exactly at the interior boundaries or approximate it using polynomials. In general, it is hard to use this approach for unstructured meshes and complex geometries, because it is difficult to control and choose the neighboring cells. Artificial viscosity and linear hybridization are in some sense similar to each other. Both of them are designed to have a special approach for the discontinuity regions. In an artificial viscosity approach the region with a steep derivative is smeared out by adding viscous terms which have small effect in smooth regions, whereas in linear hybridization this regions are treated with a low order dissipative scheme. However, the main problem which remains is to find the right region to do a special treatment. The G2 method presented here belongs to the artificial viscosity approach, with the artificial viscosity based on the residual of the equations.

Artificial viscosity was suggested by Neumann and Richtmyer [14] and has been much developed over the last decades. Too much artificial viscos-

ity makes the solution smooth, so to capture discontinuities the mesh has to be refined sufficiently. Uniform mesh refinement of the mesh is typically too expensive, but an adaptive algorithm can be used with only local mesh refinement.

The work by Johnson, Hansbo and Szepessy [12, 13, 2], and Hughes and Tezduyar [10] is focused on the optimal choice of artificial viscosity and streamline diffusion terms (SD and SUPG, streamline-upwind/Petrov-Galerkin methods). The G2 method in this paper is a simplified version of the above methods, where artificial viscosity and streamline diffusion terms are defined individually for each component, without the use of eigenvalues and eigenvectors of the linearized operator. Still the results below are satisfactory with the characteristics of the flow well resolved.

1.2. Outline. We first introduce the Euler equations in Section 2 and the cG(1)cG(1) method in Section 3. Then we derive a local energy estimate in Section 4, discuss implementation of the method in Section 5, and finally we present numerical results illustrating the method in Section 6.

2. The Euler equations. The compressible *Euler equations* expresses conservation of mass, momentum and total energy for an inviscid fluid enclosed in a fixed (open) domain Ω in three-dimensional space $\mathbb{R}^3$ with boundary Γ over a time interval $[0, \hat{t}\,]$ with initial time zero and final time $\hat{t}$.

We seek the *density* ρ, *momentum* $m = \rho u$, with $u = (u_1, u_2, u_3)$ the *velocity*, and the *total energy* e as functions of $(x, t) \in \Omega \cup \Gamma \times [0, \hat{t}\,]$, where $x = (x_1, x_2, x_3)$ denotes the coordinates in $\mathbb{R}^3$ and u_i is the velocity in the x_i-direction. The Euler equations for $\hat{u} \equiv (\rho, m, e)$ read with $Q = \Omega \times I$ and $I = (0, \hat{t}\,]$:

$$\begin{aligned} \dot{\rho} + \nabla \cdot (\rho u) &= 0 && \text{in } Q, \\ \dot{m}_i + \nabla \cdot (m_i u) + p_{,i} &= f_i && \text{in } Q, \quad i = 1, 2, 3, \\ \dot{e} + \nabla \cdot (eu + pu) &= 0 && \text{in } Q, \\ u \cdot n &= 0 && \text{on } \Gamma \times I, \\ \hat{u}(\cdot, 0) &= \hat{u}^0 && \text{in } \Omega, \end{aligned} \tag{2.1}$$

where $p = p(x, t)$ is the *pressure* of the fluid, $p_{,i} = \partial p / \partial x_i$ is the partial derivative with respect to x_i, the dot indicates differentiation with respect to time, n denotes the outward unit normal to Γ and $f = (f_1, f_2, f_3)$ is a given *volume force* (like gravity) acting on the fluid, and $\hat{u}^0 = \hat{u}^0(x)$ represents initial conditions. Further, the total energy $e = k + \theta$, where $k = \rho |u|^2/2$ is the *kinetic energy*, with $|u|^2 \equiv u_1^2 + u_2^2 + u_3^2$, and $\theta = \rho T$ is the *internal energy* with T the *temperature* scaled so that $c_v = 1$, where c_v is the heat capacity under constant volume.

The boundary condition $u \cdot n = 0$ is a *slip boundary condition* requiring the normal velocity $u \cdot n$ to vanish corresponding to an impenetrable boundary with zero friction.

The number of unknowns including the pressure is six but there are only five equations in the Euler system (2.1), and so we close the system with the *state equation* of a *perfect gas*;

$$p = (\gamma - 1)\rho T, \tag{2.2}$$

expressing the pressure p as a function of density ρ and temperature T, where $\gamma = c_p$ is the *adiabatic index* with c_p the heat capacity under constant pressure, and $(\gamma - 1)$ is the *gas constant*.

For a perfect gas, the *speed of sound* c is given by $c^2 = \gamma(\gamma - 1)T$, and the *Mach number* is defined as $M = |u|/c$, with u the velocity of the gas.

3. cG(1)cG(1) for the Euler equations. cG(1)cG(1) is a G2 method using the continuous Galerkin method cG(1) in space and time. With cG(1) in time the trial functions are continuous piecewise linear and the test functions piecewise constant. cG(1) in space corresponds to both test functions and trial functions being continuous piecewise linear. Let $0 = t_0 < t_1 < ... < t_N = \hat{t}$, be a sequence of discrete time steps with associated time intervals $I_n = (t_{n-1}, t_n]$ of length $\Delta t_n = t_n - t_{n-1}$ and space-time slabs $S_n = \Omega \times I_n$, and let $W_h \subset H^1(\Omega)$ be a finite element space consisting of continuous piecewise linear functions on a fixed mesh $\mathcal{T}_h = \{K\}$ of mesh size $h(x) < 1$, with elements K. Further, let $W_{h,0}$ be the space of vector functions in W_h^3 satisfying the slip boundary condition, that is $W_{h,0} = \{w \in W_h^3 : w \cdot n = 0 \text{ on } \Gamma\}$

We now seek functions $\hat{u}_h = (\rho_h, m_h, e_h)$, continuous piecewise linear in space and time. The cG(1)cG(1) method for the compressible Euler equations, here without source terms for simplicity, reads: For $n = 1, ..., N$, find $\hat{u}_h^n = (\rho_n, m_n, e_n) \equiv (\rho_h(t_n), m_h(t_n), e_h(t_n))$ with $\hat{u}_h^n \in V_h \equiv W_h \times W_{h,0} \times W_h$, such that

$$\begin{aligned}
&(\dot{\rho}_n, v_n^\rho) - (u_n \bar{\rho}_n, \nabla v_n^\rho) \\
&\quad + (\delta u_n \cdot \nabla \bar{\rho}_n, u_n \cdot \nabla v_n^\rho) + (\hat{\nu}_\rho \nabla \bar{\rho}_n, \nabla v_n^\rho) = 0, \qquad (3.1)\\
&(\dot{m}_n, v_n^m) - (u_n \bar{m}_n, \nabla v_n^m) - (p_n, \nabla \cdot v_n^m) \\
&\quad + (\delta u_n \cdot \nabla \bar{m}_n, u_n \cdot \nabla v_n^m) + (\hat{\nu}_m \nabla \bar{m}_n, \nabla v_n^m) = 0, \qquad (3.2)\\
&(\dot{e}_n, v_n^e) - (u_n \bar{e}_n, \nabla v_n^e) + (\nabla \cdot (u_n p_n), v_n^e) \\
&\quad + (\delta u_n \cdot \nabla \bar{e}_n, u_n \cdot \nabla v_n^e) + (\hat{\nu}_e \nabla \bar{e}_n, \nabla v_n^e) = 0, \qquad (3.3)
\end{aligned}$$

for all test functions $\hat{v}_n = (v_n^\rho, v_n^m, v_n^e) \in V_h$, where

$$(v, w) = \sum_{K \in \mathcal{T}_h} \int_K v \cdot w \, dx,$$

and

$$\dot{\rho}_n = \frac{\rho_n - \rho_{n-1}}{\Delta t_n}, \quad \dot{m}_n = \frac{m_n - m_{n-1}}{\Delta t_n}, \quad \dot{e}_n = \frac{e_n - e_{n-1}}{\Delta t_n},$$

$$\bar{\rho}_n = \frac{1}{2}(\rho_n + \rho_{n-1}), \quad \bar{m}_n = \frac{1}{2}(m_n + m_{n-1}), \quad \bar{e}_n = \frac{1}{2}(e_n + e_{n-1}),$$

where we define u_n, p_n and T_n to be finite element functions in $W_{h,0}$, W_h and W_h respectively, which are defined by their nodal values given by

$$\begin{aligned} u_n(N_i) &= \bar{m}_n(N_i)/\bar{\rho}_n(N_i), \\ p_n(N_i) &= (\gamma - 1)\bar{\rho}_n(N_i)T_n(N_i), \\ T_n(N_i) &= \bar{e}_n(N_i)/\bar{\rho}_n(N_i) - |u_n(N_i)|^2/2 \end{aligned}$$

for all nodes N_i in the mesh $\mathcal{T}_h$, and similar for p_n. We use the stabilization parameters $\delta = C_\delta(\Delta t_n^{-2} + |u_n|^2 h^{-2})^{-1/2}$ and $\hat{\nu}_\alpha = \max\left(C_\alpha |R_\alpha(\hat{u})| h^2, C_h h^{3/2}\right)$ $(\alpha = \rho, m, e)$, with constants C_δ, C_α and $C_h \sim \frac{U}{\sqrt{L}}$, where U and L are characteristic velocity and length scales, and we define the normalized strong residuals by

$$R_\rho(\hat{u}_h) = (\dot{\rho}_n + \nabla \cdot (\bar{\rho}_n u_n))/|\bar{\rho}_n|, \tag{3.4}$$

$$R_{m_j}(\hat{u}_h) = ((\dot{m}_n)_j + \nabla \cdot ((\bar{m}_n)_j u_n) + \nabla p_n)/(|(\bar{m}_n)_j| + \epsilon), \quad j = 1, 2, 3, \tag{3.5}$$

$$R_e(\hat{u}_h) = (\dot{e}_n + \nabla \cdot (\bar{e}_n u_n + p_n u_n))/|\bar{e}_n|, \tag{3.6}$$

for $t \in I_n$, and with $\epsilon > 0$ a small safety factor. The time step Δt_n be given by a CFL-condition, with typically $\Delta t_n \sim \min_{E_j \in \mathcal{T}_h}(h/|u_n|)_{E_j}$, for all elements E_j in the mesh $\mathcal{T}_h$.

For $t \in I_n$ we may also write cG(1)cG(1) as

$$R_\rho(\rho_h; v_n^\rho) + SD_\rho(\rho_h; v_n^\rho) = 0, \tag{3.7}$$

$$R_m(m_h; v_n^m) + SD_m(m_h; v_n^m) = 0, \tag{3.8}$$

$$R_e(e_h, v_n^e) + SD_e(e_h; v_n^e) = 0, \tag{3.9}$$

with

$$R_\rho(\rho_h; v_n^\rho) = (\dot\rho_n, v_n^\rho) - (u_n\bar\rho_n, \nabla v_n^\rho), \tag{3.10}$$
$$R_m(m_h; v_n^m) = (\dot m_n, v_n^m) - (u_n\bar m_n, \nabla v_n^m) - (p_n, \nabla\cdot v_n^m), \tag{3.11}$$
$$R_e(e_h; v_n^e) = (\dot e_n, v_n^e) - (u_n\bar e_n, \nabla v_n^e) + (\nabla\cdot(u_n p_n), v_n^e), \tag{3.12}$$

and

$$SD_\rho(\rho_h; v_n^\rho) = (\delta u_n\cdot\nabla\bar\rho_n, u_n\cdot\nabla v_n^\rho) + (\hat\nu_\rho\nabla\bar\rho_n, \nabla v_n^\rho), \tag{3.13}$$
$$SD_m(m_h; v_n^m) = (\delta u_n\cdot\nabla\bar m_n, u_n\cdot\nabla v_n^m) + (\hat\nu_m\nabla\bar m_n, \nabla v_n^m), \tag{3.14}$$
$$SD_e(e_h; v_n^e) = (\delta u_n\cdot\nabla\bar e_n, u_n\cdot\nabla v_n^e) + (\hat\nu_e\nabla\bar e_n, \nabla v_n^e). \tag{3.15}$$

We note that we here stabilize each equation of (2.1) individually, rather than stabilizing the complete system (2.1) using a stabilization matrix, which is a common approach, see e.g. [15]. Our approach here is cheap and very simple to implement, and is robust with little smearing of discontinuities, as shown in numerical experiments below.

3.1. Conservation of mass, momentum and total energy. To prove exact conservation of mass and energy we simply choose the test functions corresponding to the respective equation to unity:

$$0 = R_\rho(\rho_n; 1) + SD_\rho(\rho_n; 1) = (\dot\rho_n, 1) = \frac{d}{dt}\int_\Omega \rho_n(x)\ dx, \tag{3.16}$$
$$0 = R_e(e_n; 1) + SD_e(e_n; 1) = (\dot e_n, 1) = \frac{d}{dt}\int_\Omega e_n(x)\ dx. \tag{3.17}$$

For the momentum equation we use the following approximation of the pressure force on the boundary Γ in the direction $\phi = (\phi_1, \phi_2, \phi_3)$:

$$\int_\Gamma p(n\cdot\phi)\ ds \approx N^h(\Phi) \equiv R_m(m_h; \Phi) + SD_m(m_h; \Phi), \tag{3.18}$$

with $\Phi \in W_h^3$ and $\Phi = \phi$ on the boundary Γ. For a derivation of this approximation, see e.g. [1, 7]. Now, by choosing the direction $\Phi = e_j$, with e_j the standard unit vector for component j, we get conservation of momentum with the change in momentum equal to the surface pressure force $N^h(e_j)$:

$$N^h(e_j) = R_m(m_h; e_j) + SD_m(m_h; e_j) = (\dot m_n, e_j) = \frac{d}{dt}\int_\Omega (m_n)_j(x)\ dx. \tag{3.19}$$

3.2. Solution of the discrete system. To solve the system (3.1)-(3.3) we use an outer fix point iteration with an inner GMRES solver for the linearized problem, with the velocity given by the previous iteration. With this method the maximum time step length is determined by a CFL condition with a CFL number of the order 0.1. This solution strategy is probably not the most efficient, and alternative approaches will be investigated in future work, including explicit methods.

4. Energy estimate for cG(1)cG(1). Assuming that there exists an exact solution $\hat{u}$ to the Euler equations (2.1), we would have that

$$\dot{K} + W = 0, \tag{4.1}$$

with the total kinetic energy

$$K \equiv \int_\Omega k \, dx, \quad k = \rho|u|^2/2,$$

and the total work

$$W \equiv \int_\Omega w \, dx, \quad w \equiv \nabla p \cdot u,$$

which is obtained by scalar multiplication of the momentum equation by u and integration over Ω. But since existence of solutions to the Euler equations is unknown, this argument is empty. On the other hand, by an argument of vanishing viscosity, see e.g. [7], we may formally derive the following version of the 2nd Law of thermodynamics for the Euler equations:

$$\dot{K} + W + D = 0, \tag{4.2}$$

where $D \geq 0$ represents irreversible loss of kinetic energy due to turbulence and shocks, expressing that kinetic energy can be converted into work W or dissipate by D into internal energy, but only W can be converted back to kinetic energy.

Let us now derive a local 2nd Law for the cG(1)cG(1) solution $\hat{u}_h$, corresponding to (4.2), using the following notation

$$\dot{k}_n \equiv \dot{m}_n \cdot u_n/2, \quad k_n \equiv \rho_n|u_n|^2/2, \quad \bar{k}_n \equiv \bar{\rho}_n|u_n|^2/2, \quad w_n \equiv \nabla p_n \cdot u_n.$$

In the analysis below we assume that $\hat{\nu}_\alpha = \hat{\nu}$ for all α, for simplicity. We let w^h denote the interpolant of a function w in the finite element space of test

functions of cG(1)cG(1), and we let $\phi \in \mathcal{C}_0^\infty(Q)$ be a positive test function with local support on a subdomain of Q. We then have that for $t \in I_n$:

$$\begin{aligned}
0 &= R_m(m_h; (u_n\phi)^h) + SD_m(m_h; (u_n\phi)^h) \\
&= R_m(m_h; u_n\phi) + SD_m(m_h; u_n\phi) \\
&\quad -(R_m(m_h; u_n\phi - (u_n\phi)^h) + SD_m(m_h; u_n\phi - (u_n\phi)^h)) \\
&= (\dot{k}_n + \nabla\cdot(\bar{k}_n u_n) + w_n, \phi) + R_\rho(\rho_h; |u_n|^2/2\phi) + SD_m(m_h; u_n\phi) \\
&\quad -(R_m(m_h; u_n\phi - (u_n\phi)^h) + SD_m(m_h; u_n\phi - (u_n\phi)^h)) \\
&= (\dot{k}_n + \nabla\cdot(\bar{k}_n u_n) + w_n, \phi) + SD_m(m_h; u_n\phi) - SD_\rho(\rho_h; |u_n|^2/2\phi) \\
&\quad -(R_m(m_h; u_n\phi - (u_n\phi)^h) + SD_m(m_h; u_n\phi - (u_n\phi)^h)) \\
&\quad +R_\rho(\rho_h; |u_n|^2/2\phi - (|u_n|^2/2\phi)^h) + SD_\rho(\rho_h; |u_n|^2/2\phi - (|u_n|^2/2\phi)^h).
\end{aligned}$$

We further have that

$$\begin{aligned}
SD_m(m_h; u_n\phi) - SD_\rho(\rho_h; |u_n|^2/2\phi) &= (\delta\bar{\rho}_n u_n\cdot\nabla u_n, u_n\cdot\nabla u_n\phi) \\
&\quad +(\hat{\nu}\bar{\rho}_n\nabla u_n, \nabla u_n\phi) + (\delta u_n\cdot\nabla\phi, u_n\cdot\nabla\bar{k}_n) + (\hat{\nu}\nabla\phi, \nabla\bar{k}_n),
\end{aligned}$$

so that using the notation

$$d_n \equiv (\delta\bar{\rho}_n u_n\cdot\nabla u_n)\cdot(u_n\cdot\nabla u_n) + (\hat{\nu}\bar{\rho}_n\nabla u_n)\cdot\nabla u_n,$$

we have, by integration in time over all time intervals I_n, that

$$\sum_{n=1}^N \int_{I_n} (\dot{k}_n + \nabla\cdot(\bar{k}_n u_n) + w_n + d_n, \phi)\, dt = \mathcal{E}, \tag{4.3}$$

with

$$\begin{aligned}
\mathcal{E} \equiv \sum_{n=1}^N \int_{I_n} &(-(R_m(m_h; u_n\phi - (u_n\phi)^h) + SD_m(m_h; u_n\phi - (u_n\phi)^h)) \\
&+R_\rho(\rho_h; |u_n|^2/2\phi - (|u_n|^2/2\phi)^h) + SD_\rho(\rho_h; |u_n|^2/2\phi - (|u_n|^2/2\phi)^h) \\
&+(\delta u_n\cdot\nabla\phi, u_n\cdot\nabla k_n) + (\hat{\nu}\nabla\phi, \nabla k_n))\, dt.
\end{aligned} \tag{4.4}$$

With the suitable stability estimates available for the cG(1)cG(1) discretization of the equations for mass and momentum conservation, $\mathcal{E}$ could be bounded by a constant, independent of the mesh size h, times the square root of the largest element diameter of the mesh. For incompressible flow we have this type of stability estimates, see e.g. [7], but for compressible flow this result is not available a priori.

On the other hand, since $\mathcal{E}$ is function of $\hat{u}_h$ it can be evaluated a posteriori. That is, we may formulate the following a posteriori version of a 2nd Law for cG(1)cG(1):

THEOREM 4.1. *Assume that there exists a constant $C > 0$, independent of the mesh size h, such that $|\mathcal{E}| \leq C\sqrt{h_{max}}$, with h_{max} the largest element diameter of the mesh. We then have for the cG(1)cG(1) solution $\hat{u}_h$:*

$$|\sum_{n=1}^{N} \int_{I_n} (\dot{k}_n + \nabla \cdot (\bar{k}_n u_n) + w_n + d_n, \phi)\ dt| \leq C\sqrt{h_{max}}. \tag{4.5}$$

To bound $|\mathcal{E}|$ a priori, we would need stability estimates of the type $\|\sqrt{h}R_\alpha(\hat{u}_h)\| \leq C$, connected to the streamline diffusion stabilization, together with corresponding stability estimates connected to the shock-capturing terms. We would then be able to use super approximation [12] and standard interpolation error estimates to derive a bound of the type $|\mathcal{E}| \leq C\sqrt{h_{max}}$. The main obstacle seems to be the inability to a priori bound the work term w_n, since under expansion this term may be large. The analysis in [9] indicates that an additional stabilization coupled to this term may give the result a priori.

cG(1)cG(1) is used within an adaptive mesh refinement algorithm, so we can monitor the condition of Theorem 4.1 as we refine the mesh. If we find that this assumption is satisfied, then we would also know that the solution satisfies a 2nd Law of the form (4.5). In the numerical tests below we observe that the assumption in satisfied when we refine the mesh.

5. Implementation - Unicorn. The numerical implementation is done in Unicorn - a unified continuum mechanics solver based on adaptive finite element methods, which is part of a free software project FEniCS (www.fenics.org). In Unicorn adaptivity is implemented in the following way: first a solution is computed until the given final time, and then the cells of the mesh are refined based on the error indicator. The procedure is repeated until some given tolerance is matched.

The code used to generate the results below can be found in the Unicorn repositories, so that the results may be reproduced by the reader.

5.1. Slip boundary conditions. In fluid dynamics slip boundary conditions are typically used for compressible flows. The slip boundary condition is

$$u \cdot n = 0 \qquad \text{on } \Gamma \times I,$$

which means that the normal velocity on the boundary is zero. In other words the pressure is the only nonzero flux on the given boundary $\Gamma \times I$.

In our computations, we implement the slip boundary condition strongly. By "strongly" we here mean an implementation of the boundary condition after assembling the stiffness matrix and the load vector, whereas a "weak" implementation involves adding boundary integrals in the variational formulation. The row of the matrix and load vector corresponding to a vertex is found and replaced by a new row according to the boundary condition.

The idea is as follows: Initially, the test function v is expressed in the Cartesian standard basis (e_1, e_2, e_3). Now, the test function is mapped locally to normal-tangent coordinates with the basis (n, τ_1, τ_2), where $n = (n_1, n_2, n_3)$ is the normal, and $\tau_1 = (\tau_{11}, \tau_{12}, \tau_{13})$, $\tau_2 = (\tau_{21}, \tau_{22}, \tau_{23})$ are tangents to each node on the boundary. This allows us to let the normal direction to be constrained and the tangent directions be free:

$$v = (v \cdot n)n + (v \cdot \tau_1)\tau_1 + (v \cdot \tau_2)\tau_2.$$

For the matrix and vector this means that the rows corresponding to the boundary need to be multiplied with n, τ_1, τ_2, respectively, and then the normal velocity should be put 0.

In the 2D case there is only one tangential vector so that the new basis is (n, τ). Here we present the implementation for 3D, but modification to 2D is straight forward.

We split the boundary vertices of the computational domain into three types, see Fig. 5.1:

- type S_1 - where a vertex is on a surface,
- type S_2 - where a vertex is on a edge,
- type S_3 - where a vertex is on a corner.

5.1.1. Normals and tangential vectors. The discussion below is for convex boundaries, but is valid for concave boundaries as well. For the first type, S_1, we follow the algorithm in [11] to find the normal and tangential vectors. Here the tangential vectors are calculated for a given normal vector $n = (n_1, n_2, n_3)$, with the normal vector given for each facet of an element.

We define facets to belong to the same surface if the angle, α, between the facet normals are less in absolute value than $\alpha_{max} > 0$. In this paper we choose $\alpha_{max} = \frac{\pi}{6}$. In each surface a normal to a vertex is calculated as a weighted average of all the facet normals which contain the vertex with the weight being the area of each facet.

We define the vertex to belong to an *edge*, or type S_2, if the vertex belongs to two surfaces. In this case, we take the normal from the first surface as vector n; and the normal from the second surface as τ_1. The last vector τ_2 is then a

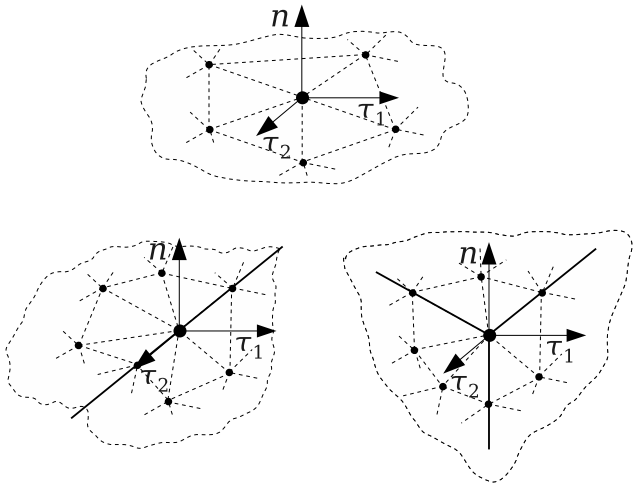

FIG. 5.1. *The possible types of the boundary, S_1 (upper), S_2 (lower left) and S_3 (lower right).*

cross product of n and τ_1, $\tau_2 = n \times \tau_1$. When $\alpha = \frac{\pi}{2}$ normals are perpendicular. But in general vectors n and τ_1 are not necessarily perpendicular, Fig. 5.2.

The third case is a vertex which belongs to more than two surfaces. Again the angle between normals on the surfaces, α, is checked. If there are more than two surfaces we say the vertex belongs to a *corner*, or type S_3.

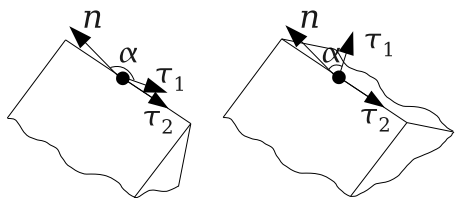

FIG. 5.2. *Some different types of edges.*

5.1.2. Implementation in Unicorn. The following substitutions are done to the corresponding rows on the linear system. Let $\mathbf{A}$ and $\mathbf{b}$ be a matrix and load vector. Let i be a vertex on the boundary and N be the number of all vertices on the mesh. We know that in 3D, where the number of unknowns of the system is 5, then $\mathbf{A}$ is a $(5N, 5N)$ sparse matrix. With the numbering

scheme used in Unicorn, for the given vertex i the system is:

$$\begin{pmatrix} \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & a_{i,N} & \cdots & a_{i,2N} & \cdots & a_{i,3N} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & a_{i+N,N} & \cdots & a_{i+N,2N} & \cdots & a_{i+N,3N} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & a_{i+2N,N} & \cdots & a_{i+2N,2N} & \cdots & a_{i+2N,3N} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{pmatrix} \cdot \begin{pmatrix} \cdots \\ \rho u_1 \\ \cdots \\ \rho u_2 \\ \cdots \\ \rho u_3 \\ \cdots \end{pmatrix} = \begin{pmatrix} \cdots \\ l_N \\ \cdots \\ l_{2N} \\ \cdots \\ l_{3N} \\ \cdots \end{pmatrix}$$

Let us to discuss the implementation of the slip boundary condition for each part separately.

(a) type S_1. After a local coordinate mapping from the Cartesian to the normal- tangent coordinate system and putting the normal velocity to zero, the matrix $\mathbf{A}$ takes the following form.

$$\begin{pmatrix} \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & n_1 & 0 & n_2 & 0 & n_3 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & a'_{i+N,N} & \cdots & a'_{i+N,2N} & \cdots & a'_{i+N,3N} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & a''_{i+2N,N} & \cdots & a''_{i+2N,2N} & \cdots & a''_{i+2N,3N} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{pmatrix} \cdot \begin{pmatrix} \cdots \\ \rho u_1 \\ \cdots \\ \rho u_2 \\ \cdots \\ \rho u_3 \\ \cdots \end{pmatrix} = \begin{pmatrix} \cdots \\ 0 \\ \cdots \\ l'_{2N} \\ \cdots \\ l''_{3N} \\ \cdots \end{pmatrix}$$

where

$$\begin{aligned} a'_{i+N,j} &= a_{i,j}\tau_{11} + a_{i+N,j}\tau_{12} + a_{i+2N,j}\tau_{13}, \quad j = 1, 2, \cdots, 5N, \\ l'_N &= l_N\tau_{11} + l_{2N}\tau_{12} + l_{3N}\tau_{13}, \end{aligned}$$

and

$$\begin{aligned} a''_{i+N,j} &= a_{i,j}\tau_{21} + a_{i+N,j}\tau_{22} + a_{i+2N,j}\tau_{23}, \quad j = 1, 2, \cdots, 5N, \\ l''_{2N} &= l_N\tau_{21} + l_{2N}\tau_{22} + l_{3N}\tau_{23}. \end{aligned}$$

Note that the mapping is done locally only for the vertex on the boundary, which means the rest of the rows remain the same.

(b) type S_2. In this case, there are two normals by the above construction, n and τ_1. Velocity in these directions is constrained. The velocity in the third

direction, τ_2, is free. The linear system takes the following form:

$$\begin{pmatrix} \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & n_1 & 0 & n_2 & 0 & n_3 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \tau_{11} & 0 & \tau_{12} & 0 & \tau_{13} & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & a''_{i+2N,N} & \cdots & a''_{i+2N,2N} & \cdots & a''_{i+2N,3N} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{pmatrix} \cdot \begin{pmatrix} \cdots \\ \rho u_1 \\ \cdots \\ \rho u_2 \\ \cdots \\ \rho u_3 \\ \cdots \end{pmatrix} = \begin{pmatrix} \cdots \\ 0 \\ \cdots \\ 0 \\ \cdots \\ l''_{3N} \\ \cdots \end{pmatrix}$$

(c) type S_3. For a corner, the velocity in all directions is constrained. Therefore, it is the same as applying a no-slip or zero velocity boundary condition for this vertex. Meaning here that the row is replaced by an identity vector and the right hand side is set to zero.

To ensure good conditioning of the matrix the rows are finally rearranged to have the largest matrix elements on the diagonal.

6. Numerical results. We now present numerical results for the cG(1)cG(1) method of this paper. We first test the method for a standard shock tube problem to be able to compare our results with the available exact analytical solution. We then test the method for problems in 2D and 3D.

6.1. Shocktube. We consider the 1D shocktube problem for a domain Ω being a unit square. The domain is divided into two equal rectangular subdomains. To the left a gas is at rest at high pressure, and to the right at low pressure. At time $t = 0$ the divider is removed and the gas in the two domains is allowed to move.

We assume an ideal gas with gas constant $\gamma = 1.4$, and the initial conditions of $\hat{u} = (\rho, u, p)$ to the left and right are $(1, 0, 2.5)$ and $(0.125, 0, 0.25)$, respectively, or $(\rho, u, p) = (1, 0, 1)$ and $(\rho, u, p) = (0.125, 0, 0.1)$.

The solution of the coarse mesh is compared with solutions obtained after uniform refinement of the mesh. This is a 1D problem for which we can obtain an exact solution. We compare this solution with cG(1)cG(1) solutions on 5 different meshes, uniform with 30×30, 60×60, 120×120, 240×240 and 480×480 vertices, see Fig 6.1. The exact solution for the compressible Euler equations is plotted together with corresponding approximations using cG(1)cG(1).

From the results we see that the cG(1)cG(1) approximate solution converges to the exact solution with mesh refinement. In the smooth region the method gives a high order of accuracy and the shock and contact discontinuities are well resolved by the refined mesh.

The Fig 6.3 shows the error indicator $\|hR(\hat{u}_h)\|$, where h is a mesh size and $R(\hat{u}_h)$ is the normalized strong residual defined in (3.4)-(3.6), as a function of the number of cells in uniform mesh refinements, decreasing as the mesh is refined.

6.2. Wind tunnel with a step. Here we consider Mach 3 flow in a wind tunnel with a step. This is a well known benchmark used for testing new methods, see e.g. [16]. Here we consider the two dimensional case.

The wind tunnel with length 3 and height 1 contains a step with height 0.2, situated at a distance 0.6 from the inflow. The initial data for the problem is $\rho = 1.4, m = (4.2, 0), e = 8.8$. The flow is affected by the numerical error appearing from the neighborhood of the singular corner, this affect can be seen in the bottom Mach stem. Woodward and Colella, [16], proposed to apply a special kind of boundary condition for the points close to the corner, to avoid the effect from the singularity. Another approach could be to round off the nonphysical sharp corner and allow the adaptive mesh refinement to follow the smooth structure of the rounded corner. But, in this paper no special treatment of this singular corner was made.

In a case of supersonic flow all characteristics of the Euler equations go to the right, therefore no physical boundary conditions are needed on the outflow. A slip boundary condition is applied on the channel walls. Usually, for numerical tests this simulation is done until time $t = 4$, and a steady solution and is obtained at time $t = 12$.

Fig.6.4 shows the results at time $t = 4$, where adaptive mesh refinement is used with respect to the error indicator $\|hR(\hat{u}_h)\|_K$ for each cell K. Fig.6.5 shows the mesh after 19 adaptive refinements of 10% of the cells of mesh at each iteration, the finest mesh has 33 102 vertices. Mesh refinement is concentrated at shocks and contact discontinuities by the error indicator, which are increasingly resolved by the adaptive algorithm.

6.3. Wind tunnel with a cylinder in 2D and a sphere in 3D. We consider the wind tunnel of length 3 and height 1 with a cylinder (sphere) with readius 0.02 centered in the channel at a distance $x = 0.3$ from the inlet. The initial conditions are as above, the flow is supersonic with Mach number 3. The inflow boundary condition is chosen to be the same as initial data: $\rho = 1.4, m = (4.2, 0, 0), e = 8.8$, and no boundary conditions needed for the outflow. A slip boundary condition is applied for the rest of the boundaries.

As in the previous example, we use adaptive mesh refinement with respect to the error indicator $\|hR(\hat{u}_h)\|_K$. Fig.6.6 shows that the norm of the error indicator decays with mesh refinements, and by mesh refinement the shock gets sharper as expected. Fig.6.7 shows a colormap and Fig.6.8 shows 36

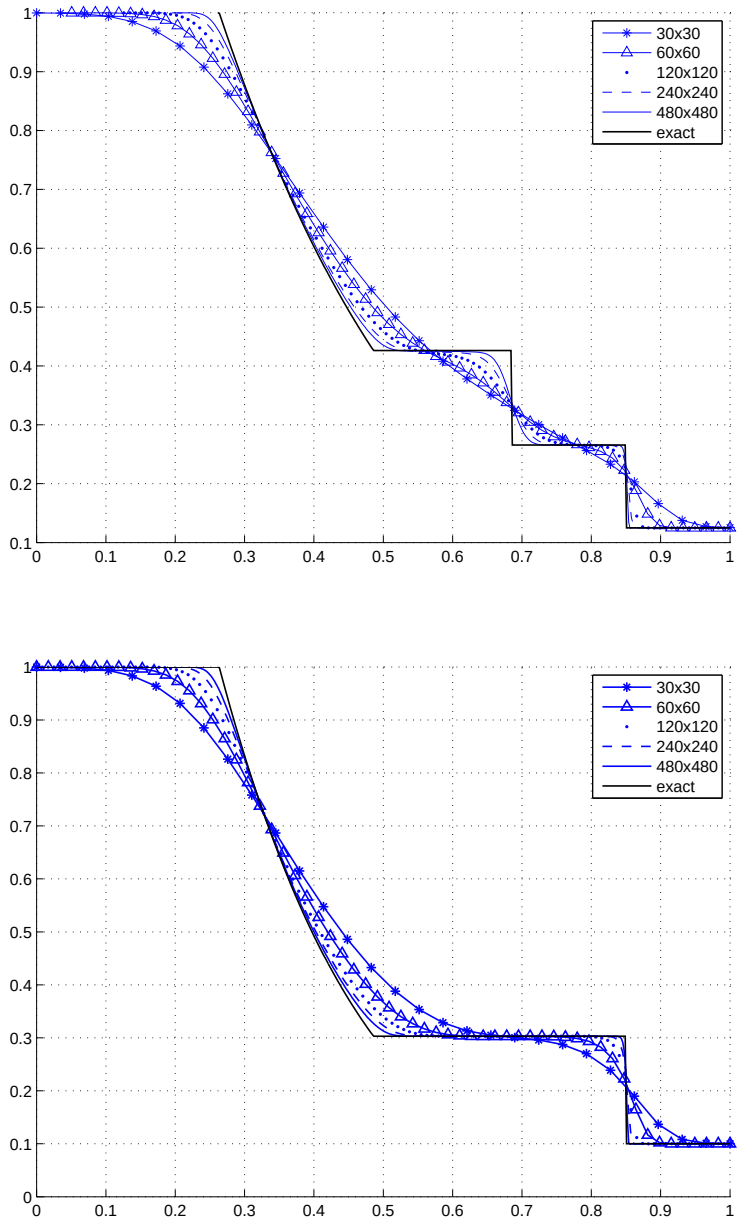

FIG. 6.1. *The density (upper) and pressure (lower) in different meshes.*

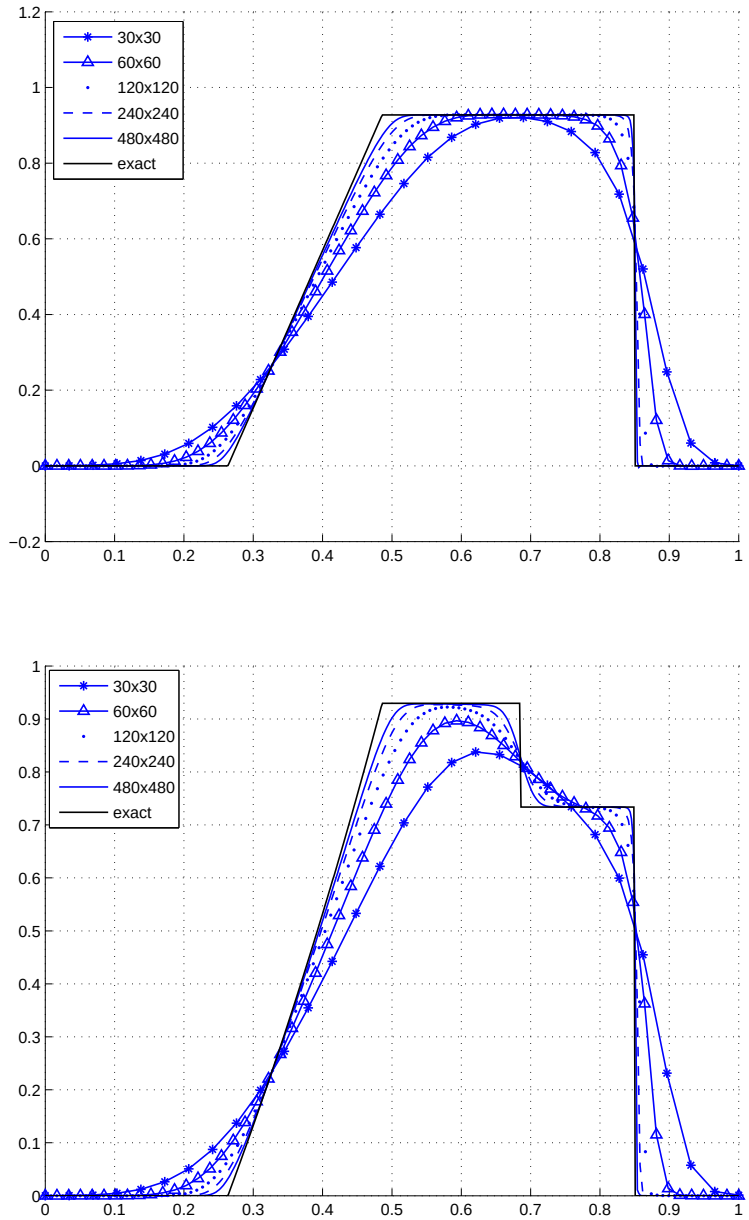

FIG. 6.2. *The velocity (upper) and Mach number (lower) in different meshes.*

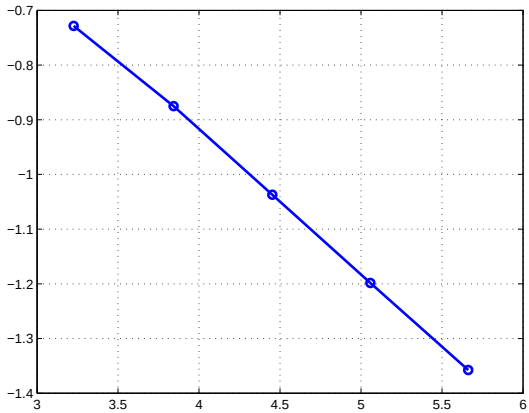

FIG. 6.3. *Log10 of the number of cells vs log10 of the L_2 norm of error indicator, $\|hR(\hat{u}_h)\|$, for the shock-tube problem. The mesh is refined uniformly 4 times, the finest mesh has 230 400 vertices and 458 882 cells.*

contours of density, Mach number and velocity in the streamwise direction in 2D at time $t = 0.4$. At this time the shock reaches the channel boundaries and gets reflected. For longer time this reflected shock develops as in the previous example. We are here only interested to see how the shock and rearfaction waves develop for the supersonic flow around the cylinder and sphere, before the shock reaches the channel boundaries.

Fig.6.10 shows density and Mach number in 3D, for a mesh adaptively refined 11 times by 5% of the cells with 91529 vertices. The adaptive algorithm with above error indicator refines the mesh where expected.

7. Conclusion and future direction. We have presented a General Galerkin method for solving the compressible Euler equations, which is a stabilized finite element method. We showed that the stabilization is simple, but still the method works well for the class of problems tested. The method conserves mass, momentum and energy, which is one of the important characteristics of the numerical methods for conservation laws. In this paper we also showed an energy estimate for the cG(1)cG(1) solution, and we illustrated the performance of the G2 method for some standard test problem in 2D and 3D.

Overall, we conclude that the G2 method is robust, easy to implement and works well for solving compressible equations in 2D and 3D.

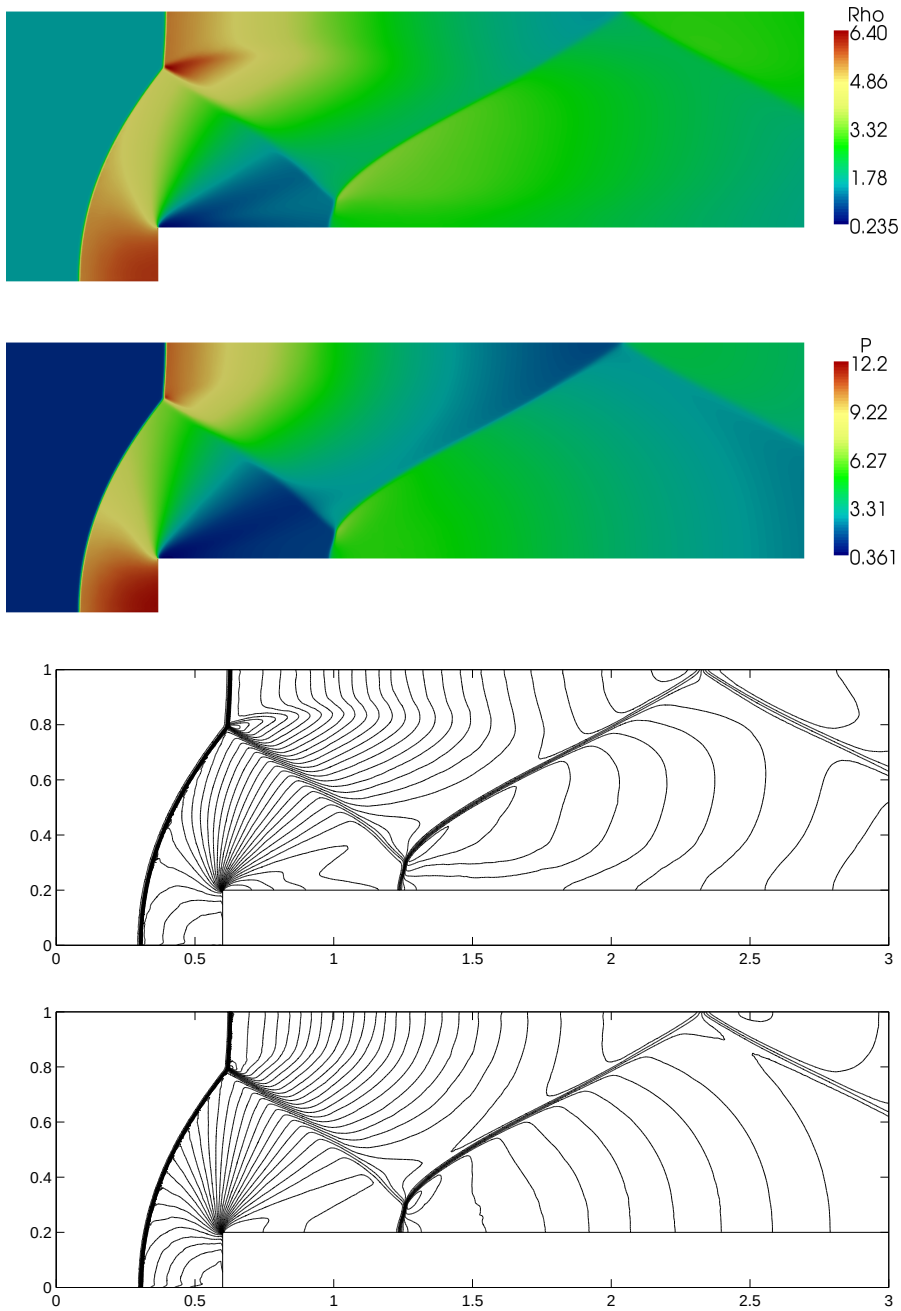

FIG. 6.4. *The results with a colormap and 30 contours: density (upper) and pressure (lower). There is no special treatment done at the singularity corner, therefore a boundary layer appears there and affects the Mach stem in the bottom of the computational domain.*

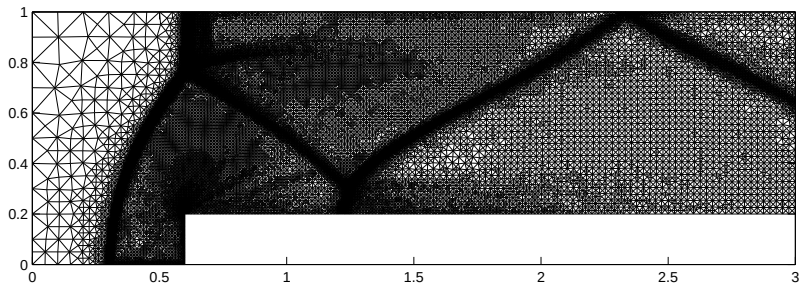

FIG. 6.5. *The mesh after 19 times adaptive refinements by 10% of the cells, the mesh has 33102 vertices.*

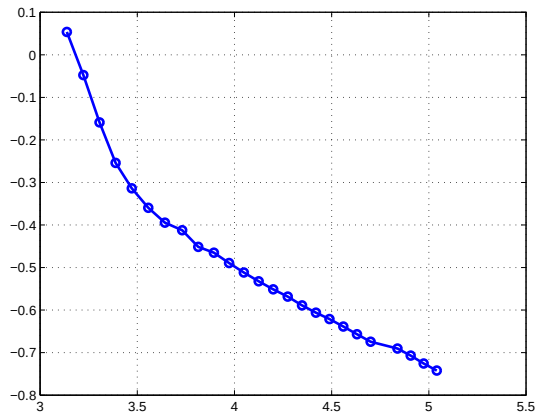

FIG. 6.6. *Log10 of the number of cells vs log10 of the L_2 norm of error indicator, $\|hR(\hat{u}_h)\|_K$, for the 2D supersonic flow around a cylinder. The mesh is refined adaptively 25 times with respect to the error indicator.*

We implemented an adaptive algorithm which is based on the residual of the system and the meshsize. The mesh refinement focuses on capturing the main characteristics of the compressible flows, such as shocks, contact discontinuities and rarefactions. The goal of the project is to implement a posteriori error estimation and adaptive methods for turbulent compressible flow using duality in 3D. The method would then be an extension of the corresponding

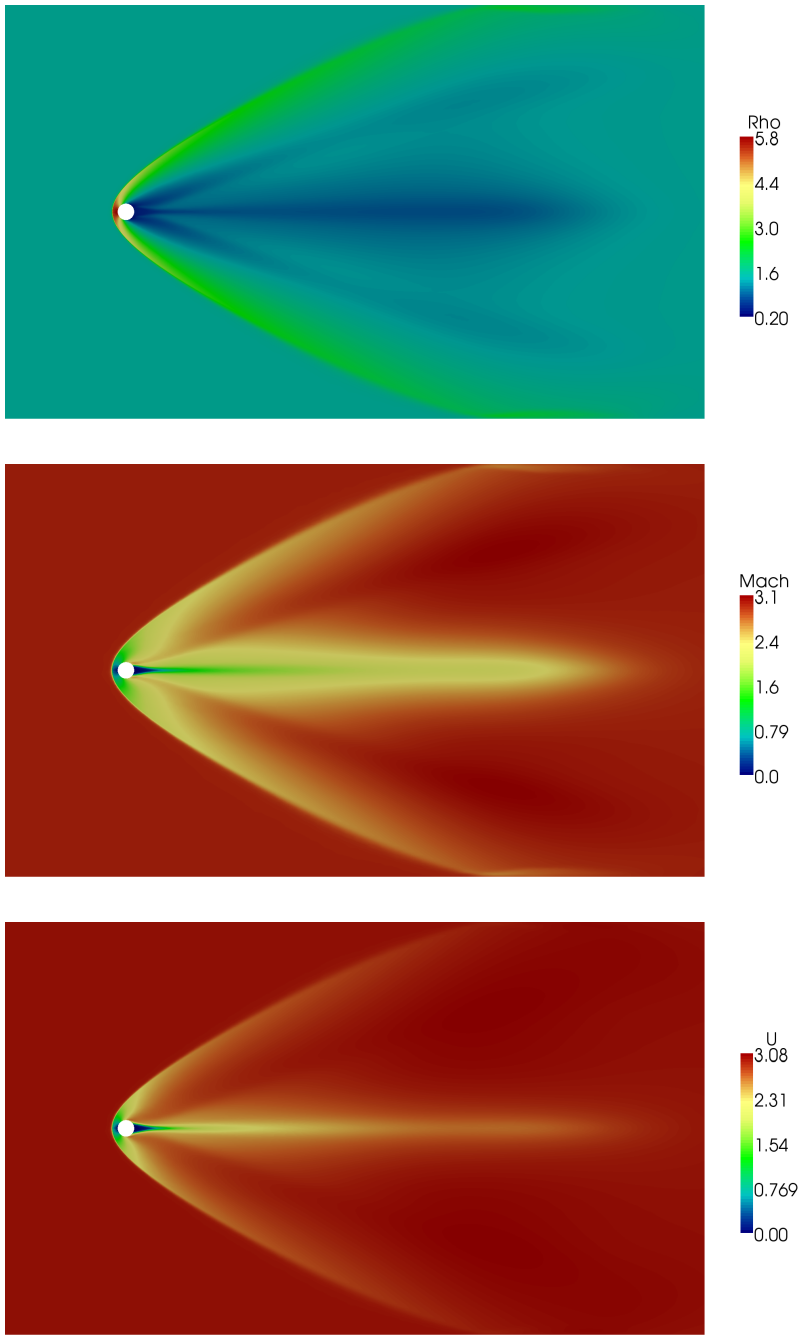

FIG. 6.7. *The colormap of the density, Mach number and velocity in x direction for supersonics flow around a cylinder in 2D at time $t = 0.4$. The figures are shown at interval $x \in [0, 1.7]$ and from the most finest mesh.*

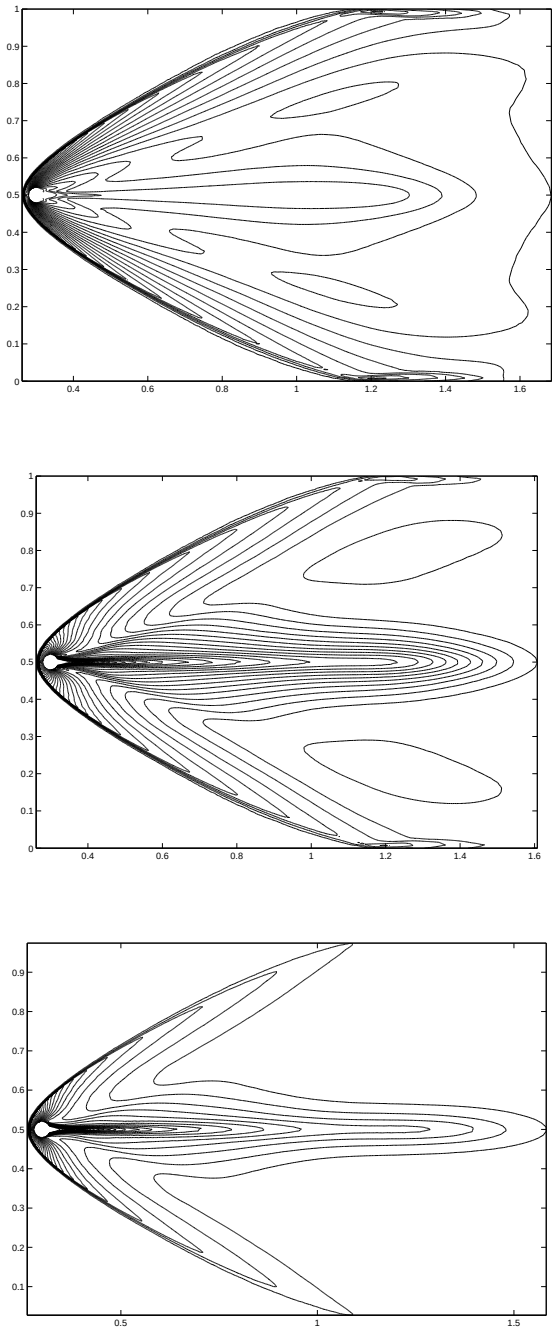

FIG. 6.8. *The 36 isolines of the density, Mach number for supersonics flow around a cylinder in 2D at time $t = 0.4$. The simulation is done until the shock reaches the channel boundaries and gets a bit reflection.*

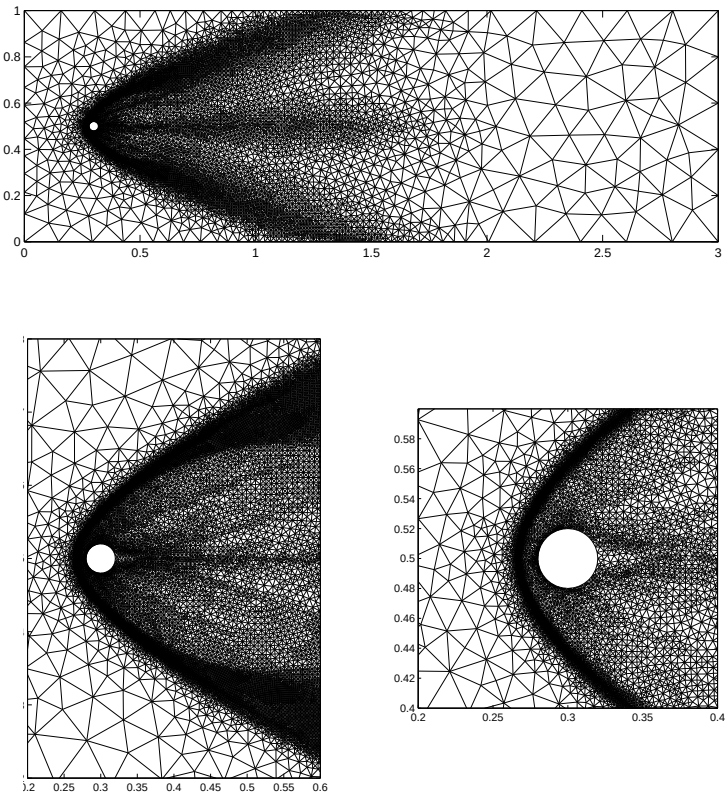

FIG. 6.9. *The mesh after 15 times (above) and 25 times (below figures) adaptive refinement by 10% of cells. The number of vertices are 11233 for the above mesh and 55125 for the below. The below figures are zoomed in close to the cylinder.*

method for incompressible turbulent flow by Hoffman and Johnson [7].

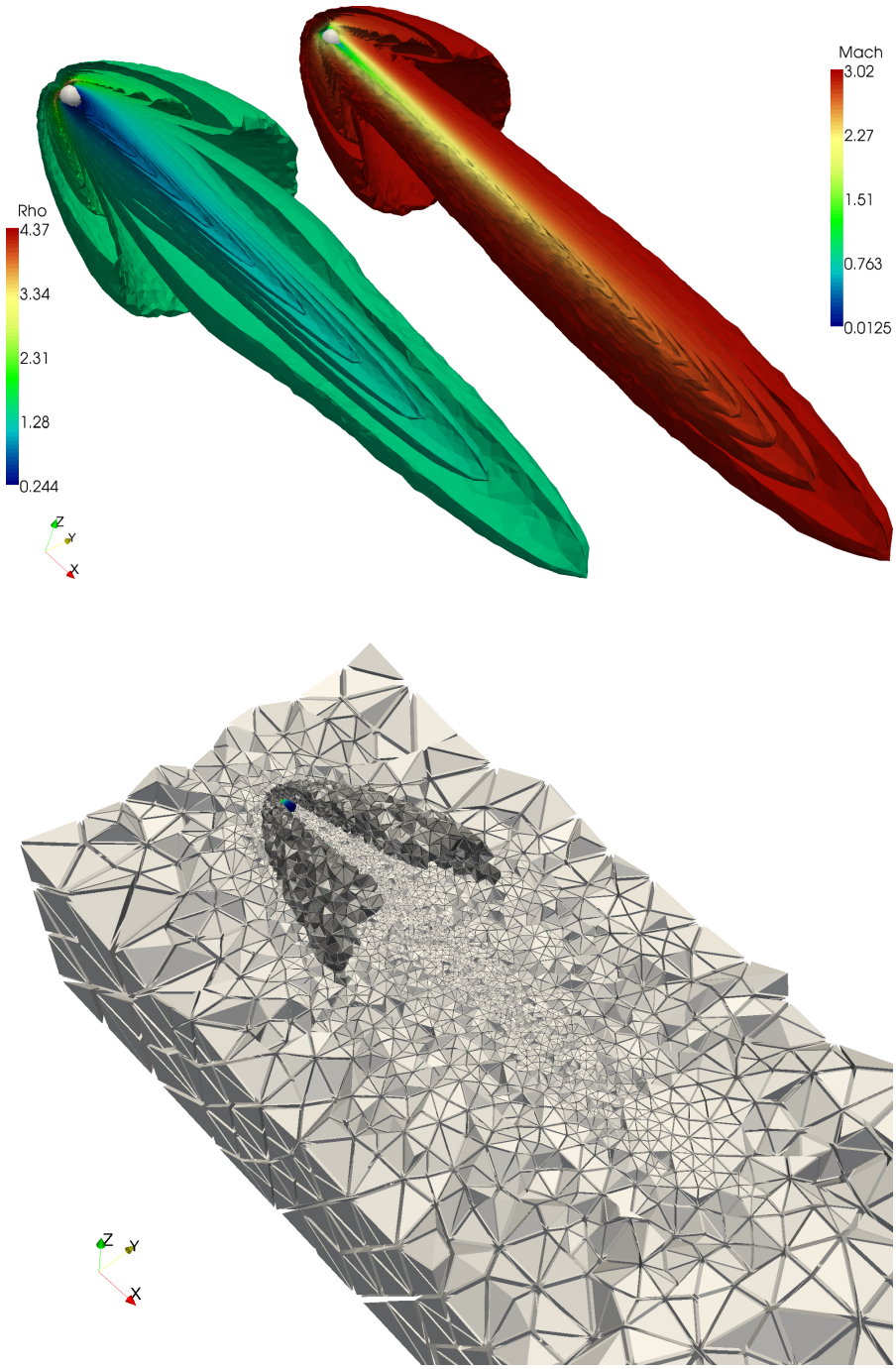

FIG. 6.10. *The contours of the density (left), Mach number (right) and mesh (lower) at time* $t = 0.338$. *The mesh is refined 11 times adaptive refinements by 5% of cells with 91529 vertices and 505792 cells. The adaptive algorithm focuses on discontinuities and shocks.*

REFERENCES

[1] M. Giles, M. Larson, M. Levenstam, and E. Süli, *Adaptive error control for finite element approximations of the lift and drag coefficients in viscous flow*, Technical Report NA-76/06, Oxford University Computing Laboratory, (1997).

[2] P. Hansbo, *Explicit streamline diffusion finite element methods for the compressible euler equations in conservation variables.*, Journal of Computational Physics, 109 (1993), pp. 274–288.

[3] ———, *A crank-nicolson type space-time finite element method for computing on moving meshes*, Journal of Computational Physics, 159 (2000), pp. 274–289.

[4] Johan Hoffman, *Computation of mean drag for bluff body problems using adaptive dns/les*, SIAM J. Sci. Comput., 27(1) (2005), pp. 184–207.

[5] ———, *Adaptive simulation of the turbulent flow past a sphere*, J. Fluid Mech., 568 (2006), pp. 77–88.

[6] ———, *Efficient computation of mean drag for the subcritical flow past a circular cylinder using general galerkin g2*, Int. J. Numer. Meth. Fluids, (accepted).

[7] Johan Hoffman and Claes Johnson, *Computational Turbulent Incompressible Flow: Applied Mathematics Body and Soul Vol 4*, Springer-Verlag Publishing, 2006.

[8] ———, *A new approach to computational turbulence modeling*, Comput. Methods Appl. Mech. Engrg., 195 (2006), pp. 2865–2880.

[9] ———, *Computational foundation of thermodynamics*, Comput. Methods Appl. Mech. Engrg., (to appear).

[10] T.J.R. Hughes and T.E. Tezduyar, *Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible euler equations*, Computer methods in applied mechanics and engineering, 45 (1984), pp. 217–284.

[11] V. John, *Slip with friction and penetration with resistance boundary conditions for the navier–stokes equations - numerical tests and aspects of the implementation*, J. Comp. Appl. Math, 147 (2002), pp. 287–300.

[12] C. Johnson and A. Szepessy, *Shock-capturing streamline diffusion finite element methods for nonlinear conservation laws*, Recent Developments in Computatioonal Fluid Dynamics, edited by T. J. R. Hughes and T. Tezduyar, AMD-Vol 95, 1988.

[13] C. Johnson, A. Szepessy, and P. Hansbo, *On the convergence of shock-capturing streamline diffusion finite element methods for hyperbolic conservation laws*, Mathematics of Computation, 54(189) (1990), pp. 107–129.

[14] J. Von Neumann and R.D. Richtmyer, *A method for the numerical calculation of hydrodynamic shocks.*, J. Appl. Phys., 21 (1950), pp. 232–237.

[15] T. E. Tezduyar and M. Senga, *Stabilization and shock-capturing parameters in supg formulation of compressible flows*, Computer Methods in Applied Mechanics and Engineering, 195 (2006), pp. 1621–1632.

[16] P. Woodward and P. Colella, *The numerical simulation of two-dimensional fluid flow with stong shocks.*, Journal of Computational Physics, 54 (1984), pp. 115–173.

Paper II

An Adaptive Finite Element Method for the Compressible Euler Equations

Murtazo Nazarov and Johan Hoffman
to be submitted, 2009.

AN ADAPTIVE FINITE ELEMENT METHOD FOR THE COMPRESSIBLE EULER EQUATIONS

MURTAZO NAZAROV AND JOHAN HOFFMAN

Abstract. We present an adaptive General Galerkin (G2) method for the compressible Euler equations, which is a finite element method with continuous piecewise linear approximation in space and time. The method is stabilized by componentwise weighted least squares stabilization of the convection terms, and residual based shock-capturing terms, resulting in a symmetric stabilization matrix in the discrete system. The a posteriori error estimation of a quantity of interest is based on a dual problem for the linearized equations. The adaptive algorithm is demonstrated numerically for the quantity of interest being the drag force on a body in the flow.

Key words. A posteriori error estimation, duality, General Galerkin method, stabilized finite element method, Euler equations, compressible flow

AMS subject classifications. 65M60

1. Introduction. In this paper we develop an adaptive finite element method for compressible flow, based on a posteriori error estimation of a quantity of interest, extending earlier work for incompressible flow [12, 17].

In fluid dynamics, the governing equations are the Euler/Navier-Stokes equations, expressing conservation of conservation of mass, momentum and energy. These equations fully describe the motion of Newtonian fluids, with the Euler equations describing the limit of vanishing viscosity in the Navier-Stokes equations. In the case of incompressible flow, the density is constant. In general, the solution to these equations varies rapidly in restricted regions, especially when the flow passes an object, where phenomena such as turbulence, wakes, shock waves, rarefactions and other discontinuities develop. These phenomena should be well resolved in order to get high accuracy in the computational simulations. In such cases, uniformly refining the mesh is not an option, since it would then impossible to resolve these phenomena with the computer power available today. For this reason, the development of efficient algorithms and numerical methods for a local adaptation of the mesh is one of the main subjects of computational science today. In particular, we want to design adaptive algorithms to construct an optimal mesh given a particular output of interest.

1.1. A posteriori error analysis. The adaptive algorithms in this paper are based on *a posteriori* error estimation, where the error is estimated from the computed solution. In most applications we are interested in ob-

taining an estimate of the error in some *quantity of interest* or *output*, rather than estimating the error itself. This approach minimizes the region which must be refined to obtain accuracy in the quantity of interest. The quantity of interest can be fluid forces such as drag and lift, stresses or fluxes. Once the bound of the error in an output of interest is estimated, it is possible to improve the approximation in order to reduce the error, which makes the method efficient and reliable. The idea of using duality analysis for a posteriori error estimation was first studied by Babuška and Miller for elliptic model problems [1, 2]. A more general framework for a systematic approach to a posteriori error estimation was later developed by Erikson and Johnson, Becker and Rannacher, with co-workers, [8, 6, 7, 3, 4].

For systems of conservation laws, a posteriori error analysis was earlier investigated by Johnson, Szepessy and Hansbo, [24, 25, 22, 23]. Adaptive discontinuous Galerkin methods for stationary compressible Euler equations in 2D were treated by Hartmann, Huston and Süli, and Larson and Barth [10, 20, 21, 26]. Burman studied a posteriori error estimation for time-dependent compressible Euler equations and performed numerical simulations in 2D [5].

In this paper we present a posteriori error estimates for time-dependent compressible flow in three dimensions, extending previous work on incompressible turbulent flow [19, 12]. We present numerical tests of the corresponding adaptive algorithm for the quantity of interest being the drag force. For convenience, we derive a posteriori error estimates for the variables density, velocity and pressure.

1.2. Turbulence modeling using adaptive finite elements. In this paper we interpret solutions to the Euler equations as viscosity solutions corresponding to the limit case of zero viscosity in the Navier-Stokes equations. A ***D****irect* ***N****umerical* ***S****imulation* (*DNS*) resolves all scales of the Euler/Navier-Stokes equations. In industrial applications such as aerodynamics, the *Reynolds number*, *Re*, of the fluid can be of order 10^6 and larger, so that full resolution of the smallest scales need more than 10^{18} grid points, which makes the computation impossible with the available computer power of today. Therefore, the traditional approach to get around DNS is to use turbulence modeling. For that the original equations are replaced by modified equations, which are obtained by averaging or filtering the original equations. In ***R****eynolds* ***A****veraged* ***N****avier* ***S****tokes* (*RANS*) the Navier-Stokes equations are replaced by time averaged flow equations, where the filter corresponds to a global average. In ***L****arge* ***E****ddy* ***S****imulation* (*LES*) the average is local in space and time. LES needs a more refined mesh than RANS, but much coarser than DNS. In LES instantaneous turbulent flow structures can be resolved, for more information

we refer to the book by Sagaut [28].

An alternative approach was proposed in [12, 18], in the sense of *weak solutions* adaptively resolving enough scales in space and time to approximate a chosen output of interest, as an application of the general methodology of adaptive stabilized Galerkin finite element methods with duality-based a posteriori error control. This method for turbulent flow, which we may refer to as *Adaptive DNS/LES*, resolves part of the flow by in a DNS and part of the flow is left unresolved in an LES with the stabilization of the method acting as a numerical type of turbulence model. This method is illustrated in a number of numerical papers: [11, 15, 13, 14, 19, 12].

This paper can be seen as an extension of Adaptive DNS/LES to the compressible Euler equations. We also use a General Galerkin method [16], and derive duality based a posteriori error estimates that extends to turbulent compressible flow. We check the method with a number of numerical examples in 2D and 3D including flow around a cylinder for different *Mach numbers*, M. The quantity of interest is here drag force, and the mesh is adaptively refined until we get convergence within some tolerance. In future work we will test the method for a set of benchmark problems of turbulent compressible flow.

1.3. Notation. In the following analysis we use the Einstein summation convention, the index based notation which is widely used in studies of partial differential equations, linear algebra and physics. For instance, the elements of a matrix are defined by $\mathbf{A} = a_{ij}$ and a vector by $\mathbf{b} = b_j$, where $i, j = 1, 2, \cdots, d$, where d is the space dimension. Then, the dot product $u \cdot v = \sum_{i=1}^{d} u_i v_i$ of vector functions u_i and v_i is written as $u_i v_i$, where the summation is understood by repeating the index i.

2. The Euler equations. The compressible *Euler equations* expresses conservation of mass, momentum and total energy for an inviscid fluid enclosed in a fixed (open) domain Ω in three-dimensional space $\mathbb{R}^3$ with boundary Γ over a time interval $[0, \hat{t}\,]$ with initial time zero and final time $\hat{t}$.

We seek the *density* ρ, *momentum* $m = \rho u$, with $u = (u_1, u_2, u_3)$ the *velocity*, and the *total energy* e as functions of $(x, t) \in \Omega \cup \Gamma \times [0, \hat{t}\,]$, where $x = (x_1, x_2, x_3)$ denotes the coordinates in $\mathbb{R}^3$ and u_i is the velocity in the x_i-direction. The Euler equations for $\hat{u} \equiv (\rho, m, e)$ read with $Q = \Omega \times I$ and

$I = (0, \hat{t}\,]$:

$$\begin{aligned} \dot{\rho} + \nabla \cdot (\rho u) &= 0 && \text{in } Q, \\ \dot{m}_i + \nabla \cdot (m_i u) + p_{,i} &= f_i && \text{in } Q, \quad i = 1, 2, 3, \\ \dot{e} + \nabla \cdot (eu + pu) &= 0 && \text{in } Q, \\ u \cdot n &= 0 && \text{on } \Gamma \times I, \\ \hat{u}(\cdot, 0) &= \hat{u}_0 && \text{in } \Omega, \end{aligned} \tag{2.1}$$

where $p = p(x,t)$ is the *pressure* of the fluid, $p_{,i} = \partial p/\partial x_i$ is the partial derivative with respect to x_i, the dot indicates differentiation with respect to time, n denotes the outward unit normal to Γ and $f = (f_1, f_2, f_3)$ is a given *volume force* (like gravity) acting on the fluid, and $\hat{u}_0 = \hat{u}_0(x)$ represents initial conditions. Further, the total energy is $e = k + \theta$, where $k = \rho|u|^2/2$ is the *kinetic energy*, with $|u|^2 \equiv u_1^2 + u_2^2 + u_3^2$, and $\theta = \rho T$ is the *internal energy* with T the *temperature* scaled so that $c_v = 1$, where c_v is the heat capacity under constant volume.

The boundary condition $u \cdot n = 0$ is a *slip boundary condition* requiring the normal velocity $u \cdot n$ to vanish corresponding to an impenetrable boundary with zero friction.

The number of unknowns including the pressure is six but there are only five equations in the Euler system (2.1), and so we close the system with the *state equation* of a *perfect gas*;

$$p = (\gamma - 1)\rho T, \tag{2.2}$$

expressing the pressure p as a function of density ρ and temperature T, where $\gamma = c_p$ is the *adiabatic index* with c_p the heat capacity under constant pressure, and $(\gamma - 1)$ is the *gas constant*.

For a perfect gas, the *speed of sound* c is given by $c^2 = \gamma(\gamma - 1)T$, and the *Mach number* is defined as $M = |u|/c$, with u the velocity of the gas.

3. The adjoint operator for the Euler equations. In this section we derive an adjoint operator for the compressible Euler equations. To simplify the derivation we rewrite the equations (2.1) in the variables: density, ρ, velocity, $u = (u_1, u_2, u_3)$, and pressure, p. Then, we linearize the modified Euler equations, for which we derive an adjoint operator.

3.1. The Euler equations in terms of density, velocity and pressure. By the relation $m_i = \rho u_i$ and using the density equation in (2.1), the conservation of momentum takes the following form

$$\rho \left(\dot{u}_i + u_j \frac{\partial u_i}{\partial x_j} \right) + \frac{\partial p}{\partial x_i} = f_i \quad \text{in } Q, \quad i = 1, 2, 3. \tag{3.1}$$

Now, we derive an equation for the pressure from the conservation of energy. For that, we first formulate an equation for the internal energy, $\theta = \rho T$, by inserting $e = k + \theta$ into the energy equation, and subtracting the equation for kinetic energy $k = \rho\frac{u_i u_i}{2}$ we get by multiplying the momentum equation by u,

$$\dot{\theta} + \frac{\partial}{\partial x_j}(u_j\theta) + p\frac{\partial u_j}{\partial x_j} = 0. \tag{3.2}$$

Using the state equation for the perfect gas, $p = (\gamma - 1)\rho T$, and by the relation $\theta = \rho T$ we obtain the pressure equation

$$\dot{p} + \frac{\partial}{\partial x_j}(u_j p) + (\gamma - 1)p\frac{\partial u_j}{\partial x_j} = 0. \tag{3.3}$$

Finally, by collecting the above equations in one system, we obtain a formulation of the Euler equation in terms of density, velocity and pressure: find $\hat{u} = (\rho, u_i, p) \in Q$, where $Q = \Omega \times I$ and $I = (0, \hat{t}]$, such that

$$\begin{aligned}
\dot{\rho} + \frac{\partial}{\partial x_j}(\rho u_j) &= 0 && \text{in } Q, \\
\rho\left(\dot{u}_i + u_j\frac{\partial u_i}{\partial x_j}\right) + \frac{\partial p}{\partial x_i} &= f_i && \text{in } Q, \quad i = 1, 2, 3, \\
\dot{p} + \frac{\partial}{\partial x_j}(u_j p) + (\gamma - 1)p\frac{\partial u_j}{\partial x_j} &= 0 && \text{in } Q, \\
u \cdot n &= 0 && \text{on } \Gamma \times I, \\
\hat{u}(\cdot, 0) &= \hat{u}_0 && \text{in } \Omega.
\end{aligned} \tag{3.4}$$

3.2. The linearized Euler equation. To illustrate the basic ideas, let us consider a nonlinear equation $f(x) = b$, where $f : \mathbb{R}^n \mapsto \mathbb{R}^n$, $x \in \mathbb{R}^n$. The *residual* of the computed solution $X \in \mathbb{R}^n$ is defined as

$$R(X) = f(X) - b, \tag{3.5}$$

or we can write

$$-R(X) = f(x) - f(X). \tag{3.6}$$

To get a relation between the error $e = x - X$ and the residual $R(X)$, we linearize the residual. One way is to use the mean value theorem for integrals and get the following relation

$$f(x) - f(X) = \left(\int_0 f'(sx + (1 - s)X)ds\right)(x - X) \tag{3.7}$$

where $f'(x)$ is a Fréchet derivative or Jacobian in finite dimensions.

Another way is to linearize the system about some state and then work with a perturbation equation. For this assume

$$x = X + \tilde{x}, \tag{3.8}$$

where $\tilde{x}$ is some small perturbation and X is an approximate solution which we get from a numerical method. Note that the perturbation in this case is equal to the error, $e = x - X$, between the approximate and exact solutions.

By inserting the last relation in the nonlinear equation, by Taylor expansion about X we get

$$\begin{aligned} f(X+\tilde{x}) = b, \quad &\Rightarrow \\ f(X) + f'(X)\tilde{x} + \mathcal{O}(\tilde{x}^2) = b, \quad &\Rightarrow \\ f'(X)\tilde{x} + \mathcal{O}(\tilde{x}^2) = -(f(X) - b), \quad &\Rightarrow \\ f'(X)\tilde{x} + \mathcal{O}(\tilde{x}^2) = -R(X), & \end{aligned} \tag{3.9}$$

where $f'(X)$ is a Jacobian matrix. If we drop higher order terms in $\tilde{x}$, the last equality is linear in the perturbation variable $\tilde{x}$ for which we can easily find an adjoint problem. In the analysis below we linearize the Euler equations (3.4), in this way.

Linearization of the Euler equations. Now, consider a linearization of the Euler equations, (3.4). For simplicity we assume there are no source terms in the equations. Let

$$\begin{aligned} \rho(x,t) &= \rho_h + \tilde{\rho}(x,t), \\ p(x,t) &= p_h + \tilde{p}(x,t), \\ u(x,t) &= u_h + \tilde{u}(x,t), \end{aligned} \tag{3.10}$$

where $\bar{\tilde{u}} = (\rho_h, p_h, u_h)$ is the background state, or the approximate solutions, we linearize about and $\tilde{\tilde{u}} = (\tilde{\rho}, \tilde{p}, \tilde{u})$ is the perturbation, or error. The idea is to derive equations for the perturbation in terms of the state. We insert (3.10) into (3.4), where we denote higher order terms in $\tilde{u}$ by *h.o.t.*, to get the

following equations for the perturbation variables:

$$\begin{aligned}
\dot{\tilde{\rho}} + \frac{\partial}{\partial x_j}(\rho_h \tilde{u}_j + \tilde{\rho} u_{h_j}) &= -R_\rho(\hat{u}_h) + h.o.t. && \text{in } Q,\\
\rho_h \left(\dot{\tilde{u}}_i + u_{h_j} \frac{\partial \tilde{u}_i}{\partial x_j} + \tilde{u}_j \frac{\partial u_{h_i}}{\partial x_j} \right) + & \\
\tilde{\rho} \left(\dot{u}_{h_i} + u_{h_j} \frac{\partial u_{h_i}}{\partial x_j} \right) + \frac{\partial \tilde{p}}{\partial x_i} &= -R_{u_i}(\hat{u}_h) + h.o.t. && \text{in } Q, \quad i = 1,2,3,\\
\dot{\tilde{p}} + \frac{\partial}{\partial x_j}(u_{h_j}\tilde{p} + \tilde{u}_j p_h) + & \\
(\gamma - 1) \left(p_h \frac{\partial \tilde{u}_j}{\partial x_j} + \tilde{p} \frac{\partial u_{h_j}}{\partial x_j} \right) &= -R_p(\hat{u}_h) + h.o.t. && \text{in } Q,\\
\tilde{u} \cdot n &= 0 && \text{on } \Gamma \times I,\\
\hat{\tilde{u}}(\cdot, 0) &= \hat{\tilde{u}}_0 && \text{in } \Omega,
\end{aligned} \tag{3.11}$$

where $R(\hat{u}_h) = (R_\rho(\hat{u}_h), R_{u_i}(\hat{u}_h), R_p(\hat{u}_h)$ are residuals of the Euler equations (3.4). The last system (3.11), when dropping higher order terms, is the linearized Euler equations (LEEs), which models the evolution of small disturbances such as numerical errors. In the analyzes below we drop the tilde on $\tilde{\hat{u}} = (\tilde{\rho}, \tilde{p}, \tilde{u})$ and use $\hat{u}(x,t) = (\rho(x,t), p(x,t), u(x,t))$ to denote the density, pressure and velocity in the LEEs.

REMARK 3.1. *Acoustic phenomena of compressible flow such as sound generation and propagation represent difficult problems in numerics. The acoustic fluctuations are very small compared to the background flow fields. Therefore their effect on the flow field is usually negligible. The LEEs are widely used in Computational Aeroacoustic.*

3.3. Relation of the dual problem to error estimation. From the definition of an adjoint operator we know that it is a linear transformation defined in a Hilbert space, H.

DEFINITION 3.2. *An operator A is said to be adjoint to the bounded linear operator A if*

$$(A(u), v) = (u, A^*(v)), \quad \forall u, v \in H.$$

LEMMA 3.3. *There exists an unique adjoint operator.*

Proof. Indeed, assume there are two adjoint operators A^* and $\hat{A}$ for a linear operator A. Then, for $\forall u, v \in H$ we can write

$$(A(u), v) = (u, A^*(v))$$

and

$$(A(u), v) = (u, \hat{A}(v)).$$

By subtracting these two relations and by the linearity of the operators we have

$$0 = (u, (A^*(v) - \hat{A}(v)), \Rightarrow \quad 0 = (u, (A^* - \hat{A})(v)), \Rightarrow$$
$$(A^* - \hat{A})(v) = 0, \quad \forall v \in H.$$

Here we get a zero operator acting to v, which proves the above statement about the uniqueness of the adjoint operator for linear problems.
□

For non-linear problems we can define an adjoint operator for the corresponding linearized problem.

As in the previous sections we are interested in the relation between the residual $R(X)$ and the error $e = x - X$ for the nonlinear *primal problem* $A(x) = f(x) = b$. Below we use this relation to estimate the error in the problem. We denote the linearization of this problem by $\underline{\mathrm{A}}(e) = f'(X)e$. Then

$$\underline{\mathrm{A}}(e) = f(x) - f(X) + \mathcal{O}(\tilde{x}^2), \tag{3.12}$$

and the relation between the error and residual can be written as

$$\underline{\mathrm{A}}(e) = -R(X) + \mathcal{O}(\tilde{x}^2). \tag{3.13}$$

Recall the definition of the adjoint operator, we define an adjoint A^* to the linear operator $\underline{\mathrm{A}}$ as follows:

$$(\underline{\mathrm{A}}(e), \phi) = (e, A^*(\phi)), \quad \forall e, \phi \in H. \tag{3.14}$$

The function ϕ in the last equality is also called *Generalized Green's function.* Using the adjoint operator A^* in (3.14), we define the following *dual problem* for the variable ϕ: find $\phi \in H$ such that

$$A^*(\phi) = \psi, \tag{3.15}$$

where $\psi \in H$ defines *a quantity of interest* $M(\hat{u})$ of the problem by $M(\hat{u}) = (\hat{u}, \psi)$. The notation $(.,.)$ denotes an inner product in the Hilbert space H. The quantity $M(\hat{u})$ can define an error at some point, the error in average

over some subset, or some norm of the error. In the computations in this paper, we consider the drag force of the compressible flow around some object as the quantity of interest.

LEMMA 3.4. *The error representation for the quantity of interest, $M(e)$, has the following relation to the primal residual, $R(X)$, and dual solution, ϕ:*

$$M(e) = (-R(X) + \mathcal{O}(e^2), \phi). \tag{3.16}$$

Proof. Using the definition of the adjoint operator, (3.14), and the dual problem for the quantity of interest ϕ, (3.15), we can have the following error representation for the quantity of interest $M(e)$:

$$\begin{aligned} M(e) = (e, \psi) = (e, A^*(\phi)) = (\underline{\mathrm{A}}(e), \phi) = \\ (-R(X) + \mathcal{O}(e^2), \phi). \end{aligned} \tag{3.17}$$

□

In equation (3.16) the residual of the primal problem, $R(X)$ is given from the approximate solution X, and the solution to the linearized dual problem ϕ we can approximate by computational solution of the dual problem.

3.4. Dual problem for LLEs. From the definition of the adjoint operator one see that the adjoint for the linearized Euler equations is obtained by integration by parts. A straight forward way of calculating adjoint operators is to take the inner product of the linear operator acting to the error, $\underline{\mathrm{A}}(e)$, with the dual solution ϕ and integrate by parts. For this purpose we start by introducing the following variables:

$$\begin{aligned} e_\rho &= \rho - \rho_h, \\ e_{u_i} &= u_i - u_{h_i}, \\ e_p &= p - p_h, \end{aligned} \tag{3.18}$$

where we assume (ρ, u_i, p) to be functions in $L_2(I; H^1(\Omega) \times [H^1(\Omega)]^3 \times H^1(\Omega))$ with also time derivatives in $L_2(I; H^1(\Omega) \times [H^1(\Omega)]^3 \times H^1(\Omega))$ and ρ_h, u_{h_i}, p_h are finite element approximations in the same spaces, where we use standard notation from functional analysis, see e.g. [9]. Then, the relation between (3.18) take the following form:

$$
\begin{aligned}
\dot{e}_\rho + \frac{\partial}{\partial x_j}(\rho e_{u_j} + e_\rho u_j) &= -R_\rho(\hat{u}_h) + h.o.t., \\
\rho\left(\dot{e}_{u_i} + u_j\frac{\partial e_{u_i}}{\partial x_j} + e_{u_j}\frac{\partial u_i}{\partial x_j}\right) + e_\rho\left(\dot{u}_i + u_j\frac{\partial u_i}{\partial x_j}\right) + \frac{\partial e_p}{\partial x_i} &= -R_{u_i}(\hat{u}_h) + h.o.t., \\
\dot{e}_p + \frac{\partial}{\partial x_j}(u_j e_p + e_{u_j} p) + (\gamma-1)\left(p\frac{\partial e_{u_j}}{\partial x_j} + e_p\frac{\partial u_j}{\partial x_j}\right) &= -R_p(\hat{u}_h) + h.o.t.,
\end{aligned}
\tag{3.19}
$$

where (ρ, u_i, p) is a given state, $\hat{e} = (e_\rho, e_{u_i}, e_p)$ and $i = 1, 2, 3$. By dropping the $h.o.t.$ we get the linearized Euler equations (LLEs).

THEOREM 3.5. *Let $\hat{\phi} = (\phi_\rho, \phi_{u_i}, \phi_p) \in L_2(I; H^1(\Omega) \times [H^1(\Omega)]^3 \times H^1(\Omega))$ in $Q = \Omega \times I$ be a dual density, velocity and pressure. Let $\dot{\hat{\phi}} = (\phi_\rho, \phi_{u_i}, \phi_p) \in L_2(I; L_2(\Omega) \times [L_2(\Omega)]3 \times L_2(\Omega))$ and $n = (n_1, n_2, n_3)$ be the outward unit surface normal to the boundary Γ. Assume the normal component of the dual velocity vanishes on the boundary Γ: $\phi_u \cdot n = 0$, and $\hat{\phi}(\cdot, T) = 0$ in Ω. We also assume that the initial data for the approximate solution is the same as the for exact solution, $e(\cdot, 0) = \hat{u}(\cdot, 0) - \hat{u}_h(\cdot, 0) = 0$. Then, the dual problem for the linearized Euler equation is*

$$
\begin{aligned}
-\dot{\phi}_\rho - u_j\frac{\partial \phi_\rho}{\partial x_j} + \left(\dot{u}_i + u_j\frac{\partial u_i}{\partial x_j}\right)\phi_{u_i} &= \psi_\rho, && \text{in } Q, \\
-(\dot{\phi_{u_i}}\rho) - \frac{\partial}{\partial x_j}(\rho u_j \phi_{u_i}) + \rho\phi_{u_j}\frac{\partial u_j}{\partial x_i} - p\frac{\partial \phi_p}{\partial x_i} & \\
-(\gamma-1)\frac{\partial}{\partial x_i}(p\phi_p) - \rho\frac{\partial \phi_\rho}{\partial x_i} &= \psi_{u_i}, && \text{in } Q, i = 1, 2, 3, \\
-\dot{\phi}_p - u_j\frac{\partial \phi_p}{\partial x_j} + (\gamma-1)\phi_p\frac{\partial u_j}{\partial x_j} - \frac{\partial \phi_{u_i}}{\partial x_i} &= \psi_p && \text{in } Q, \\
\phi_u \cdot n &= 0 && \text{on } \Gamma \times I, \\
\hat{\phi}(\cdot, T) &= 0 && \text{in } \Omega,
\end{aligned}
\tag{3.20}
$$

where d is the space dimension, $\psi = (\psi_\rho, \psi_{u_i}, \psi_p) \in L_2(I; L_2(\Omega) \times [L_2(\Omega)]^3 \times L_2(\Omega))$ is data that defines a quantity of interest and T is the final computational time for the primal equation.

Proof. In a standard way we multiply the dual solution, $\hat{\phi} = (\phi_\rho, \phi_{u_i}, \phi_p)$, to the error equation equation, (3.19), and integrate every term by parts, to obtain the following relations, note that the boundary terms in time vanishes because of the assumptions of the theorem. We here let $(v, w) = \int_0^T \int_\Omega v \cdot w dx dt$, denote the $L_2(Q)$-inner product:

The error equation for the density:

$$\begin{aligned}\mathrm{I} = &\left(\dot{e_\rho} + \frac{\partial}{\partial x_j}(\rho e_{u_j} + e_\rho u_j), \phi_\rho\right) = \\ &- (e_\rho, \dot{\phi_\rho}) - \left(e_{u_j}, \rho\frac{\partial}{\partial x_j}\phi_\rho\right) + \int_0^T \int_\Gamma \rho e_{u_j} n_j \phi_\rho \mathrm{d}S\mathrm{d}t \\ &- \left(e_\rho, u_j\frac{\partial}{\partial x_j}\phi_\rho\right) + \int_0^T \int_\Gamma e_\rho u_j n_j \phi_\rho \mathrm{d}S\mathrm{d}t.\end{aligned} \tag{3.21}$$

Here, the boundary terms vanish, since the normal primal velocity is zero from the boundary condition in (2.1).

The error equation for the velocity:

$$\begin{aligned}\mathrm{II} = &\left[\rho\left(\dot{e_{u_i}} + u_j\frac{\partial e_{u_i}}{\partial x_j} + e_{u_j}\frac{\partial u_i}{\partial x_j}\right) + e_\rho\left(\dot{u}_i + u_j\frac{\partial u_i}{\partial x_j}\right) + \frac{\partial e_p}{\partial x_i}, \phi_{u_i}\right] = \\ &- (e_{u_i}, (\dot{\phi_{u_i}\rho})) - \left(e_{u_i}, \frac{\partial}{\partial x_j}(\rho u_j \phi_{u_i})\right) + \int_0^T \int_\Gamma e_{u_i} n_j \rho u_j \phi_{u_i} \mathrm{d}S\mathrm{d}t \\ &+ \left(e_{u_j}, \rho\phi_{u_i}\frac{\partial u_i}{\partial x_j}\right) + \left(e_\rho, \left(\dot{u}_i + u_j\frac{\partial u_i}{\partial x_j}\right)\phi_{u_i}\right) \\ &- \left(e_p, \frac{\partial \phi_{u_i}}{\partial x_i}\right) + \int_0^T \int_\Gamma \phi_{u_i} n_i e_p \mathrm{d}S\mathrm{d}t.\end{aligned} \tag{3.22}$$

Here, the first boundary integral vanishes, because of the boundary condition in (2.1). The second boundary integral vanishes by the theorem assumption, $\phi_{u_i} n_i = 0$.

The error equation for the pressure:

$$\begin{aligned}\mathrm{III} = &\left[\dot{e_p} + \frac{\partial}{\partial x_j}(u_j e_p + e_{u_j} p) + (\gamma - 1)\left(p\frac{\partial e_{u_j}}{\partial x_j} + e_p\frac{\partial u_j}{\partial x_j}\right), \phi_p\right] = \\ &- (e_p, \dot{\phi_p}) - \left(e_p, u_j\frac{\partial \phi_p}{\partial x_j}\right) + \int_0^T \int_\Gamma u_j e_p n_j \phi_p \mathrm{d}S\mathrm{d}t \\ &- \left(e_{u_j}, p\frac{\partial \phi_p}{\partial x_j}\right) + \int_0^T \int_\Gamma e_{u_j} p n_j \phi_p \mathrm{d}S\mathrm{d}t - \\ &- (\gamma - 1)\left(e_{u_j}, \frac{\partial}{\partial x_j}(p\phi_p)\right) + (\gamma - 1)\int_0^T \int_\Gamma e_{u_j} n_j p \phi_p \mathrm{d}S\mathrm{d}t \\ &+ (\gamma - 1)\left(e_p, \phi_p\frac{\partial u_j}{\partial x_j}\right).\end{aligned} \tag{3.23}$$

The boundary integral in the last equation, since $u_j n_j = 0$, $u_{h_j} n_j = 0$, $\Rightarrow (u_j - u_{h_j}) n_j = e_{u_j} n_j = 0$.

Then, by collecting terms for each error variable, e_ρ, e_{u_i}, e_p, we get the above linearized dual problem, (3.20),

$$\mathrm{I} + \mathrm{II} + \mathrm{III} = (e_\rho, A^*_{\phi_\rho}(\hat{\phi})) + (e_{u_i}, A^*_{\phi_{u_i}}(\hat{\phi})) + (e_p, A^*_{\phi_p}(\hat{\phi})), \tag{3.24}$$

for the compressible Euler equation, where $A^*(\hat{\phi}) = \left(A^*_{\phi_\rho}(\hat{\phi}), A^*_{\phi_{u_i}}(\hat{\phi}), A^*_{\phi_p}(\hat{\phi})\right)$ is the adjoint operator for the linearized Euler equations, (3.11).

In the equation (3.22), we used the swapping property of the Einstein index notation, e.g.:

$$\left(e_{u_j}, \phi_{u_i} \frac{\partial u_i}{\partial x_j}\right) \equiv \left(e_{u_i}, \phi_{u_j} \frac{\partial u_j}{\partial x_i}\right). \tag{3.25}$$

By lemma 3.3 we know that this dual operator is unique for the LLEs. □

Remark 3.6. *In the analysis above, we get the linearized dual problem from the continuous problem, (3.4). An alternative way, which is often used for error analysis, is to derive a dual problem from the discrete problem. Meaning that, the equations are first discretized by some numerical method, then the corresponding dual problem is found.*

4. A General Galerkin (G2) finite element method. In time, the trial functions are continuous piecewise linear and the test functions are piecewise constant, and in space both test functions and trial functions are continuous piecewise linear. This choice of the finite element functions is called a continuous Galerkin cG(1)cG(1) method and we refer to it as a General Galerkin or G2 method.

Let $0 = t_0 < t_1 < ... < t_N = \hat{t}$, be a sequence of discrete time steps with associated time intervals $I_n = (t_{n-1}, t_n]$ of length $\Delta t_n = t_n - t_{n-1}$ and space-time slabs $S_n = \Omega \times I_n$, and let $W_h \subset H^1(\Omega)$ be a finite element space consisting of continuous piecewise linear functions on a fixed mesh $\mathcal{T}_h = \{K\}$ of mesh size $h(x) < 1$, with elements K. Further, let $W_{h,0}$ be the space of vector functions in W^3_h satisfying the slip boundary condition, that is $W_{h,0} = \{w \in W^3_h : w \cdot n = 0 \text{ on } \Gamma\}$.

We solve the primal problem for density, momentum and energy, while we solve the dual problem for the dual variables: density, velocity and pressure. The following subsections show the finite element discretization of the equations (2.1) and (3.20) using G2.

4.1. G2 for the primal compressible Euler equations. We now seek functions $\hat{u}_h = (\rho_h, m_h, e_h)$, continuous piecewise linear in space and time. The cG(1)cG(1) method for the compressible Euler equations, here without source terms for simplicity, reads: For $n = 1, ..., N$, find $\hat{u}_h^n = (\rho_n, m_n, e_n) \equiv (\rho_h(t_n), m_h(t_n), e_h(t_n))$ with $\hat{u}_h^n \in V_h \equiv W_h \times W_{h,0} \times W_h$, such that

$$(\dot{\rho}_n, v_n^\rho) - (u_n\bar{\rho}_n, \nabla v_n^\rho) + SD_\rho(\rho_h; v_n^\rho) = 0, \tag{4.1}$$

$$(\dot{m}_{n_j}, v_{n_j}^m) - (u_n\bar{m}_{n_j}, \nabla v_{n_j}^m) - (p_n, \nabla \cdot v_{n_j}^m) + SD_{m_j}(m_h; v_{n_j}^m) = 0, \quad j = 1, 2, 3 \tag{4.2}$$

$$(\dot{e}_n, v_n^e) - (u_n\bar{e}_n, \nabla v_n^e) + (\nabla \cdot (u_n p_n), v_n^e) + SD_e(e_h; v_n^e) = 0, \tag{4.3}$$

for all test functions $\hat{v}_n = (v_n^\rho, v_{n_j}^m, v_n^e) \in V_h$, where

$$(v, w) = \sum_{K \in \mathcal{T}_h} \int_K v \cdot w \, dx,$$

and

$$\dot{\rho}_n = \frac{\rho_n - \rho_{n-1}}{\Delta t_n}, \quad \dot{m}_n = \frac{m_n - m_{n-1}}{\Delta t_n}, \quad \dot{e}_n = \frac{e_n - e_{n-1}}{\Delta t_n},$$

$$\bar{\rho}_n = \frac{1}{2}(\rho_n + \rho_{n-1}), \quad \bar{m}_n = \frac{1}{2}(m_n + m_{n-1}), \quad \bar{e}_n = \frac{1}{2}(e_n + e_{n-1}),$$

where we define u_n, p_n and T_n to be finite element functions in $W_{h,0}$, W_h and W_h respectively, which are defined by their nodal values given by

$$\begin{aligned} u_n(N_i) &= \bar{m}_n(N_i)/\bar{\rho}_n(N_i), \\ p_n(N_i) &= (\gamma - 1)\bar{\rho}_n(N_i)T_n(N_i), \\ T_n(N_i) &= \bar{e}_n(N_i)/\bar{\rho}_n(N_i) - |u_n(N_i)|^2/2 \end{aligned}$$

for all nodes N_i in the mesh $\mathcal{T}_h$,

We define a componentwise stabilization in the form of a weighted least squares stabilization of the convection terms together with residual based shock-capturing,

$$SD_\rho(\rho_h; v_n^\rho) = (\delta u_n \cdot \nabla\bar{\rho}_n, u_n \cdot \nabla v_n^\rho) + (\hat{\nu}_\rho \nabla\bar{\rho}_n, \nabla v_n^\rho), \tag{4.4}$$

$$SD_{m_j}(m_h; v_{n_j}^m) = (\delta u_n \cdot \nabla\bar{m}_{n_j}, u_n \cdot \nabla v_{n_j}^m) + (\hat{\nu}_m \nabla\bar{m}_{n_j}, \nabla v_{n_j}^m), j = 1, 2, 3 \tag{4.5}$$

$$SD_e(e_h; v_n^e) = (\delta u_n \cdot \nabla\bar{e}_n, u_n \cdot \nabla v_n^e) + (\hat{\nu}_e \nabla\bar{e}_n, \nabla v_n^e), \tag{4.6}$$

where $\delta = C_\delta(\Delta t_n^{-2} + |u_n|^2 h^{-2})^{-1/2}$ and $\hat{\nu}_\alpha = \max\left(C_\alpha \dfrac{|R_\alpha(\hat{u})|}{|\alpha_h|} h^2, C_h h^{3/2}\right)$, $(\alpha = \rho, m, e$ and $\alpha_h = \rho_h, m_h + \epsilon, e_h)$, with constants C_δ, C_α and $C_h \sim \frac{U}{\sqrt{L}}$,

where U and L are characteristic velocity and length scales, and we define the strong residuals by

$$R_\rho(\hat{u}_h) = \dot{\rho}_n + \nabla \cdot (\bar{\rho}_n u_n), \tag{4.7}$$
$$R_{m_j}(\hat{u}_h) = (\dot{m}_n)_j + \nabla \cdot ((\bar{m}_n)_j u_n) + \nabla p_n, \quad j = 1, 2, 3, \tag{4.8}$$
$$R_e(\hat{u}_h) = \dot{e}_n + \nabla \cdot (\bar{e}_n u_n + p_n u_n), \tag{4.9}$$

for $t \in I_n$, and with $\epsilon > 0$ a small safety factor.

Then, the cG(1)cG(1) method (4.1) - (4.3) can be written in the following form:

$$(R_\rho(\hat{u}_h), v_n^\rho) + SD_\rho(\rho_h; v_n^\rho) = 0, \tag{4.10}$$
$$(R_{m_j}(\hat{u}_h), v_n^m) + SD_{m_j}(m_h; v_n^m) = 0, \quad j = 1, 2, 3, \tag{4.11}$$
$$(R_e(\hat{u}_h), v_n^e) + SD_e(e_h; v_n^e) = 0. \tag{4.12}$$

The common approach of stabilizing finite element methods is using the complete system, see [29], where the full residual is used for convective and shock-capturing stabilization. Here, we use only a least squares stabilization of the convection term and the shock capturing term is based on the individual residuals of each equations. With this choice, the method conserves mass, momentum and energy, since the stabilization only contains derivatives of the test functions. Moreover, it results in a symmetric stabilization matrix in the discrete system, [16].

We solve the discretized equations by a fixed-point iteration with velocity given from the previous iteration. The iteration starts with u_0, and each iteration calculates u_{m+1} for a given u_m. The resulting linear system is solved with GMRES. With this method the time step Δt_n is given by a CFL-condition, with typically $\Delta t_n \sim \min_{E_j \in \mathcal{T}_h}(h/|u_n|)_{E_j}$, for all elements E_j in the mesh $\mathcal{T}_h$.

4.2. G2 for the linearized dual Euler equations. Similar to the primal problem, we seek functions $\hat{\phi}_h^n = (\phi_{\rho_n}, \phi_{u_{n_i}}, \phi_{p_n}) = (\phi_\rho(t_n), \phi_{u_i}(t_n), \phi_p(t_n))$, continuous piecewise linear in space and time, and as above $n = 1, \cdots, N$ and $i = 1, 2, 3$. Then, cG(1)cG(1) for the linearized dual Euler equations become: find $\hat{\phi}_h^n \in V_h \equiv W_h \times W_{h,0} \times W_h$, such that

$$-(\dot{\phi}_{\rho_n}, v_n^\rho) - \left(u_j^n \frac{\partial \bar{\phi}_{\rho_n}}{\partial x_j}, v_n^\rho\right) + \left(\left(\dot{u}_j^n + u_j^n \frac{\partial u_i^n}{\partial x_j}\right)\bar{\phi}_{u_{n_i}}, v_n^\rho\right) + \nu_h \left(\frac{\partial \bar{\phi}_{\rho_n}}{\partial x_j}, \frac{\partial v_n^\rho}{\partial x_j}\right) = (\psi_{\rho_n}, v_n^\rho), \tag{4.13}$$

$$-\left(\dot{\phi}_{u_{n_i}}\rho^n, v_{n_i}^u\right) + \left(\rho u_j^n \bar{\phi}_{u_{n_i}}, \frac{\partial v_{n_i}^u}{\partial x_j}\right) + \left(\rho^n \bar{\phi}_{u_{n_j}} \frac{\partial u_j^n}{\partial x_i}, v_{n_i}^u\right) - \left(p^n \frac{\partial \bar{\phi}_{p_n}}{\partial x_i}, v_{n_i}^u\right) - \left(\rho^n \frac{\partial \bar{\phi}_{\rho_n}}{\partial x_i}, v_{n_i}^u\right) + (\gamma - 1)\left(p^n \bar{\phi}_{p_n}, \frac{\partial v_{n_i}^u}{\partial x_i}\right) + \nu_h \left(\frac{\partial \bar{\phi}_{u_{n_i}}}{\partial x_j}, \frac{\partial v_{n_i}^u}{\partial x_j}\right) = (\psi_{u_{n_i}}, v_{n_i}^u), \tag{4.14}$$

$$-(\dot{\phi}_{p_n}, v_n^p) - \left(u_j^n \frac{\partial \bar{\phi}_{p_n}}{\partial x_j}, v_n^p\right) + (\gamma - 1)\left(\bar{\phi}_{p_n} \frac{\partial u_j^n}{\partial x_j}, v_n^p\right) + \nu_h \left(\frac{\partial \bar{\phi}_{p_n}}{\partial x_j}, \frac{\partial v_n^p}{\partial x_j}\right) + \left(\phi_{u_{n_i}}, \frac{\partial v_n^p}{\partial x_i}\right) = (\psi_{p_n}, v_n^p), \tag{4.15}$$

for all test functions $\hat{v}_n = (v_n^\rho, v_{n_i}^u, v_n^p) \in V_h$, where $\hat{\psi}_n = (\psi_n^\rho, \psi_{n_i}^u, \psi_n^p) \in V_h$ defines a quantity of interest and ν_h is artificial viscosity. In our computations we use artificial viscosity $\nu_h = h$. As above

$$(v, w) = \sum_{K \in \mathcal{T}_h} \int_K v \cdot w \, dx,$$

dot "˙" denotes the time derivative in I_n, and "¯" denotes the mean value.

The function $\hat{u}^n = (\rho^n, u^n, p_n) \in V_h$ is the computed primal solution at time t_n, where the corresponding time derivatives are defined as $\dot{u}^n \equiv (u^n - u^{n-1})/\Delta t_n$.

4.3. Slip boundary conditions. In fluid dynamics slip boundary conditions are typically used for inviscid compressible flow, corresponds to the normal velocity on the boundary being zero.

In our computations, we implement the slip boundary condition strongly. By "strongly" we here mean an implementation of the boundary condition after assembling the stiffness matrix and the load vector, whereas a "weak" implementation involves adding boundary integrals to the variational formulation. The row of the matrix and load vector corresponds to a boundary vertex is found and replaced by a new row according to the boundary condition.

The idea is as follows: Initially, the test function v is expressed in the Cartesian standard basis (e_1, e_2, e_3). Now, the test function is mapped locally to normal-tangent coordinates with the basis (n, τ_1, τ_2), where $n = (n_1, n_2, n_3)$ is the normal, and $\tau_1 = (\tau_{11}, \tau_{12}, \tau_{13})$, $\tau_2 = (\tau_{21}, \tau_{22}, \tau_{23})$ are tangents for each node on the boundary. This allows us to let the normal direction be constrained and the tangent directions to be free:

$$v = (v \cdot n)n + (v \cdot \tau_1)\tau_1 + (v \cdot \tau_2)\tau_2.$$

For the matrix and vector this means that the rows corresponding to the boundary need to be multiplied with n, τ_1, τ_2, respectively, and then the normal velocity should be put 0. For detailed information, we refer to [16].

5. An a posteriori error estimate. We introduce the following notation which we use for the a posteriori error analysis:

$$\begin{aligned}
&(u, w)_K = \int_K v \cdot w \mathrm{d}x, \quad (u, w)_{\partial K} = \int_{\partial K} v \cdot w \mathrm{d}s, \\
&\|v\|_K = \|v\|_{L_2(K)} = (v, v)_K^{1/2}, \\
&|v|_K = (\|v_1\|_K, \|v_2\|_K, \|v_3\|_K), \\
&I = \bigcup_{n=1}^{N} I_n, \quad \mathcal{T}_h = \bigcup_{K \in \mathcal{T}_h} K.
\end{aligned} \tag{5.1}$$

Assume $\hat{u}_h = (\rho_n, u_{n_i}, p_n) \in V_h \equiv W_h \times W_{h0} \times W_h$ for $i = 1, 2, 3$ be an approximate solution of the equation (3.4). Then, the strong residuals for the compressible Euler equations, $R(\hat{u}_h) = (R_\rho(\hat{u}_h), R_{u_i}(\hat{u}_h), R_p(\hat{u}_h))$ in terms of density, velocity and pressure takes the following form

$$R_\rho(\hat{u}_h) = \dot{\rho}_n + \frac{\partial}{\partial x_j}(\rho_n u_{n_j}), \tag{5.2}$$

$$R_{u_i}(\hat{u}_h) = \rho_n \left(\dot{u}_{n_i} + u_{n_j} \frac{\partial u_{n_i}}{\partial x_j} \right) + \frac{\partial p_n}{\partial x_i}, \tag{5.3}$$

$$R_p(\hat{u}_h) = \dot{p}_n + \frac{\partial}{\partial x_j}(u_{n_j} p_n) + (\gamma - 1) p \frac{\partial u_{n_j}}{\partial x_j}, \tag{5.4}$$

where $i = 1, 2, 3$, and we define time derivatives of velocity and pressure as in section 4.2 .

Then, the following theorem expresses an error representation for the compressible Euler equations.

THEOREM 5.1. *Assume* $\hat{\phi} = (\phi_\rho, \phi_{u_i}, \phi_p) \in \mathcal{C}(I; H^1(\Omega) \times [H^1(\Omega)]^3 \times H^1(\Omega))$ *is an exact solution of the linearized dual problem (3.20) with* $\hat{\phi} \in L_2(I; L_2(\Omega \times L_2(\Omega)^3 \times L_2(\Omega))$. *Let* $\psi = (\psi_\rho, \psi_{u_i}, \psi_p) \in L_2(I; L_2(\Omega) \times [L_2(\Omega)]^3 \times L_2(\Omega))$ *define a quantity of interest* $M(\hat{u}) = (\hat{u}, \hat{\psi})$ *and* $\pi_h\hat{\phi} = (\pi_h\phi_\rho, \pi_h\phi_{u_i}, \pi_h\phi_p) \in V_h$, $i = 1, 2, 3$ *denote an interpolator defined in the above finite element space. Then, for the exact solution,* $\hat{u}$ *and the approximate (G2) solution,* $\hat{u}_h$, *the following error representation formula for the compressible Euler equations, (3.4), holds*

$$\begin{aligned} M(\hat{u}) - M(\hat{u}_h) = &\int_{I_n} \sum_{K \in \mathcal{T}_n} \left(-R(\hat{u}_h) + h.o.t., \hat{\phi} - \pi_h\hat{\phi} \right)_K \mathrm{d}t \\ &- \int_{I_n} \sum_{K \in \mathcal{T}_n} SD(\hat{u}_h; \pi_h\hat{\phi}) \mathrm{d}t, \end{aligned} \tag{5.5}$$

where $SD(\hat{u}_h; w) = (SD_\rho(\rho_h; w), SD_{u_i}(u_h; w), SD_p(p_h; w)), \forall w \in V_h, i = 1, 2, 3$.

Proof. Using (3.18) and (3.20), we get

$$\begin{aligned} M(\hat{u}) - M(\hat{u}_h) &= (e, \hat{\psi}) = (e_\rho, \psi_\rho) + (e_{u_i}, \psi_{u_i}) + (e_p, \psi_p) \\ &= (e_\rho, A^*_{\phi_\rho}(\hat{\phi})) + (e_{u_i}, A^*_{\phi_{u_i}}(\hat{\phi})) + (e_p, A^*_{\phi_p}(\hat{\phi})), \end{aligned} \tag{5.6}$$

which is nothing other than the relation (3.24) we got earlier in Theorem 3.5, and by repeating the same steps backwards (integration by parts) we get that

$$M(e) = M(\hat{u}) - M(\hat{u}_h) = (A_\rho(e_\rho), \phi_\rho) + (A_{u_i}(e_{u_i}), \phi_{u_i}) + (A_p(e_p), \phi_p). \tag{5.7}$$

Note that, $A = (A_\rho, A_{u_i}, A_p)$ is the corresponding primal operator to A^* from the above section 3.4. To prove the theorem we now use the relation between the residual and the error (3.19), and by subtracting (4.10) - (4.12) using $\pi_h\hat{\phi}$ as test function, we get an extra term from stabilization. The boundary terms from integration vanish because of the choice of boundary conditions in the primal and dual Euler equations. Splitting the space integration in, Ω, over all cells, K, in the triangulation, $\mathcal{T}_h$ we prove the theorem. □

Now we introduce the main theorem of the paper, an *a posteriori error estimate of the compressible Euler equation for the G2 method.*

THEOREM 5.2. (An a posteriori error estimate for G2). *Let* $\hat{u} = (\rho, u_i, p) \in L_2(I; H^1(\Omega) \times [H^1(\Omega)]^3 \times H^1(\Omega))$ *be the exact solution of (3.4);* $\hat{u}_h = (\rho_h, u_{h_i}, p_h)$ *defined in the G2 method (4.1) - (4.3), and* $D\hat{\phi}$ *denotes first order derivatives*

of $\hat{\phi}$. $\hat{\phi} = (\phi_\rho, \phi_{u_i}, \phi_p) \in \mathcal{C}(I; L_2(\Omega \times L_2(\Omega)^3 \times L_2(\Omega))$ *be the exact solution of the dual problem (3.20), with* $\dot{\hat{\phi}} \in L_2(I; L_2(\Omega \times L_2(\Omega)^3 \times L_2(\Omega))$, *and* $t_m \in I_n$ *be some point in the time interval* I_n, *then*

$$|M(\hat{u}) - M(\hat{u}_h)| \leq \sum_{n=1}^{N} k \max_{I_n} \left[C_n \sum_{K \in \mathcal{T}_n} h| \ D\hat{\phi}|_K \cdot |R(\hat{u}_h) + h.o.t.|_K + \sum_{K \in \mathcal{T}_n} |SD(\hat{u}_h; \pi_h \hat{\phi})| \right]_{t=t_m}, \tag{5.8}$$

where $SD(\hat{u}_h; w) = (SD_\rho(\rho_h; w), SD_{u_i}(u_h; w), SD_p(p_h; w)), \forall w \in V_h$ *is the stabilization term defined in the G2 method defined as above, (4.4 - 4.6).*

Here $\mathcal{C}(I; X)$ denotes the space of all continuous functions $v : I \to X$, with $sup_{t \in I} \|v(t)\|_X < \infty$, where X denotes a Banach space with norm $\|\cdot\|_X$.

Proof. We start from the result of Theorem 5.1.

Using a standard interpolation error estimate of the form $\|h_K^{-1}(\hat{\phi} - \pi\hat{\phi})\|_K \leq C_n \|D\phi\|_K$, we get:

$$\begin{aligned}
&|M(\hat{u}) - M(\hat{u}_h)|_{I_n} = \\
&\quad |\int_{I_n} \sum_{K \in \mathcal{T}_n} (-R(\hat{u}_h) + h.o.t., \hat{\phi} - \pi_h \hat{\phi}) \mathrm{d}t - \int_{I_n} \sum_{K \in \mathcal{T}_n} SD(\hat{u}_h; \pi_h \hat{\phi}) \mathrm{d}t| \\
&\quad \{\text{triangle inequality:}\} \\
&\quad \leq |\int_{I_n} \sum_{K \in \mathcal{T}_n} (-R(\hat{u}_h) + h.o.t., \hat{\phi} - \pi_h \hat{\phi}) \mathrm{d}t| + |\int_{I_n} \sum_{K \in \mathcal{T}_n} SD(\hat{u}_h; \hat{\phi}_h) \mathrm{d}t| \\
&\quad \leq \int_{I_n} \sum_{K \in \mathcal{T}_n} |(-R(\hat{u}_h) + h.o.t., \hat{\phi} - \pi_h \hat{\phi})| \mathrm{d}t + \int_{I_n} \sum_{K \in \mathcal{T}_n} |SD(\hat{u}_h; \pi_h \hat{\phi})| \mathrm{d}t \\
&\quad \{\text{CauchySchwartz inequality:}\} \\
&\quad \leq \int_{I_n} \sum_{K \in \mathcal{T}_n} \left(\int_K |R(\hat{u}_h) + h.o.t.|_K^2 \mathrm{d}x \right)_K^{1/2} \cdot \left(\int_K | \ \hat{\phi} - \pi_h \hat{\phi}|_K^2 \mathrm{d}x \right)_K^{1/2} \mathrm{d}t \\
&\quad + \int_{I_n} \sum_{K \in \mathcal{T}_n} |SD(\hat{u}_h; \pi_h \hat{\phi})| \mathrm{d}t \\
&\quad \leq \int_{I_n} \sum_{K \in \mathcal{T}_n} C_n h |R(\hat{u}_h) + h.o.t.|_K \cdot |D\hat{\phi}|_K + \int_{I_n} \sum_{K \in \mathcal{T}_n} |SD(\hat{u}_h; \pi_h \hat{\phi})| \mathrm{d}t
\end{aligned} \tag{5.9}$$

By numerical integration, the integral of any continuous function $f(t)$

can be estimated as $\int_{I_n} f(t)\mathrm{d}t \leq k \max_{I_n} f(t_m)$, where t_m is some point in the interval I_n. Using this property we have that

$$|M(\hat{u}) - M(\hat{u}_h)| \leq \underbrace{\sum_{n=1}^{N} k \max_{I_n} \left[C_n \sum_{K\in\mathcal{T}_n} h| \ D\hat{\phi}|_K \cdot |R(\hat{u}_h) + h.o.t.|_K + \sum_{K\in\mathcal{T}_n} |SD(\hat{u}_h; \pi_h\hat{\phi})| \right]_{t=t_m}}_{\sum_n \eta_n}. \tag{5.10}$$

□

REMARK 5.3. *In our numerical implementation we approximate $\hat{\phi}$ with a piecewise continuous function in space and time.*

The result of the theorem can be further estimated as:

$$|M(\hat{u}) - M(\hat{u}_h)| \leq S\epsilon, \tag{5.11}$$

where S is a *stability factor* that measures a certain norm of the computed approximation of the dual equation and ϵ measures the size of the residual, $R(\hat{u}_h)$, in a weak sense. The stability factor shows output sensitivity of the quantity of interest for the problem. Depending this sensitivity the stability factor has different values. To show the computability of the quantity of interest within given tolerance, TOL, one need to show that $S\epsilon \leq$ TOL. If the stability factor has a certain size, then one can make the G2 weak residual smaller in order to obtain bound within TOL. Therefore, if S is too large, it may not possible to make ϵ enough small.

In the numerical examples below, we study the stability factor for different time intervals.

5.1. An adaptive algorithm. In our simulation the mesh, $\mathcal{T}_h$, remains the same until the final time. Then, we solve the discretized dual equation with initial data at the final time and going backward. The mesh is adaptively refined in space according to the a posteriori error estimate (5.8). The timestep is calculated from the smallest cell diameter, with typically $\Delta t_n \sim \min_{E_j\in\mathcal{T}_h}(h/|u_n|)_{E_j}$, for all elements E_j in the mesh $\mathcal{T}_h$ and the velocity u_n.

Then, the adaptive algorithm is written as the following:

ALGORITHM 5.4. *Given some tolerance* TOL *and initial coarse mesh $\mathcal{T}_h^0$. Starting with time $k = 0$ do the following adaptive loop:*

1. Compute the primal solution of the compressible Euler equations, $\hat{u}_h \in V_h$*, on the current mesh* $\mathcal{T}_h^k$*;*

2. Compute the dual solution, $\hat{\phi}_h \in V_h$*, on the same mesh;*

3. Compute the error indicator defined as (5.8), if $|M(\hat{u}) - M(\hat{u}_h)| <$ TOL*, then* STOP*;*

4. Refine elements in $\mathcal{T}_h^k$ *with the largest error indicator and get mesh* $\mathcal{T}_h^{k+1}$*.*

5. Set $k = k + 1$ *and go to 1.*

As it is mentioned above, in this paper the adaptive algorithm based on the approximation of drag. In the computations below, the output of interest is the drag force. Below we describe how the normalized drag coefficient is computed.

5.2. Computation of drag. The drag coefficient is computed as follows:

$$C_{pd} = \frac{F_{pd}}{1/2\rho|u|^2 A}, \tag{5.12}$$

where F_{pd} is a drag force, ρ is the free stream density and u is the free stream velocity of the fluid, and A is the reference area. Usually, the reference area, A, is a projected area perpendicular to the direction of the fluid.

Since we are dealing with an inviscid compressible flow the drag force is computed only from the pressure, shear stress due to friction is not included. Therefore, the drag in this problem is *pressure drag* or *wave drag* and C_{pd} is the wave drag coefficient. Note, that if we normalize density ρ, with the gas constant γ, and velocity u, with speed of sound $c = 1$, then the drag coefficient, C_{pd} can be computed as follows

$$C_{pd} = \frac{F_{pd}}{1/2\gamma M_\infty^2 A}, \tag{5.13}$$

where M_∞ is the free-stream Mach number.

The drag force, F_{pd}, is computed from the surface integral of pressure and unit normal vector, n_i, directed toward the inside of the body:

$$F_{pd} = \int_{\Gamma_{body}} P n_i \mathrm{d}s. \tag{5.14}$$

6. Numerical Examples. Here we present numerical examples of compressible flow around an object in 2D and 3D with different Mach numbers. The computations are done according to above adaptive algorithm 5.4. First

we solve the primal equations, then with respect to a quantity of interest we solve the dual problem backward in time. Then we refine the mesh by the algorithm, and the simulations are repeated until the convergence. The quantity of interest in the below examples is a drag force acting on an object.

In the simulations we use dimensionless variables: Let L be a length, then the density ρ^*, sound speed c^* and the temperature T^* are normalized by their free stream values, $\rho^*_\infty, c^*_\infty, T^*_\infty$. The time t is normalized by L/c^*_∞, pressure by $p = \frac{p^*}{\rho^*_\infty c^2_\infty}$ and velocity by $u = \frac{u^*}{c_\infty}$.

We use the following boundary conditions for the primal equations: at the inlet all variables of the solution are given, at the outlet boundary conditions based on characteristics are used, and a slip boundary condition is applied in the rest of the computational domain. For the supersonic case where $M > 1$, the characteristics of the Euler equation goes outward at the outlet, a homogeneous Neumann or "do nothing" boundary condition is applied at the outlet. The homogeneous Dirichlet boundary condition is used for the dual equations for the density and pressure, and for the dual velocity we use $\phi_{u_i}(\cdot, x_i) = (-1, 0, 0) \quad \forall x_i \in \Gamma_{body}$ otherwise $\phi_{u_i}(\cdot, x_i) = 0$.

6.1. 2D flow around a cylinder. We consider a compressible supersonic flow with $M = 1.4$ around a cylinder, with diameter $d = 0.0254$. Here we compute the stability factor, $S = |D\hat{\phi}|$, for different end times, t, see Figure 6.1. We here would like to investigate the computational cost associated with the length of the interval over which we compute the mean drag. Indeed, Figure 6.1 shows that the larger space-time average are less computationally demanding than for shorter time averages.

Then we run the simulation until $t = 0.5$ to observe the behavior of the adaptive algorithm. Note, that this is not a stationary solution to the problem at this time, but the reference drag force which we compute with a very fine mesh, which is constructed by uniform refinements, behaves as constant from this time. We plot the solution which is obtained after eleven adaptive refinements. Figure 6.2 shows the solution of the dual problem at time $t = 0.35$ together with the sonic contour $M = 1$ at time $t = 0.5$. The shock waves develop along the sonic contour. One see that the dual solution indicates the upstream region, close to the supersonic region, and the downstream region where the wake will develop, to be important for the drag value. However, the residual of the primal equations is not large close to the inlet, it is usually large along the shock waves and wake. According the adaptive algorithm 5.4 the error indicator is computed in each cell K of the triangulation $\mathcal{T}_h$.

Figure 6.3 shows the drag coefficient C_{dp} from the last adaptive iteration together with a reference C^{ref}_{dp}, which is obtained from a fine mesh with

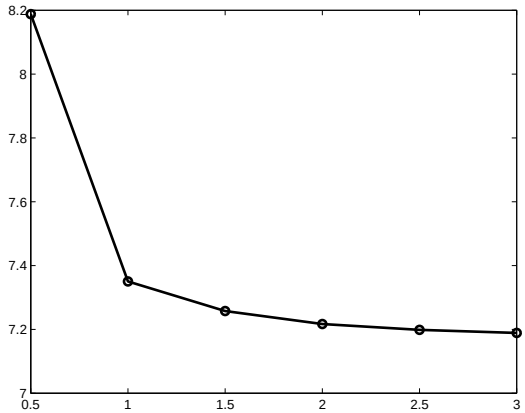

FIG. 6.1. *The stability factor, $S = |D\hat{\phi}|$ is plotted versus different final times from the flow around a cylinder in 2D.*

115718 vertices and 230852 cells. We can see that already after sevens adaptive iterations the drag oscillates about the reference drag. Figure 6.4 shows log10 of $\sum_n \eta_n$ versus log10 of number of cells.

We summarize the output from the adaptive algorithm in the Table 6.1. From the ratio of the reference and adaptive drag coefficients, $C_{dp}/\bar{C}_{dp}^{ref}$ one see the convergence already after eights adaptive iterations.

6.2. 2D flow around a wedge. In this example we simulate supersonic $M = 3$ flow around a wedge using the G2 method. For this, consider a 30^o angle wedge at zero angle of attack in $M = 3$ flow Figure 6.5. Similar to the above example the problem setup is according to the adaptive algorithm 5.4. One main feature of this simulation is that, the oblique shock in compressible flow is attached to the wedge, however for the cylinder there is a detached bow shock, see Figure 6.2. Also, in the downstream of the wedge, the shock-expansion waves develops by the sharp edges.

The boundary conditions are chosen the same as above mentioned. There is not done any treatment at the corners of the wedge, resulting a strong effect to the flow close to the body and downstream. This effect gets negligible when the mesh is enough refined. We can see that the adaptive algorithm focuses to this singularity points.

#iter	#vertices	#cells	S	$\bar{C}_{dp}$	$\bar{C}_{dp}^{ref}$	$\sum_n \eta_n$	$\bar{C}_{dp}/\bar{C}_{dp}^{ref}$
7	21716	42888	3.1018	1.3672	1.3970	0.5653	0.9786
8	24017	47482	3.5975	1.4238	-	0.3730	1.0191
9	26584	52600	4.2220	1.3640	-	0.2927	0.9764
10	29557	58530	4.7777	1.4453	-	0.1923	1.0345
11	32885	65154	5.4866	1.4090	-	0.1521	1.0086
12	36662	72676	6.0550	1.4438	-	0.1291	1.0335

TABLE 6.1

The convergence history of the drag coefficient. Here S is the stability factor, $\bar{C}_{dp}$ is a mean value of C_{dp} over the time interval $[0, t]$, C_{dp}^{ref} is a reference drag coefficient obtained from the fine mesh. $\sum_n \eta_n$ denotes a sum of error indicators.

We follow the same steps as example above, and solve the primal and dual problems. We compute a reference drag coefficient from two fine meshes: $C_{dp}^{ref_1}$ is obtained from the mesh with 91074 vertices and 181844 cells, and $C_{dp}^{ref_2}$ is obtained from the mesh with 144086 vertices and 287740 cells. However, the drag coefficients obtained from these two reference meshes are not close to each other. The reason can be the effect from a singularity point at the tip of the wedge, for the finest mesh this effect is smaller. Figure 6.7 shows the triangulation close the the tip. The left plot is the result of the adaptive algorithm after eight iterations with 41067 vertices and 81112 cells and the right plot is the triangulation for the finest reference mesh. We can see, that the adaptive algorithm focuses to resolve the singularity point, which makes the effect smaller. Figure 6.6 shows the drag coefficients from the adapted mesh together with the reference coefficients. We see that the drag for the finest reference mesh is closer to the results from the adapted mesh.

We summarized the computational outputs in the Table 6.2. The numbers in the table show that, the mean value of drag for the adapted meshes are quite close to each other. For the first reference mesh, the coefficient is higher, since the effect of the singularity point is large for the coarser mesh.

Figure 6.8 shows the dual and primal solutions at time $t = 0$ and $t = 0.5$ respectively. The upper plot shows the dual pressure in color, dual density contours and dual velocity arrows together with 10 contours of the Mach number. We can see that the oblique shock is attached to the wedge, it starts from the tip and develops downstream. From the other edges, the shock

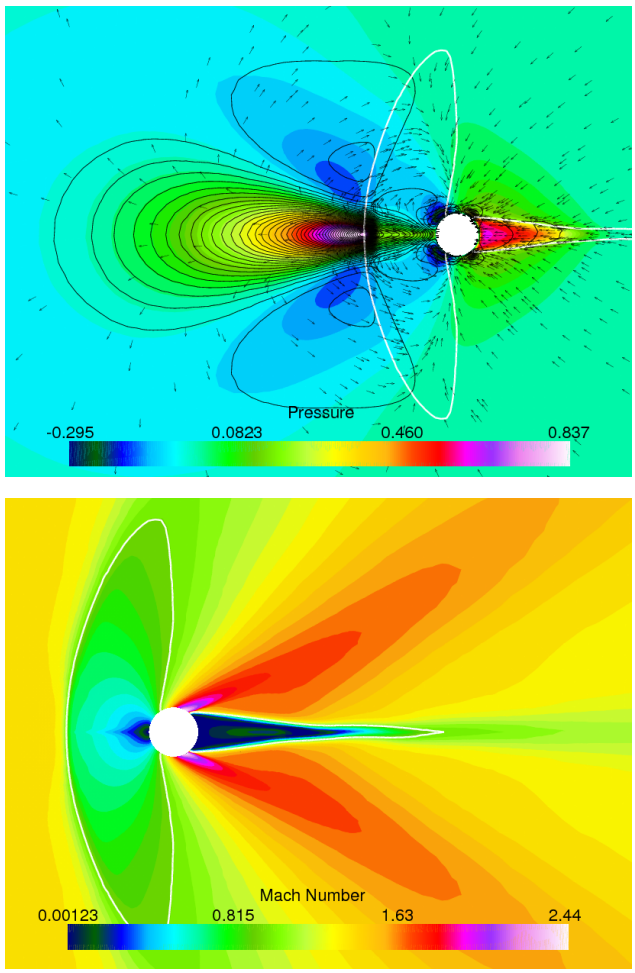

FIG. 6.2. *Supersonic flow around a 2D cylinder: the pressure in the colors, density contours and velocity arrow at time* $t = -0.15$ *together with the sonic contour* $M = 1$ *at time* $t = 0.5$*, (upper). The Mach number at time* $t = 0.5$*, (lower).*

expansion appears. The dual density follows the oblique shock for a while and advects upstream. All dual solutions are focusing on the area close to the tip, and the wake which develops after the wedge. The residual is usually high along the shock waves, however for the regions far from the wedge the dual

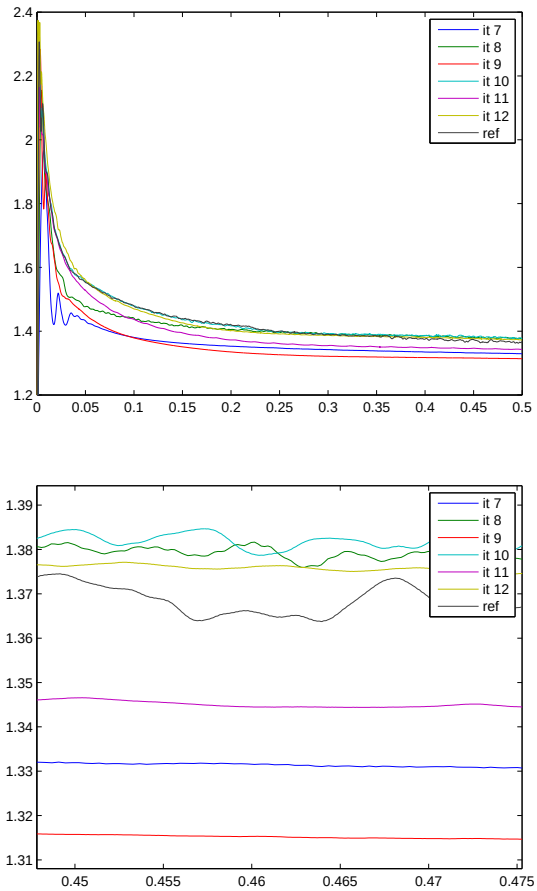

FIG. 6.3. *Supersonic flow around a 2D cylinder: the drag coefficient C_{dp} in different adaptive iterations and the reference C_{dp} which is obtained from the reference mesh.*

solution is almost zero. This indicates that for accurately approximating the drag force, it is not important to resolve the area with zero dual solution. This significantly minimizes the computational cost of the simulation. We plot the meshes from the adaptive algorithm in Figure 6.9 both for the 2D cylinder and wedge. In both examples, the algorithm refines only the part of the shock

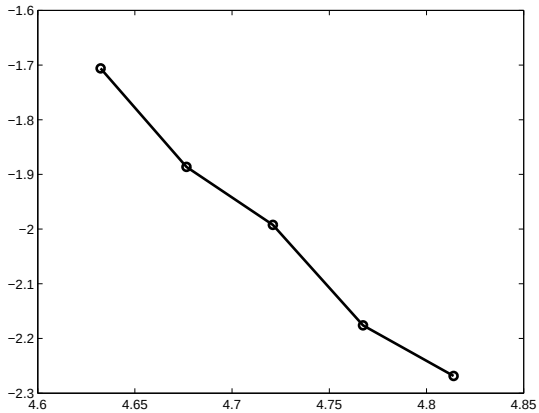

FIG. 6.4. *Supersonic flow around a 2D cylinder: Log10 of* $\sum_n \eta_n$ *versus log10 of number of cells.*

region which has significantly effect to the drag force.

#iter	#vertices	$\bar{C}_{dp}$	$\bar{C}_{dp}^{ref_1}$	$\bar{C}_{dp}^{ref_2}$	$\sum_n \eta_n$	$\bar{C}_{dp}/\bar{C}_{dp}^{ref_1}$	$\bar{C}_{dp}/\bar{C}_{dp}^{ref_2}$
5	25194	0.2489	0.2874	0.2616	0.4286	0.8663	0.9516
6	28292	0.2492	-	-	0.3429	0.8671	0.9526
7	31898	0.2497	-	-	0.2286	0.8690	0.9546
8	36140	0.2528	-	-	0.1810	0.8796	0.9663

TABLE 6.2

The convergence history of a drag coefficient for the Mach 3 flow around a wedge. Here $\bar{C}_{dp}$ *is a mean value of* C_{dp} *over the time interval* $[0,t]$*,* $C_{dp}^{ref_1}$ *is a reference drag coefficient from obtained from the fine mesh with 91074 vertices and 181844 cells,* $C_{dp}^{ref_2}$ *is a reference drag coefficient from the finest mesh with 144086 vertices and 287740 cells, and* $\sum_n \eta_n$ *denotes a sum of error indicators.*

6.3. 3D subsonic flow around the circular cylinder. We consider compressible supersonic flow, for $M = 0.8$ around a cylinder, with diameter $d = 0.0254$ and height $12d$ in three dimensions. This is a well-known

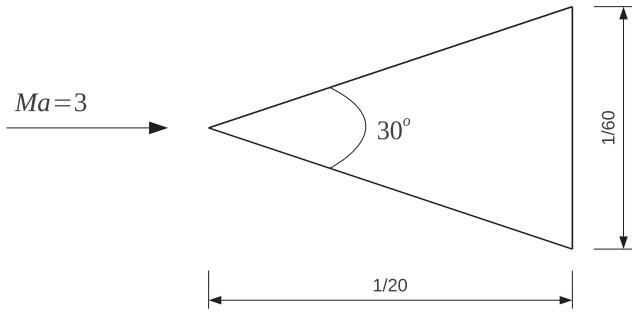

FIG. 6.5. *The geometry of the two dimensional wedge.*

benchmark problem with available experimental data by Murthy and Rose [27]. In the experiments they studied the transonic and supersonic flow with high Reynold's number and different Mach numbers. To make the discussion of the paper shorter, here we simulate the benchmark problem only for one Mach number. We leave the complete discussion together with different Mach numbers for the future research.

Again, the same steps of the adaptive algorithm is done in this case. The initial mesh with 17523 vertices and 89525 tetrahedrons adaptively refined 10 times up to 99387 vertices and 524141 tetrahedrons. For Mach 0.8 flow, there is transition to supersonic flow close to the cylinder, where the shock wave is attached. Due to this there occurs a large change in the pressure distribution which is a reason of a substantial increase in the total drag. This source of drag is usually called a wave drag. To accurately compute the wave drag, one need to capture the attached shock wave to the surface.

Table 6.3 shows the convergence history of the drag coefficient with the adaptive iterations which is approaching experimental data 1.5 by Murthy and Rose [27].

Here we present the solution from the dual problem from the finest mesh. Also, the adapted mesh is compared with the initial mesh, see Figure 6.10. and 6.11.

7. Conclusion. In this paper we presented an adaptive General Galerkin finite element method for the compressible Euler equations. A General Galerkin method is a stabilized cG(1)cG(1) finite element method with componentwise least square stabilization and residual based shock capturing term. We use the cG(1)cG(1) method for solving a dual problem for the compressible Euler equations.

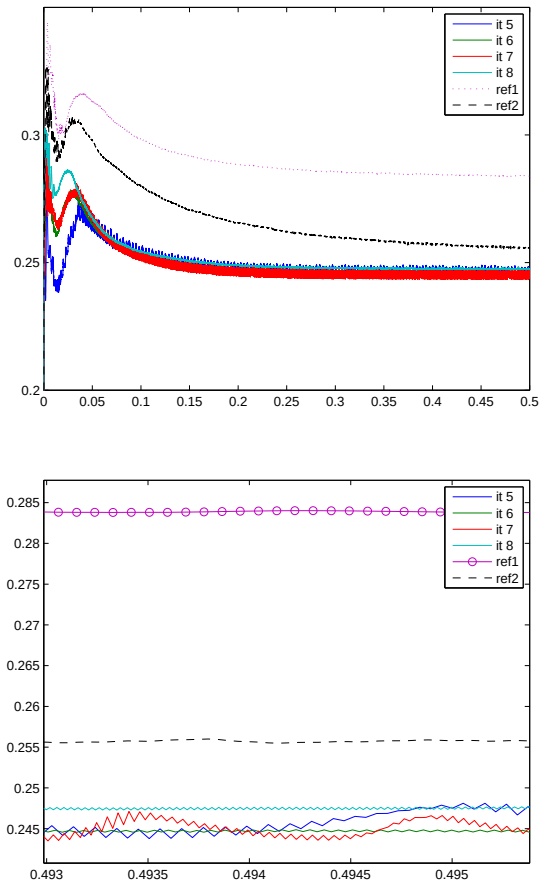

FIG. 6.6. *Supersonic flow around a 2D wedge: the drag coefficient C_{dp} in different adaptive iterations and two reference coefficients $C_{dp}^{ref_1}$ and $C_{dp}^{ref_2}$.*

The objective of the paper was to derive a duality based a posteriori error estimate which could be used for example to compute the drag of an object. This paper extends earlier work on Adaptive DNS/LES for incompressible flow, focussing on hig Reynolds numbers for different Mach numbers. The method is demonstrated for 2D and 3D numerical examples. The a posteriori

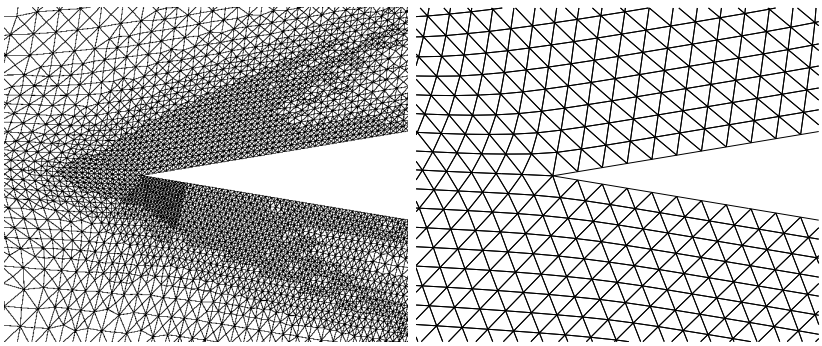

FIG. 6.7. *Supersonic flow around a 2D wedge: the mesh close to the tip of the wedge: after nine adapted iteration (left), and the reference mesh (right) of the two dimensional wedge.*

#iter	$\bar{C}_{dp}$
1	2.3933
2	2.2463
3	1.8993
4	1.6847
5	1.6924
6	1.6389
7	1.5681
8	1.5312

TABLE 6.3

The convergence history of a drag coefficient versus adaptive iterations for the transonic flow in 3D.

error estimation, including the dual problem, is based on expressing the Euler equations in the variables density, velocity and pressure, contrary to the more common choice of density, momentum and energy. Future work will focus on studying the method for turbulence benchmarks problems, and to extend the method to viscous compressible flow modeled by the Navier-Stokes equations.

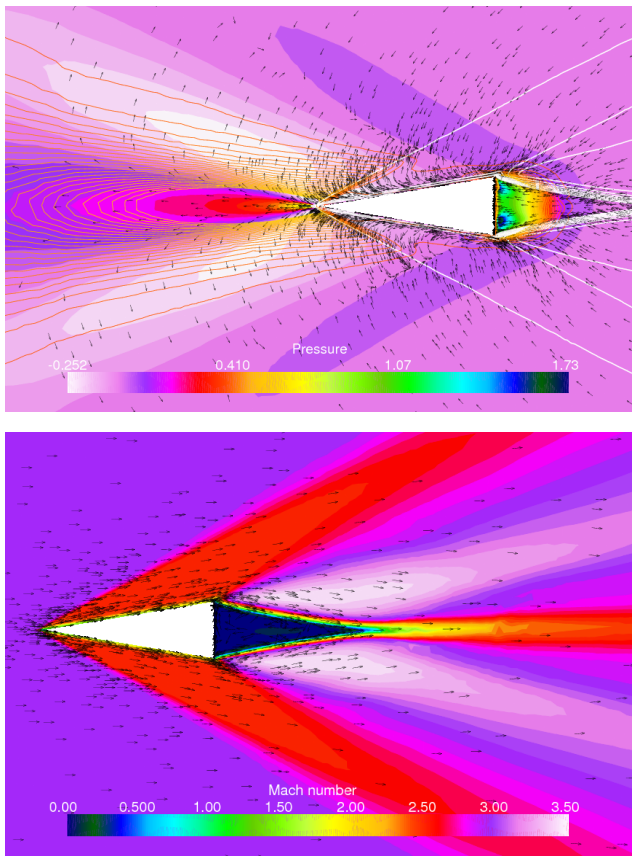

FIG. 6.8. *Supersonic flow around a 2D wedge: the dual pressure in the colors, dual density contours and velocity arrow at time* $t = -0.5$ *together with 10 contour of Mach numbers at time* $t = 0.5$*, (upper). The Mach number at time and primal velocity arrows* $t = 0.5$*, (lower).*

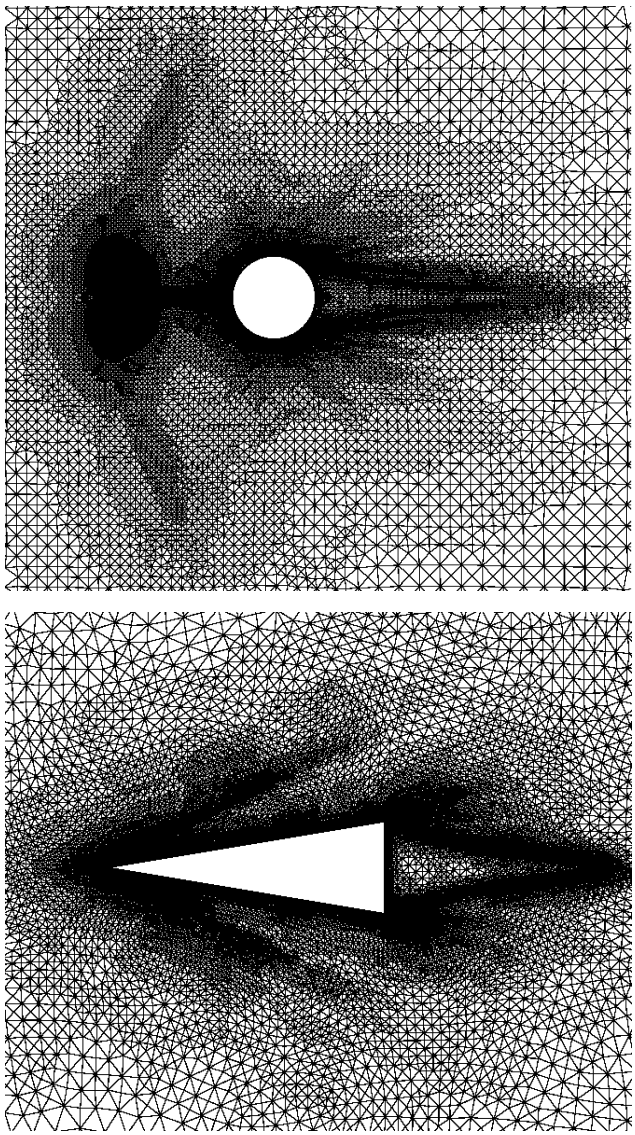

FIG. 6.9. *Adapted meshes for the 2D (above) cylinder after twelve, and wedge (below) after nine adaptive iterations.*

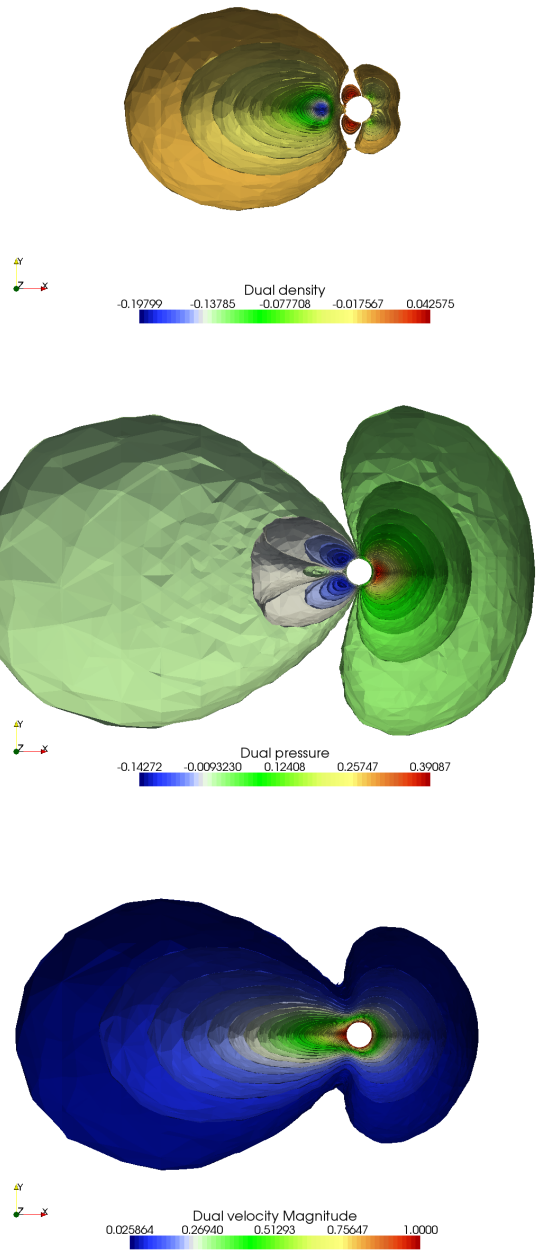

FIG. 6.10. *Transonic flow around a 3D cylinder: 30 contours of the dual density (above), pressure (middle) and velocity (below) from the finest mesh*

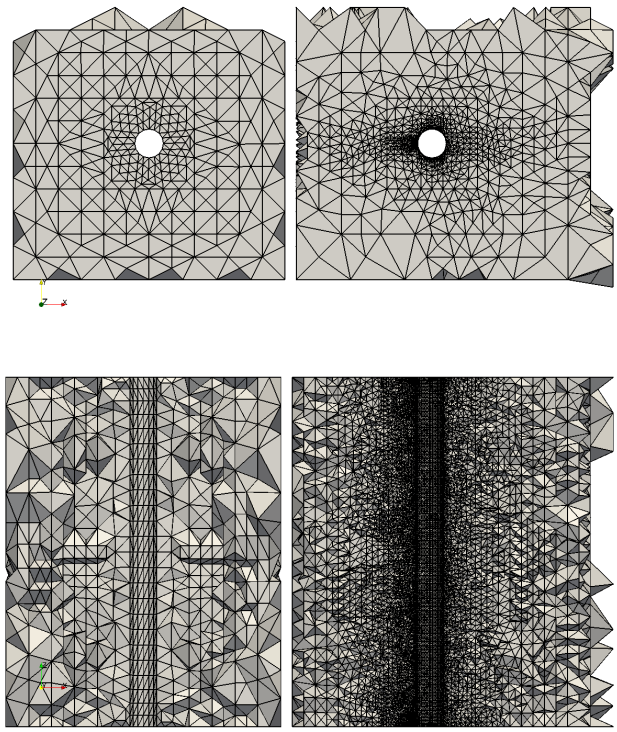

FIG. 6.11. *Transonic flow around a 3D cylinder: Initial mesh (left) and adapted mesh from the last iteration (right).*

REFERENCES

[1] I. Babuška, *Feedback, adaptivity and a posteriori estimates in finite elements: Aims, theory, and experience*, Accuracy, Estimates and Adaptive Refinements in Finite Element Computation (I. Babuška, O. C. Zienkiewicz, J. Gago, and E. R. de A. Oliveira eds.), Wiley, New York, 1986.

[2] I. Babuška and A. D. Miller, *The post-processing approach in the finite element method, i: Calculation of displacements, stresses and other higher derivatives of the dispacements.*, Int. J. Numer. Meth. Eng., 20 (1984), pp. 1085–1109.

[3] Roland Becker and Rolf Rannacher, *A feed-back approach to error control in adaptive finite element methods: Basic analysis and examples*, East-West J. Numer. Math., 4 (1996), pp. 237–264.

[4] ———, *A posteriori error estimation in finite element methods*, Acta Numer., 10 (2001), pp. 1–103.

[5] Erik Burman, *Adaptive finite element methods for compressible flow*, Computer methods in applied mechanics and engineering, 190 (2000), pp. 1137–1162.

[6] Kenneth Eriksson, Don Estep, Peter Hansbo, and Claes Johnson, *Introduction to adaptive methods for differential equations*, Acta Numer., 4 (1995), pp. 105–158.

[7] ———, *Computational Differential Equations*, Cambridge University Press New York, 1996.

[8] Kenneth Eriksson and Claes Johnson, *An adaptive finite element method for linear elliptic problems*, Math. Comp., 50 (1988), pp. 361–383.

[9] L. C. Evans, *Partial Differential Equations*, American Mathematical Society, 1998.

[10] Ralf Hartmann, *Adaptive FE Methods for Conservation Equations*, in Hyperbolic Problems: theory, numerics, applications: eighth international conference in Magdeburg, February, March 2000, Heinrich Freistühler and Gerald Warnecke, eds., vol. 141 of International series of numerical mathematics, Birkhäuser, Basel, 2001, pp. 495–503.

[11] Johan Hoffman, *On duality based a posteriori error estimation in various norms and linear functionals for les*, SIAM J. Sci. Comput., 26(1) (2004), pp. 178–195.

[12] ———, *Computation of mean drag for bluff body problems using adaptive dns/les*, SIAM J. Sci. Comput., 27(1) (2005), pp. 184–207.

[13] ———, *Adaptive simulation of the sub-critical flow past a sphere*, J. Fluid Mech., 568 (2006), pp. 77–88.

[14] ———, *Computation of turbulent flow past bluff bodies using adaptive general galerkin methods: drag crisis and turbulent euler solutions*, Comput. Mech., 38 (2006), pp. 390–402.

[15] ———, *Efficient computation of mean drag for the subcritical flow past a circular cylinder using general galerkin g2*, International Journal of Numerical Methods in Fluids, 59(11) (2009), pp. 1241–1258.

[16] Johan Hoffman, Johan Jansson, and Murtazo Nazarov, *A general galerkin finite element method for the compressible equations*, SIAM J. Scientific Computing, *submitted* (2008).

[17] Johan Hoffman and Claes Johnson, *A new approach to computational turbulence modeling*, Compit. Meth. Appl. Engrg, 195 (2006), pp. 2865–2880.

[18] J. Hoffman and C. Johnson, *A new approach to computational turbulence modeling*, Comput. Methods Appl. Mech. Engrg, 195 (2006), p. 319.

[19] Johan Hoffman and Claes Johnson, *Stability of the dual navier-stokes equations*

and efficient computation of mean output in turbulent flow using adaptive dns/les, Comput. Methods Appl. Mech. Engrg., 195 (2006), pp. 1709–1721.

[20] Paul Houston and Ralf Hartmann, *Goal–oriented a posteriori error estimation for compressible fluid flows*, in Numerical Mathematics and Advanced Applications, F. Brezzi, A. Buffa, S. Corsaro, and A. Murli, eds., Springer, 2003, pp. 775–784.

[21] Paul Houston, Ralf Hartmann, and Andre Süli, *Adaptive discontinuous Galerkin finite element methods for compressible fluid flows.*, in Numerical methods for Fluid Dynamics VII, ICFD, M. Baines, ed., 2001, pp. 347–353.

[22] C. Johnson, *Adaptive Finite Element Methods for Conservation Laws*, Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, Springer Lecture Notes in Mathematics, Springer Verlag, 1997.

[23] ———, *Adaptive Finite Element Methods for Conservation Laws*, Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, Springer Lecture Notes in Mathematics, Springer Verlag, 1998, pp. 269–323.

[24] C. Johnson and A. Szepessy, *Adaptive finite element methods for conservation laws based on a posteriori error estimates.*, vol. 48, Commun. Pure. Appl. Math.

[25] C. Johnson, A. Szepessy, and P. Hansbo, *On the convergence of shock-capturing streamline diffusion finite element methods for hyperbolic conservation laws*, Mathematics of Computation, 54(189) (1990), pp. 107–129.

[26] M.G. Larson and T.J. Barth, *A posteriori error estimation for discontinuous galerkin approximations of hyperbolic systems*, In Discontinuous Galerkin Methods, B. Cockburn, G. Karniadakis, and C.-W. Shu, editors, Lecture Notes in Comput. Sci. Engrg., 11, Springer, Berlin, (2000), pp. 363–368.

[27] V. S. Murthy and W. C. Rose, *Detailed measurements on a circular cylinder in cross flow*, AIAA Journal, 16 no. 6 (1978), pp. 549–550.

[28] P. Sagaut, *Large Eddy Simulation for Incompressible Flows*, Springer-Verlag, Berlin, Heidelberg, New York, 2001.

[29] T. E. Tezduyar and M. Senga, *Stabilization and shock-capturing parameters in supg formulation of compressible flows*, Computer Methods in Applied Mechanics and Engineering, 195 (2006), pp. 1621–1632.