Adaptive error control for steady state solutions of inviscid flow

Lars Ferm and Per Lötstedt*

February 4, 2000

Abstract

The steady state solution of the Euler equations of inviscid flow is computed by an adaptive method. The grid is structured and is refined and coarsened in predefined blocks. The equations are discretized by a finite volume method. Error equations, satisfied by the solution errors, are derived with the discretization error as the driving right hand side. An algorithm based on the error equations is developed for errors propagated along streamlines. Numerical examples from two-dimensional compressible and incompressible flow illustrate the method.

Keywords: finite volume method, discretization error, error control, error equation, Euler equations

AMS subject classification: 65N15, 65N50

1 Introduction

Inadequate grid resolution is the most important contributor to inaccuracy in computational fluid dynamics calculations (CFD) [28]. This is a serious obstacle in the industrial use of CFD methods. Grid refinement studies reveal the sensitivity of the solution to the grid spacing. Such studies are expensive and time-consuming to make for an applications engineer and usually only one grid is generated based on his or her experience. There is a need for solution adaptive methods.

The errors in the numerical solution of the partial differential equations (PDEs) governing inviscid flow can be controlled by an adaptive method. By changing the grid spacing so that the solution errors fulfill given tolerances, the quality of the solution is guaranteed. Furthermore, computational work and memory is saved since the grid is not made unnecessarily fine. Two important ingredients in an

*Dept. of Information Technology, Scientific Computing, Uppsala University, SE-75104 Uppsala, Sweden

adaptive algorithm are a data structure for the grid that allows for modifications of the original grid and a way of estimating the solution error.

In this paper the computational domain is partitioned into a number of blocks. The grid in each block is structured and all cells in a block are refined or coarsened following the error estimate. The data structure is simple since the neighbors of a block are the same in the original grid and all the adapted grids. Inside a block the organization of the data is also simple since the size of all cells in a block are changed when the grid there is refined or coarsened as in [18] and [26]. There will be jumps in the grid size only at the block boundaries and interpolation is needed for the cell variables at the boundary. Although we use structured grids the principles are applicable also to unstructured grids and hybrid grids with a mixture of blocks with structured and unstructured grids. Block partitioning is also suitable for parallelization of a code. Each processor solves the equations in a subset of the blocks with communication limited to the boundaries of blocks with a neighbor in another subset.

The equations are discretized by finite volume methods. They are of second order on Cartesian grids. The interpolation at block boundaries is such that second order accuracy is preserved in the cells adjacent to the boundary.

The flow equations are solved on structured grids with patches of refinement in [3],[4],[5],[12],[30],[36]. The refined parts follow the grid lines but can be positioned arbitrarily in the domain. The data structure is more complicated than ours but has greater flexibility since the patches can surround the points where refinement is needed with less extra cells. Other equations related to the flow equations are solved with similar techniques and adaptivity in [10],[22],[32]. Individual cells are split for refinement in [13],[31]. The parental and adjacent cells are kept track of by a tree data structure. The idea is here the same for both structured and unstructured grids (cf. [9] and [31]). The price to be paid for the possibility to introduce cells precisely where they are needed is the administration of the advanced data structure and the indirect addressing of the data. For structured grids it is also complicated to obtain second order accuracy in the discretization. Instead of introducing more cells, the grid points in 1D are moved in [6] and [24] and the grid lines are moved in 2D in [25] to adapt the original grid. The advantage is that the available cells are clustered in areas where high resolution is required and fewer cells are used in smooth areas. A disadvantage is the distortion of the grid which may be severe in three dimensions around complicated geometries. Moreover, instabilities with moving grid algorithms are discussed in [24] and there is no guarantee that spurious solutions do not appear [6].

The refinement and the coarsening of the grid is determined by estimates of the solution error. Grid adaptation is necessary not only at discontinuities in the solution but also in the smooth parts. Differential equations are derived which are satisfied approximately by the errors. The right hand side of the error equations is the discretization error in the finite volume method. This error is

estimated by comparing the space discretization on two different grids. Then the error equations are solved numerically for the error in the solution. For certain variables, these equations are particularly simple and the error is propagated along the streamlines of the flow. An algorithm is devised for such errors. It takes the sign of the error into account when the grid is adjusted so that an error tolerance is fulfilled. The truncation error generated at a point may propagate and influence the solution error far downstream. The large error in the solution in a cell is not necessarily caused by too coarse a grid there.

In [3],[4],[5],[27],[30],[36] the discretization or truncation error is estimated by comparing the discretization on different grids as in Richardson extrapolation. There is no control of the error in the solution. The estimation of a local discretization error is the standard procedure in the time step regulation of the numerical solution of ordinary differential equations [21],[35]. Another possibility is to base the refinement on sensors such as pressure gradients or vorticity, see e.g. [9],[12],[13],[31]. This approach yields good results in practice but it is difficult to relate the sensor values to quantitative information on the solution errors. Wavelets are used in [19] to detect sharp features in the solution. Bounds on the errors in finite element methods are computed a posteriori using the solution to the adjoint problem in [17]. Then a another linear system of differential equations has to be solved for the adjoint solution. A general discussion of numerical errors and their estimation in CFD is found in [34].

In the next section, the errors in finite volume methods are investigated and the estimate of the discretization error is justified. The error equations for compressible and incompressible inviscid flow problems are derived in Sect. 3 following the results in Sect. 2. The equations are particularly simple for the total pressure in incompressible flow and the entropy and the enthalpy in compressible flow. The adaptation algorithm is found in Sect. 4. Finally, numerical solutions of the incompressible flow around a half-cylinder and compressible flow around an airfoil at transonic and subsonic speeds are computed with the algorithm.

2 Numerical errors in finite volume methods

The partial differential equation

$$\frac{\partial u}{\partial t} + \nabla \cdot \mathcal{F}(u) = 0,\ L(u) = 0 \text{ on } \partial\Omega,\ u = u_0 \text{ at } t = 0, \tag{1}$$

with the flux vector $\mathcal{F}$ is discretized on $\Omega \times [0, t_{end}]$ by the finite volume method (FVM) in space and a linear multistep method (LMM) in time. The domain Ω with boundary $\partial\Omega$ is covered by m cells ω_j with a normal $\hat{n}$, boundary $\partial\omega$ and volume or area A_j. This section is devoted to the analysis of the solution errors in the discrete approximation of (1). The adaptive algorithm is developed only for steady state problems in this paper, but we include a discussion of temporal errors in this and the next section for completeness.

2.1 Space discretization

Integrate (1) over ω_j and use Gauss' theorem to obtain

$$\int_{\omega_j} u_t d\omega + \int_{\partial\omega_j} \mathcal{F}(u) \cdot \hat{n} ds = 0.$$

In the sequel, a coordinate as a subscript of a variable denotes differentiation with respect to that coordinate. Let $\bar{u}_j$ be the average of u in cell j and let G be defined by

$$G_j(u) = \bar{u}_{jt} + A_j^{-1} \int_{\partial\omega_j} \mathcal{F}(u) \cdot \hat{n} ds = \bar{u}_{jt} + F_j(u). \tag{2}$$

The solution u of (1) satisfies

$$G(u) = 0. \tag{3}$$

At time t_n, $\bar{u}_j$ is approximated by the numerical solution $\bar{v}_j^n$. By reconstruction from $\bar{v}_j^n$ (see e.g. [1], [7]), v_j^n is created in Ω such that $v_j^n \in C^q(\Omega),\ q \geq 0$, and

$$\bar{v}_j^n = A_j^{-1} \int_{\omega_j} v^n d\omega.$$

Let $\Psi_j^i(v)$ be the discrete flux in cell j on edge i, l be the number of edges and $\hat{n}_i$ the normal on edge i. Then the discretization of the integral in (2) is

$$\Phi_j(\bar{v}) = A_j^{-1} \sum_{i=1}^{l} \Psi_j^i(v) \cdot \hat{n}_i. \tag{4}$$

The discretization of the boundary condition on $\partial\Omega$ in (1) is

$$\Lambda_i(\bar{v}) = 0,\ i = 1 : m_b, \tag{5}$$

at m_b boundary cells. Since the numerical scheme may need more boundary conditions than the analytical solution, $\Lambda(u)$ is usually not a direct approximation of $L(u)$.

2.2 Time discretization

A linear multistep method, defined by its coefficients $\alpha_i, \beta_i, i = 0 : k$ (see [21]), is chosen for the approximation of (2) in time. Let Δt be the time step from t_{n-1} to t_n and introduce

$$, _j(\bar{v}^n) = \Delta t^{-1} \sum_{i=0}^{k} \alpha_i \bar{v}_j^{n-i} + \sum_{i=0}^{k} \beta_i \Phi_j(\bar{v}^{n-i}). \tag{6}$$

Then $\bar{v}_j^n$ satisfies

$$, (\bar{v}^n) = 0. \tag{7}$$

There are advantages of LMMs compared to Runge-Kutta methods. Fewer evaluations of the space discretization Φ are usually needed and less data are communicated in each time step in a parallel implementation based on domain decomposition of Ω. Two families of these methods are of particular interest here: the Adams methods and the backward differentiation methods (BDF).

In an Adams method, the equation after finite volume discretization

$$\bar{v}_{jt} + \Phi_j(\bar{v}) = 0,$$

is integrated over an interval $[t_{n-1}, t_n]$

$$\begin{aligned}&\Delta t^{-1} \textstyle\int_{t_{n-1}}^{t_n} (\bar{v}_{jt} + \Phi_j(\bar{v}))dt = \\ &\Delta t^{-1}(\bar{v}_j(t_n) - \bar{v}_j(t_{n-1})) + \Delta t^{-1} \textstyle\int_{t_{n-1}}^{t_n} \Phi_j(\bar{v})dt = 0.\end{aligned}$$

Thus, $\alpha_0 = 1$ and $\alpha_1 = -1$ in (6). The integral of Φ_j is then approximated by choosing suitable β_i parameters. The derivative $\bar{v}_{jt}$ at t_n is discretized in a BDF-method so that

$$\bar{v}_{jt} \approx \Delta t^{-1} \sum_{i=0}^{k} \alpha_i \bar{v}_j^{n-i},\ \beta_0 = 1,\ \beta_i = 0, i = 1:k.$$

2.3 The error equation

As a measure of how well $, _j$ approximates G_j in Ω take

$$\tau_j(w) = G_j(w) - , _j(\bar{w}) \tag{8}$$

for $w \in C^q$. If u is the solution of (3), then the truncation error of $, _j$ is

$$, _j(\bar{u}) = -\tau_j(u). \tag{9}$$

In finite element analysis (see [17]), the FEM solution v is inserted into the differential equation G to compute the *residual* $r(v) = G(v)$ used in the error control. Here, we have the reconstruction v from the average solution satisfying $\tau(v) = G(v)$ since $, (\bar{v}) = 0$ in (8). The numerical error δv^n in v^n satisfying (7) is

$$\delta v^n = v^n - u(t_n),$$

where $u(t_n)$ solves (3) at t_n. We have from (8) and (3)

$$G_j(u(t_n) + \delta v^n) - G_j(u(t_n)) = \tau_j(v^n),\ j = 1:m. \tag{10}$$

The coefficients of the linearized error equation and the right hand side are evaluated at the numerical solution.

Since the numerical solution often needs more boundary conditions than the analytical solution, L is extended to $\tilde{L}$ on $\partial\Omega$ so that

$$\chi_i(w) = \tilde{L}_i(w) - \Lambda_i(\bar{w}). \tag{11}$$

Here, χ measures the discretization error in the boundary conditions. Consequently,

$$\tilde{L}_i(u(t_n) + \delta v^n) = \chi_i(v^n), \tag{12}$$

which is the boundary condition of δv^n on $\partial\Omega$. The inital condition is assumed to be exact.

We have derived an error equation for δv^n with accompanying boundary condition. The average error in a cell is

$$\delta\bar{v}_j^n = A_j^{-1} \int_{\omega_j} \delta v^n d\omega = \bar{v}_j^n - \bar{u}_j(t_n).$$

The numerical approximation fulfills a modification of the original equation. If G is linear, then from (10)

$$G_j \delta v^n = \tau_j(v^n). \tag{13}$$

There are two components of τ_j in (8). Subtract $,_j$ in (6) from G_j in (2) and split the difference into two parts: one due to the temporal discretization (τ_j^t) and one due to the spatial discretization (τ_j^s). The result is

$$G_j(w) - ,_j(\bar{w}) = \tau_j^t(w) + \tau_j^s(w),$$

where for the BDF schemes

$$\begin{array}{ll} \tau_j^t & = \bar{w}_{jt} - \Delta t^{-1} \sum_{i=0}^k \alpha_i \bar{w}_j^{n-i}, \\ \tau_j^s(w) & = F_j(w) - \Phi_j(\bar{w}), \end{array} \tag{14}$$

and for the Adams schemes

$$\begin{array}{ll} \tau_j^t & = \Delta t^{-1} \int_{t_{n-1}}^{t_n} \Phi_j(\bar{w}) dt - \sum_{i=0}^k \beta_i \Phi_j(\bar{w}^{n-i}), \\ \tau_j^s(w) & = \Delta t^{-1} \int_{t_{n-1}}^{t_n} (F_j(w) - \Phi_j(\bar{w})) dt, \end{array} \tag{15}$$

with F_j and Φ_j from (2) and (4). The space error is an average over $[t_{n-1}, t_n]$ for the Adams methods and is evaluated pointwise at t_n for the BDF family.

2.4 FVM error estimate

Assume that Ω is covered by triangles or quadrangles in 2D and by tetrahedrons or hexahedrons in 3D. Then the discretization error in space can be estimated asymptotically as follows by comparing the residual Φ_j on different grids.

Proposition 2.1 *Let cell ω_0 consist of the cells $\omega_j,\ j = 1:p$, and let h be a measure of the cell size so that size(ω_0)=h and size(ω_j)=$h/r,\ j = 1:p$. Assume that $u \in C^1(\Omega)$ and that the space discretization in cell j satisfies*

$$\Phi_j(\bar{u}) = F_j(u) - \tau(u_{\omega_j}), \tag{16}$$

where u_{ω_j} is determined at a point in ω_j. Furthermore, assume that the discretization error satifies

$$\tau(u) = h^\nu c(u) + h^{\nu+1} d(u), \tag{17}$$

where c is a linear operator

$$c(a_1 u_1 + a_2 u_2) = a_1 c(u_1) + a_2 c(u_2), \tag{18}$$

and that $\gamma_j = A_j/A_0,\ j = 1:p$. Then in $\omega_j,\ j = 0:p$,

$$\tau(u_{\omega_j}) = \tfrac{b_j}{r^\nu - 1}\left(\textstyle\sum_{j=1}^p \gamma_j \Phi_j(\bar{u}_j) - \Phi_0(\sum_{j=1}^p \gamma_j \bar{u}_j)\right) + O(h^{\nu+1}),$$
$$b_0 = r^\nu,\ b_j = 1,\ j > 1.$$

Proof For u in ω_0 we have

$$\Phi_0(\bar{u}_0) = \Phi_0\Big(\sum_{j=1}^p \gamma_j \bar{u}_j\Big) = F_0(u) - \tau(u_{\omega_0}). \tag{19}$$

Since $u \in C^1(\Omega)$ and by (17) and (18)

$$\tau(u_{\omega_0}) = h^\nu c(u_{\omega_0}) + h^{\nu+1} d(u_{\omega_0}) = h^\nu \sum_{j=1}^p \gamma_j c(u_{\omega_j}) + O(h^{\nu+1}). \tag{20}$$

A summation over ω_0 gives

$$\begin{array}{ll} \sum_{j=1}^p \gamma_j \Phi_j(\bar{u}) & = \sum_{j=1}^p \gamma_j F_j(u) - \sum_{j=1}^p \gamma_j \tau(u_{\omega_j}) \\ & = F_0(u) - (h/r)^\nu \sum_{j=1}^p \gamma_j c(u_{\omega_j}) + O(h^{\nu+1}). \end{array} \tag{21}$$

We infer from (19), (20), and (21) that

$$h^\nu \sum_{j=1}^p \gamma_j c(u_{\omega_j}) = \frac{r^\nu}{r^\nu - 1}\Big(\sum_{j=1}^p \gamma_j \Phi_j(\bar{u}_j) - \Phi_0\Big(\sum_{j=1}^p \gamma_j \bar{u}_j\Big)\Big)$$

By this expression the proposition follows.

The discretization error in space can be estimated by comparing the expression for the space derivatives on a fine and a coarse grid using the fine grid solution as in proposition 2.1. The result is similar to what is usually referred to as a Richardson estimate [4],[30],[34]. By comparing the solutions using two different time steps the discretization error in time can be approximated. Another possibility is to use two different time stepping methods and compare their solutions. This approach is developed in a separate paper [26].

3 Errors in the equations of inviscid flow

The equations satisfied by the reconstruction of the numerical solution in (10) are derived for compressible and incompressible flow. The error in the solution can then be estimated by numerical solution of these equations. An estimate of the effect of the discretization error, which can be controlled by changing the space discretization and the timestep, on the solution error, which is desirable to keep within given bounds, is obtained. We write the equations in the form (1) instead of the integrated form (2), since if the solution is smooth and satisfies (2), then it also satisfies (1). If the analytical and the numerical solutions are smooth, then the solution error satisfies PDEs similar to linearized versions of the original PDE.

3.1 Error equations

The analytical solution of a variable u is denoted by $\hat{u}$ and the numerical error by δu. The reconstructed numerical solution is $u = \hat{u} + \delta u$. Let ρ be the density, $\mathbf{U} = (u, v, w)^T$ the velocity vector, p the pressure, E the total energy and H the total enthalpy. Then the error equations for compressible flow on conservation form are [2]

$$\rho_t + (\rho u)_x + (\rho v)_y + (\rho w)_z = \tau_1, \tag{22a}$$
$$(\rho u)_t + (\rho u^2)_x + (\rho uv)_y + (\rho uw)_z + p_x = \tau_2, \tag{22b}$$
$$(\rho v)_t + (\rho uv)_x + (\rho v^2)_y + (\rho vw)_z + p_y = \tau_3, \tag{22c}$$
$$(\rho w)_t + (\rho uw)_x + (\rho vw)_y + (\rho w^2)_z + p_z = \tau_4, \tag{22d}$$
$$(\rho E)_t + (\rho uH)_x + (\rho vH)_y + (\rho wH)_z = \tau_5. \tag{22e}$$

These equations conserve mass, momentum, and energy. The system is closed by the internal energy e, the gas constant γ, and the relations

$$e = \frac{p}{(\gamma - 1)\rho},\ q^2 = u^2 + v^2 + w^2,\ E = e + \frac{1}{2}q^2,\ H = E + \frac{p}{\rho}.$$

The substantial derivative is defined by

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + u\frac{\partial}{\partial x} + v\frac{\partial}{\partial y} + w\frac{\partial}{\partial z} = \frac{\partial}{\partial t} + \mathbf{U}\cdot\nabla. \tag{23}$$

The equations in (22) can be written in nonconservation form using D/Dt

$$\frac{D\rho}{Dt} + \rho\nabla\cdot\mathbf{U} = \tilde{\tau}_1, \tag{24a}$$

$$\rho\frac{Du}{Dt} + p_x = \tilde{\tau}_2, \tag{24b}$$

$$\rho\frac{Dv}{Dt} + p_y = \tilde{\tau}_3, \tag{24c}$$

$$\rho\frac{Dw}{Dt} + p_z = \tilde{\tau}_4, \tag{24d}$$

$$\rho\frac{De}{Dt} + p\nabla\cdot\mathbf{U} = \tilde{\tau}_5. \tag{24e}$$

The relations between τ in (22) and $\tilde{\tau}$ in (24) are

$$\tilde{\tau}_1 = \tau_1, \tag{25a}$$

$$\tilde{\tau}_2 = \tau_2 - u\tau_1, \tag{25b}$$

$$\tilde{\tau}_3 = \tau_3 - v\tau_1, \tag{25c}$$

$$\tilde{\tau}_4 = \tau_4 - w\tau_1, \tag{25d}$$

$$\tilde{\tau}_5 = \tau_1(q^2 - E) - (u\tau_2 + v\tau_3 + w\tau_4) + \tau_5. \tag{25e}$$

The first relation in (25) is immediate. The next three equalities follow from (24a) and the definitions of τ and $\tilde{\tau}$ in (22) and (24). The last equation is derived by combining (22e) with the equations in (24) and (25).

The entropy S of the flow is defined by

$$S = c_v \log(p/\rho^\gamma). \tag{26}$$

Here c_v is the specific heat at constant volume. With the temperature T we have the equation for the entropy [2]

$$T\frac{DS}{Dt} = \frac{De}{Dt} + p\frac{D(1/\rho)}{Dt} = 0. \tag{27}$$

By (27), (24), and the equation of state $p = c_v(\gamma-1)\rho T$ the differential equation for the computed entropy S is

$$\frac{DS}{Dt} = c_v(\gamma-1)(\frac{\tilde{\tau}_5}{p} - \frac{\tau_1}{\rho}) = \tau_S. \tag{28}$$

The equation for the numerical error δS is particularly simple if the analytical solution $\hat{S}$ is constant. Then

$$\frac{D(\hat{S}+\delta S)}{Dt} = \frac{D\delta S}{Dt} = \tau_S. \tag{29}$$

In the numerical experiments in Sect. 5, $c_v = 1$ and $\gamma = 1.4$.

Another simple equation is fulfilled by the enthalpy H. From the last equation in (22) and the definition of H we arrive at (see also [2])

$$\rho\frac{DH}{Dt} - \frac{\partial p}{\partial t} = \tau_5 - H\tau_1. \tag{30}$$

In incompressible flow, we let $\rho = 1$ and ignore the last equation in (22) and (24) to obtain the conservation form

$$\nabla \cdot \mathbf{U} = \tau_1, \tag{31a}$$
$$u_t + (u^2)_x + (uv)_y + (uw)_z + p_x = \tau_2, \tag{31b}$$
$$v_t + (uv)_x + (v^2)_y + (vw)_z + p_y = \tau_3, \tag{31c}$$
$$w_t + (uw)_x + (vw)_y + (w^2)_z + p_z = \tau_4, \tag{31d}$$

and the nonconservation form

$$\nabla \cdot \mathbf{U} = \tilde{\tau}_1, \tag{32a}$$
$$\frac{Du}{Dt} + p_x = \tilde{\tau}_2, \tag{32b}$$
$$\frac{Dv}{Dt} + p_y = \tilde{\tau}_3, \tag{32c}$$
$$\frac{Dw}{Dt} + p_z = \tilde{\tau}_4. \tag{32d}$$

The relation between τ in (31) and $\tilde{\tau}$ in (32) is given by (25).

The total pressure P for an incompressible fluid is defined by

$$P = p + 0.5q^2. \tag{33}$$

Multiply the momentum equations in (32) by the velocity vector and use the definition of the substantial derivative (23) to obtain

$$\frac{DP}{Dt} - \frac{\partial p}{\partial t} = u\tilde{\tau}_2 + v\tilde{\tau}_3 + w\tilde{\tau}_4. \tag{34}$$

Hence, if $\hat{P}$ is constant in Ω in a steady state solution, the error in P fulfills the linear equation

$$\mathbf{U} \cdot \nabla \delta P = u\tilde{\tau}_2 + v\tilde{\tau}_3 + w\tilde{\tau}_4, \tag{35}$$

where $\mathbf{U}$ is the computed solution.

In 2D incompressible flow, another linear equation is satisfied by the vorticity ζ

$$\zeta = u_y - v_x.$$

Differentiate (32b) and (32c) with respect to y and x and subtract ζ times the first equation to obtain

$$\frac{D\zeta}{Dt} = \frac{D(\hat{\zeta} + \delta\zeta)}{Dt} = \tilde{\tau}_{2y} - \tilde{\tau}_{3x} - \zeta\tilde{\tau}_1. \tag{36}$$

3.2 Linearized equations

It is possible to simplify (24) if we make a few additional assumptions. For steady, compressible flow assume that

1. All derivatives of the analytical solution are small,
2. u, ρ, and p are of $O(1)$ but v and w are small,
3. The errors in the solution are small.

The first two assumptions are common in linearized flow analysis [2].

By (22a) and the assumptions after ignoring small terms we have

$$\begin{aligned} &\nabla \cdot ((\hat{\rho} + \delta\rho)(\hat{\mathbf{U}} + \delta\mathbf{U})) = \\ &\hat{\rho}\nabla \cdot \delta\mathbf{U} + \delta\rho\nabla \cdot \hat{\mathbf{U}} + \hat{\mathbf{U}} \cdot \nabla\delta\rho + \delta\mathbf{U} \cdot \nabla\hat{\rho} = \hat{\rho}\nabla \cdot \delta\mathbf{U} + \hat{u}\delta\rho_x = \tau_1. \end{aligned} \tag{37}$$

In the same manner, for the first momentum equation (22b) we find

$$\begin{aligned} &\nabla \cdot ((\hat{\rho} + \delta\rho)(\hat{\mathbf{U}} + \delta\mathbf{U})(\hat{u} + \delta u)) + (\hat{p} + \delta p)_x = \\ &(\hat{u} + \delta u)\nabla \cdot ((\hat{\rho} + \delta\rho)(\hat{\mathbf{U}} + \delta\mathbf{U})) + \hat{\rho}\hat{\mathbf{U}} \cdot \nabla\delta u + \\ &\rho\delta\mathbf{U} \cdot \nabla\hat{u} + \delta\rho\hat{\mathbf{U}} \cdot \nabla\hat{u} + \delta p_x = \hat{u}\tau_1 + \hat{\rho}\hat{u}\delta u_x + \delta p_x = \tau_2. \end{aligned}$$

Thus,

$$\hat{\rho}\hat{u}\delta u_x + \delta p_x = \tau_2 - \hat{u}\tau_1 = \tau_2'. \tag{38}$$

The other two momentum equations are simplified to

$$\begin{aligned} &\hat{\rho}\hat{u}\delta v_x + \delta p_y = \tau_3, \\ &\hat{\rho}\hat{u}\delta w_x + \delta p_z = \tau_4. \end{aligned} \tag{39}$$

The linearized energy equation follows from (22e) and (37)

$$\begin{aligned} &\nabla \cdot ((\hat{\rho} + \delta\rho)(\hat{H} + \delta H)(\hat{\mathbf{U}} + \delta\mathbf{U})) &&= \\ &\hat{H}\nabla \cdot ((\hat{\rho} + \delta\rho)(\hat{\mathbf{U}} + \delta\mathbf{U})) + \hat{\rho}\hat{u}\delta H_x &&= \\ &\hat{\rho}\hat{H}\nabla \cdot \delta\mathbf{U} + \hat{H}\hat{u}\delta\rho_x + \hat{\rho}\hat{u}\delta H_x &&= \tau_5. \end{aligned} \tag{40}$$

The error components $\delta\rho, \delta\mathbf{U}$, and δH satisfy (37), (38), (39), and (40).

One error equation for $\delta\mathbf{U}$ and one for δp will now be derived. From the definition of the enthalpy and elimination of δp via (38) and (37) we have

$$\begin{aligned}
&\hat{H}\hat{u}\delta\rho_x + \hat{\rho}\hat{u}\delta H_x = (\gamma/(\gamma-1))\hat{u}\delta p_x + 0.5\hat{u}^3\delta\rho_x + \hat{\rho}\hat{u}^2\delta u_x = \\
&(\gamma/(\gamma-1))\hat{u}\tau_2' - 1/(\gamma-1)\hat{\rho}\hat{u}^2\delta u_x + 0.5\hat{u}^3\delta\rho_x = \\
&(\gamma/(\gamma-1))\hat{u}\tau_2' - 1/(\gamma-1)\hat{\rho}\hat{u}^2\delta u_x + 0.5\hat{u}^2(\tau_1 - \hat{\rho}\nabla\cdot\delta\mathbf{U}).
\end{aligned}$$

Insert this expression into (40) and use the definitions $\hat{H} = (\gamma\hat{p}/(\gamma-1)\hat{\rho}) + 0.5\hat{u}^2$, the speed of sound $a^2 = \gamma\hat{p}/\hat{\rho}$, and the Mach number $M = \hat{u}/a$ to arrive at

$$\begin{aligned}
\nabla\cdot\delta\mathbf{U} - \tfrac{\hat{\rho}}{\gamma\hat{p}}\hat{u}^2\delta u_x &= (1-M^2)\delta u_x + \delta v_y + \delta w_z \\
&= -\hat{u}\tau_2'/\hat{p} + (\gamma-1)(\tau_5 - 0.5\hat{u}^2\tau_1)/(\gamma\hat{p}) = \tau_5'.
\end{aligned} \tag{41}$$

We have obtained an approximate error equation for the velocity components. For incompressible flow, $M = 0$ and this is the error equation derived from the continuity equation.

The error in the pressure fulfills an equation similar to (41). Take the derivatives of the three equations in (38) and (39) with respect to x, y and z, multiply the first equation by $1 - M^2$ and add them together. Then we have from (41)

$$\begin{aligned}
&(1-M^2)\delta p_{xx} + \delta p_{yy} + \delta p_{zz} = \\
&(1-M^2)\tau_{2x}' + \tau_{3y} + \tau_{4z} - \hat{\rho}\hat{u}((1-M^2)\delta u_x + \delta v_y + \delta w_z)_x = \\
&(1-M^2)\tau_{2x}' + \tau_{3y} + \tau_{4z} - \hat{\rho}\hat{u}\tau_{5x}' = \tau_p.
\end{aligned} \tag{42}$$

The equation for the numerical error in the pressure is elliptic for subsonic flow ($M < 1$) and hyperbolic for supersonic flow ($M > 1$).

If we ignore the boundaries of the computational domain, we can write the solution of (42) for $M < 1$ by means of the fundamental solution g of the Laplace operator [33]. First, introduce the change of variables

$$x = \beta x_1,\ y = y_1,\ z = z_1, \beta^2 = 1 - M^2, \tag{43}$$

and then let $\mathbf{r}$ be the vector

$$\mathbf{r} = (x_1 - \xi, y_1 - \eta, z_1 - \zeta)^T,\ r = |\mathbf{r}|.$$

In (42), τ_p has the form of a divergence $\tau_p = \nabla\cdot\tau_p'$. Suppose that τ_p' stays bounded as $r \to \infty$. The error in the pressure can now be written with $g = -1/(4\pi r)$

$$\delta p(x_1, y_1, z_1) = \int_\Omega \tau_p(\xi,\eta,\zeta) g\, d\Omega = -\int_\Omega \tau_p'\cdot\nabla g\, d\Omega + \int_{\partial\Omega} g\tau_p'\cdot\hat{n}\, dA, \tag{44}$$

using Gauss' theorem. Let $\partial\Omega$ expand so that the last term in (44) vanishes and $\Omega = \mathbf{R}^3$. Then we obtain

$$\delta p(x/\beta, y, z) = \frac{1}{4\pi}\int \frac{\tau_p'\cdot\mathbf{r}}{r^3} d\Omega. \tag{45}$$

The conclusion from (45) is that the influence of a discretization error $\tau_p'(\xi,\eta,\zeta)$ decays rapidly as $1/r^2$ when r grows.

In the hyperbolic case when $M > 1$ let $\beta^2 = M^2 - 1$ and use the same coordinate transformation as above. Then from (42)

$$\beta^2 \delta p_{xx} - \delta p_{yy} - \delta p_{zz} = -\tau_p. \tag{46}$$

With the fundamental solution to the wave equation (46), see [33], the pressure error is

$$\delta p(x/\beta, y, z) = -\frac{1}{2\pi}\int_\Omega \frac{\tau_p(\xi,\eta,\zeta)}{\sqrt{(x_1-\xi)^2 - ((y_1-\eta)^2 + (z_1-\zeta)^2)}} d\Omega, \tag{47}$$

where $\Omega = \{(\xi,\eta,\zeta) |\ x_1 - \xi > \sqrt{(y_1-\eta)^2 + (z_1-\zeta)^2}\}$. Thus, for fixed y and z the contribution from $\tau_p(\xi,\eta,\zeta)$ vanishes when x grows.

The analysis can be extended to include also boundaries by introducing Green's function for both $M < 1$ and $M > 1$.

3.3 Natural coordinates

The natural coordinates (s,n) in 2D are such that one coordinate s follows the streamlines and the other n is orthogonal to s. The transformation between the (x,y) and (s,n) systems is

$$\begin{aligned} dx &= u\,ds - v\,dn, \\ dy &= v\,ds + u\,dn. \end{aligned}$$

Note that $\mathbf{U}\cdot\nabla = \partial/\partial s$. Define the angle $\theta = \arctan(v/u)$. The speed of sound a is here given by $dp = a^2\,d\rho$ and the Mach number is $M = q/a$. Then the momentum equations in (24) and (32) can be replaced by two equations in the (s,n) coordinate system. In 2D, the steady, compressible flow is governed by

$$(1-M^2)p_s - \rho q^2\theta_n = u\tilde{\tau}_2 + v\tilde{\tau}_3 - q^2\tau_1 = \tilde{\tau}^s, \tag{48a}$$

$$p_n + \rho q^2\theta_s = -v\tilde{\tau}_2 + u\tilde{\tau}_3 = \tilde{\tau}^n, \tag{48b}$$

and from (28) and (30) we infer

$$\begin{aligned} S_s &= c_v(\gamma-1)(\tilde{\tau}_5/p - \tau_1/\rho), \\ H_s &= (\tau_5 - H\tau_1)/\rho. \end{aligned} \tag{49}$$

For incompressible flow, take $M = 0$ and $\rho = 1$ in (48) and replace (49) by

$$P_s = u\tilde{\tau}_2 + v\tilde{\tau}_3 + w\tilde{\tau}_4. \tag{50}$$

Divide both equations in (48) by ρq^2, differentiate (48a) with respect to s and (48b) with respect to n and compute the sum. With $\kappa = 1/(\rho q^2)$ the result is

$$((1-M^2)\kappa p_s)_s + (\kappa p_n)_n = (\kappa\tilde{\tau}^s)_s + (\kappa\tilde{\tau}^n)_n. \tag{51}$$

The equation is elliptic in p if the flow is subsonic and $M < 1$. If $M > 1$ as in supersonic flow, then (51) is hyperbolic. A similar equation is satisfied by θ. The other governing equations (49) or (50) are hyperbolic. The conclusion is that part of the solution error is transported along streamlines, as in (49) or (50), and part of the error is spread in an elliptic fashion (51) in subsonic flow. If the variation in p, q and ρ is small in (51), then subtract the analytical solution satisfying (51) without the error terms on the right hand side as in (10). For the error in the pressure δp we have approximately

$$(1 - M^2)\kappa\delta p_{ss} + \kappa\delta p_{nn} = (\kappa\tilde{\tau}^s)_s + (\kappa\tilde{\tau}^n)_n.$$

Under the assumptions in Sect 3.2 the coordinates s and n coincide with the x and y coordinates in (42) and the error equations are the same.

3.4 Flow in a channel

The compressible flow in a channel, see Fig. 1, is perturbed by a right hand side $\tau_2 = 1$ in a circle Ω_c with radius 0.2 at $(x_c, y_c) = (1.25, 1.5)$ simulating the effect of a discretization error in the momentum equation in the x-direction in (22b). The grid is uniform with 180×76 cells. The solution is computed at a subsonic speed, $M = 0.6$, and a supersonic speed, $M = 2$. The width of the channel is changed after $x = 1$ and a shock is generated by the lower wall at that point in the supersonic case.

The solution for $\tau_2 = 0$ is subtracted from the solution when $\tau_2 = 1$ and plotted for different variables in Fig. 1. The isolines of the difference illustrate how the discretization error at $(1.25, 1.5)$ generates errors in the solution variables. The behavior is different depending on the Mach number. An explanation is offered by the error equations.

Except for the shock, the flow in Fig. 1 satisfies the assumptions for the linear analysis in Sect. 3.2. The error equations in 2D follow from (37), (38), (39), (41), and (42)

$$\hat{u}\delta\rho_x + \hat{\rho}(\delta u_x + \delta v_y) = 0, \tag{52a}$$
$$\hat{\rho}\hat{u}\delta u_x + \delta p_x = \tau_2, \tag{52b}$$
$$\hat{\rho}\hat{u}\delta v_x + \delta p_y = 0, \tag{52c}$$
$$(1 - M^2)\delta p_{xx} + \delta p_{yy} = (1 + M^2(\gamma - 1))\tau_{2x}, \tag{52d}$$
$$(1 - M^2)\delta u_x + \delta v_y = -\hat{u}\tau_2/\hat{p}. \tag{52e}$$

Let $M < 1$ and $\beta = \sqrt{1 - M^2}$ and introduce the coordinate transformation (43). Following [33], the solution of (52d) in free space is

$$\delta p(x/\beta, y) = \sigma \int \tau_{2\xi} g(x_1 - \xi, y - \eta) d\Omega = -\sigma \int \tau_2 g_\xi(x_1 - \xi, y - \eta) d\Omega, \tag{53}$$

where g is the fundamental solution in 2D

$$g(x,y) = \frac{1}{4\pi}\log(x^2+y^2)$$

and $\sigma = (1+M^2(\gamma-1))/\beta$. Since the support of τ_2 is so small we have approximately

$$\delta p \approx \frac{\sigma\tau_2(x-x_c)}{2\pi\beta((x-x_c)/\beta)^2+(y-y_c)^2)}. \tag{54}$$

In subsonic flow, δp will decay at least as $1/r$ outside Ω_c.

In supersonic flow, let $\beta = \sqrt{M^2-1}$. The fundamental solution is

$$g(x,y) = \begin{cases} 1/2, & |y| < x \\ 0, & |y| \geq x \end{cases}$$

see [33]. Hence,

$$\begin{aligned} \delta p(x/\beta, y) &= -0.5\sigma \int_{-\infty}^{x_1} \int_{x_1-\xi>|y_1-\eta|} \tau_{2\xi} d\eta d\xi \\ &= -0.5\sigma \int_{-\infty}^{\infty} \int_{-\infty}^{x_1-|y_1-\eta|} \tau_{2\xi} d\xi d\eta = -0.5\sigma \int_{-\infty}^{\infty} \tau_2(x_1-|y_1-\eta|,\eta) d\eta. \end{aligned} \tag{55}$$

Outside the small circle Ω_c, $\tau_2 = 0$ and we have

$$\delta p(x,y) \approx -0.5\sigma\tau_2(x_c,y_c), \text{ when } x \geq x_c,\ x - x_c = \beta|y-y_c|. \tag{56}$$

Contrary to subsonic flow, δp will not decay outside Ω_c on two rays from (x_c, y_c). This behavior is also different from what we have in 3D in (47).

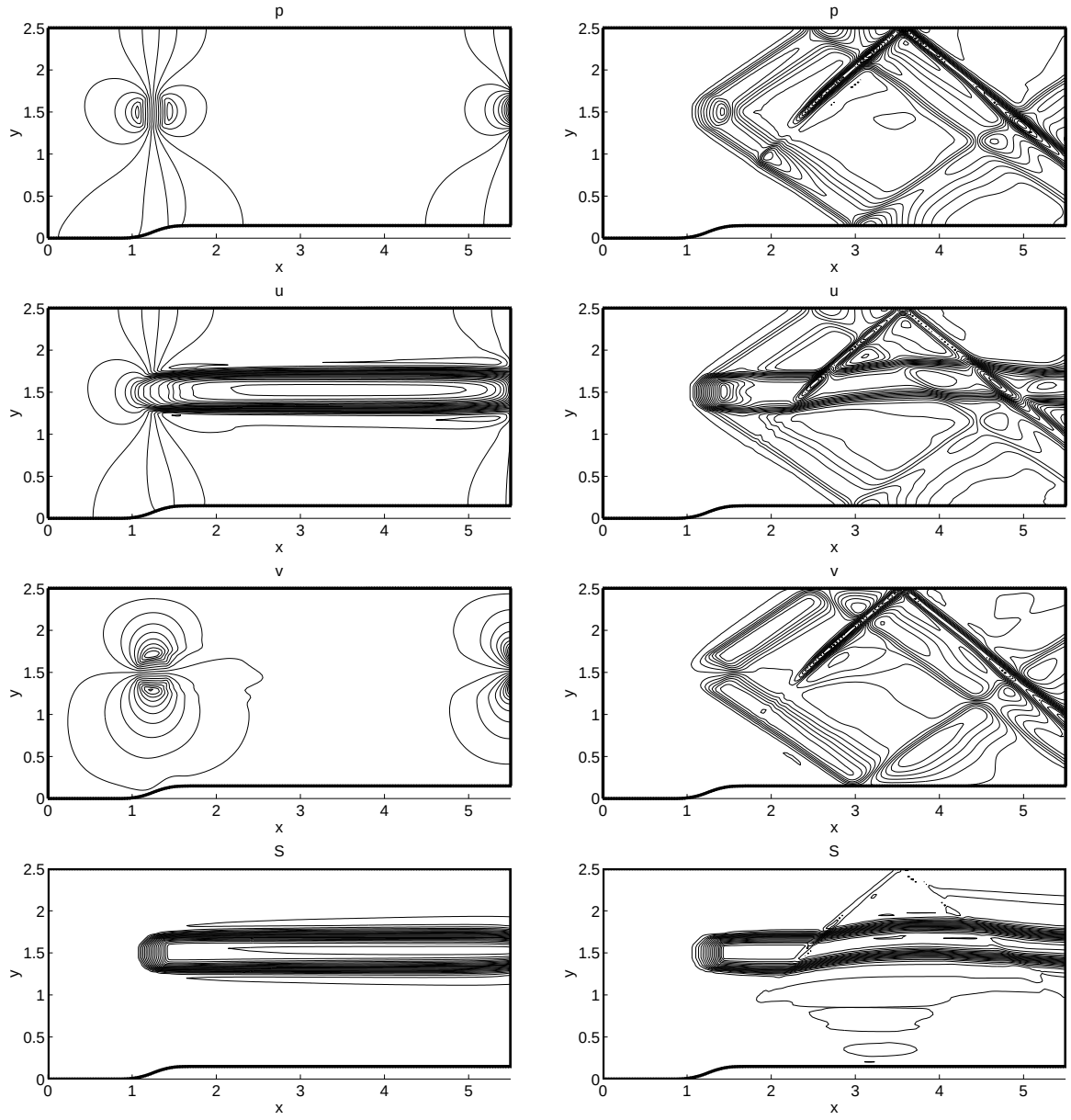


Figure 1: Isolines of the perturbation in the pressure p (upper), velocity component u (upper middle), velocity component v (lower middle), and entropy S (lower) for subsonic (left) and supersonic (right) flow.

From (52b) we conclude that

$$\delta u = (\int_{-\infty}^{x} \tau_2 d\xi - \delta p)/(\hat{\rho}\hat{u}). \tag{57}$$

The error δu consists of two parts: one part is convected downstream from the source τ_2 and the other part is spread like the error in p.

Take the x-derivative of (52c), the y-derivatives of (52b) and (52e) and combine to arrive at

$$(1 - M^2)\delta v_{xx} + \delta v_{yy} = -(1 + M^2(\gamma - 1))\tau_{2y}/(\hat{\rho}\hat{u}). \tag{58}$$

Then δv can be written as δp in (54) or (56).

An equation for $\delta\rho$ follows from (52a) and (52e)

$$\hat{u}\delta\rho_x + \hat{\rho}M^2\delta u_x = \hat{\rho}\hat{u}\tau_2/\hat{p}.$$

Thus, by (57)

$$\delta\rho = \frac{\hat{\rho}}{\gamma\hat{p}}((\gamma - 1)\int_{-\infty}^{x} \tau_2 d\xi + \delta p).$$

The error in ρ has one propagating part and one part depending on δp (cf. u).

By (28) and (29) the entropy error δS is

$$\delta S = -\frac{c_v(\gamma - 1)}{\hat{p}}\int_{-\infty}^{x} \tau_2 d\xi. \tag{59}$$

The solutions for δp, δu, δv, and δS in (54), (56), (57), and (59) agree with the solutions in Fig. 1 locally around (x_c, y_c) when the influence of the boundaries is negligible. At the supersonic speed, the waves are reflected in the channel walls and the shock position is shifted above $y \approx y_c$ and in the reflected part by the convected error. The perturbations in δp and δv for $M = 0.6$ at the downstream boundary is explained by the outflow boundary conditions there.

4 Numerical algorithms

In this section we describe the space discretization briefly and how the grid is refined given the computed discretization errors.

4.1 Space discretization

The flow equations are solved on a structured grid in 2D. The computational domain is partitioned into a number of blocks. The location of these blocks is predetermined and the edges of a block follow grid lines. The grids in each block can be refined or coarsened. At the block boundaries, jumps are allowed in the grid size. The step length along a boundary may increase by a factor two when crossing the boundary. By changing all cells in a block, the administration of the adaptivity is simplified and the interpolation of data between cells of different size is concentrated at the block boundaries. There is a waste of cells since all of them are not needed for the accuracy. In [4], the refined patches can be

placed without restrictions, but the bookkeeping is more complicated in such an algorithm. Every cell in [13] can be refined. This approach requires an extra data structure and second order accuracy is difficult to obtain.

The flux vectors in (4) are computed either with the Jameson scheme [23] or with the Osher scheme [29]. Both are second order accurate on Cartesian grids with constant step sizes in each direction.

Boundary conditions are approximated by introducing ghost cells. In the steady state solutions, the variables in these cells are either given by the boundary data in (1) or by extrapolation from the interior of the domain. The order of the extrapolation is important for the accuracy of the solution, in particular at solid walls. For the compressible equations, the extra numerical conditions are calculated according to [11].

For characteristics starting at a boundary going into the domain it is sufficient to have the boundary condition satisifed to order $\nu - 1$ if ν is the order of the discretization in the interior without losing accuracy [20]. Along a solid wall we have a streamline in the Euler equations and the behavior of the error in the total pressure or the entropy is governed by (34) and (28). If the error in the boundary conditon is of order $\nu - 1$, then the error in P or S will also be of that order.

Interpolation is necessary at block faces where there is an increase in the grid size. In the coarse ghost cells, the average is computed by summing the averages in the corresponding fine cells with weights proportional to the fine cell areas. The fine ghost cells are computed by combining the neighboring fine and coarse cell values, so that the discretization error is of second order on Cartesian grids in the fine cells adjacent to the boundary.

The steady state solution of incompressible flow is computed by adding $\partial p/\partial t$ to the first equation of (31) as in [8]. Then the time-dependent equations are integrated for both compressible and incompressible flow by a Runge-Kutta scheme until the time derivatives vanish [23].

4.2 Adaptation algorithm

The solution on the original grid is computed first. Then that solution is used to compute the solution error with the estimates in proposition 2.1 and the error equations in Sect. 3. The choice of grid size in the different blocks of the computational domain is determined by these estimates.

Assume that a maximum error is required to be below a certain tolerance ϵ in a variable, whose error is propagated along streamlines as in (28), (30), (35). Let s denote the coordinate along the streamlines. They pass through a number of block interfaces at $s = s_0, s_1, s_2 \ldots$. The contribution to the error in block k is $I_k(s) = \int_{s_{k-1}}^{s} \tau \, ds$ for $s_{k-1} \leq s \leq s_k$. The accumulated error is denoted by $E(s)$ with $E(s_0) = 0$, and in block k

$$E(s) = E(s_{k-1}) + I_k(s) \quad \text{for} \quad s_{k-1} \leq s \leq s_k.$$

The algorithm for choosing the grid size in the blocks along a streamline is initialized by calculating the integrals I_k and the corresponding estimate of the accumulated error $E(s)$. Then the adaptive algorithm is with $\nu = 2$

$$
\begin{array}{l}
\textbf{while } \max|E(s)| > \epsilon \\
\quad \text{Locate the maximum of } |E(s)| : \ s_{M-1} < s_{max} < s_M \\
\quad I_k^+(s) = I_k(s)\,\text{sign}\,(E(s_{max})) \text{ for all } k \\
\quad \text{Find the block } k_m \text{ with the maximum of} \\
\qquad I_1^+(s_1), I_2^+(s_2), \ldots, I_{M-1}^+(s_{M-1}), I_M^+(s_{max}) \\
\quad \text{Refine block } k_m : \ I_{k_m}(s) := I_{k_m}(s)/2^\nu \\
\quad \text{Update } E(s) \\
\textbf{end while}
\end{array}
\tag{60}
$$

The error may increase in occasional steps in the process when the maximum error changes sign. No jumps in grid size at the block boundaries are allowed to be larger than a factor two. To fulfill this condition, the grid sometimes has to be refined in some blocks to an unnecessarily fine level. If the error along a number of streamlines in a block is controlled, then the smallest grid size given by the algorithm applied to each streamline is chosen.

5 Numerical results

The steady state solution of the Euler equations of inviscid flow is computed around two two-dimensional objects: The upper part of a cylinder and the wing profile NACA 0012. The flow around the half-cylinder is incompressible and the numerical solution is compared to the analytical solution of the linearized potential flow equation. For the second geometry, the flow is compressible in a standard transonic test case and a subsonic case. The enthalpy and the entropy in the numerical solution are compared to the exact constant value of the enthalpy and to the exact entropy, which is constant at least in parts of the domain.

5.1 Incompressible flow

The time-independent solution of the incompressible Euler equations ((31) with $\tilde{\tau} = 0$) is computed in 2D around the upper part of a cylinder. The grid is cylindrical and partitioned into blocks. The system of equations is discretized by the scheme of Jameson, Schmidt, and Turkel [23]. Fourth order extrapolation at the boundaries is used to supply missing numerical boundary conditions at the cylinder.

The solution is compared to the analytical solution of the linearized potential flow equation. Let Θ be the potential and n the normal direction on the cylinder.

Then

$$\begin{aligned} &\Delta\Theta = 0 \text{ in } \Omega,\ \mathbf{U} = \nabla\Theta,\ P = p + 0.5q^2 = \text{const}, \\ &\mathbf{U} = (1,0)^T \text{ at } \infty, n \cdot \mathbf{U} = 0 \text{ on cylinder.} \end{aligned} \tag{61}$$

With the origin at the center of the cylinder and r the radius, the solution to (61) is

$$u = 1 + r^2(y^2 - x^2)/(x^2 + y^2)^2,\ v = -2r^2xy/(x^2 + y^2)^2. \tag{62}$$

If there is no vorticity in the Euler solution, then (62) is the steady state solution of (31) with $\tilde{\tau} = 0$.

The analytical solution (62) defines the boundary conditions at the outer boundary. Missing numerical boundary conditions at the cylinder are determined by fourth order extrapolation.

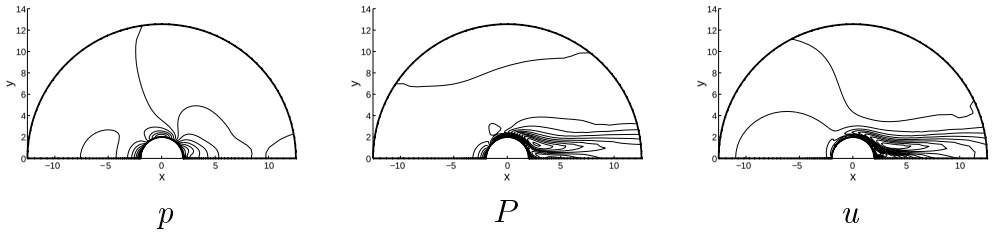


Figure 2: Error distribution around the cylinder.

The difference between the analytical solution and the Euler solution around the cylinder on a uniform grid is plotted in Fig. 2. The error propagation downstream can as expected be observed in the variables u and P. Most of the error in p is generated at the surface of the cylinder and it decays in an elliptic way when we go away from the cylinder.

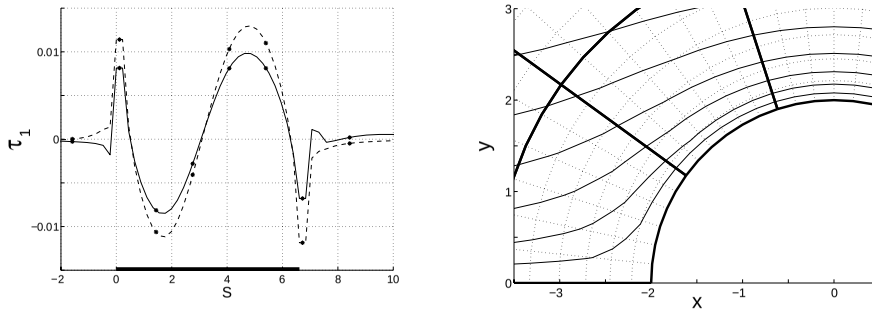


Figure 3: The discretization error τ_1 used for the error estimation of P (left) and streamlines close to the cylinder (right).

.

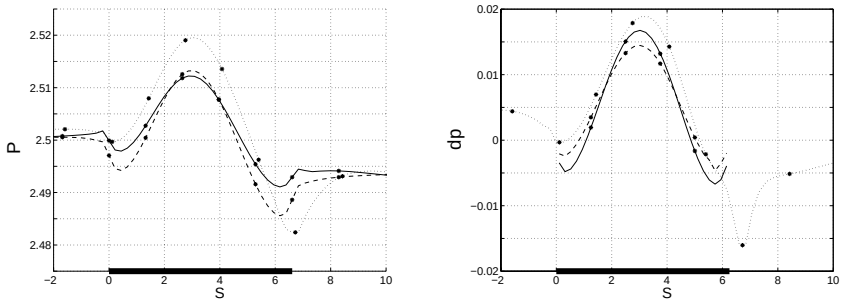


Figure 4: Estimated (dashed and solid) and true values (dotted) of P (left) and δp (right) along a streamline close to the surface of the cylinder.

In Fig. 4, the errors δP and δp are estimated on a streamline close to the cylinder ($s \in [0, 2\pi]$) and to $y = 0$ in front of ($s < 0$) and behind the cylinder ($s > 2\pi$). The numerical solution is computed on a coarse grid with 30×18 cells divided into 15 blocks. For the total pressure P, (35) is integrated with $P = 2.5$ as initial condition with two different right hand sides. Firstly, the discretization error is computed by inserting the analytical solution into the discretization as in (9). Secondly, τ is estimated by Proposition 2.1. The difference in τ_1 between the two alternatives is shown in Fig. 3. The first (dashed) and the second (solid) estimates are close. The difference is comparatively small in Fig. 4 between the first (dashed) and the second (solid) right hand sides and the true value of P (dotted). The error in P is somewhat underestimated. This is partly explained by the fact that the estimates are obtained via integrations along the line of symmetry $y = 0$ and the surface of the cylinder. As we can see in Fig. 3, this line does not follow the streamline at the left stagnation point and an error there cannot be corrected downstream. The error in p in Fig. 4 is calculated by integrating (48b) from the outer boundary with initial condition $p = 0$ assuming $\rho q^2 \theta_s$ to be negligible.

We let the error tolerance be $\epsilon = 4.3 \cdot 10^{-3}$ and apply the adaptive algorithm in Sect. 4.2 once to our example. The grid when P is controlled so that $|\delta P| \leq \epsilon$ is plotted in Fig. 5, along with obtained errors on that grid. The abscissa follows the streamline along the cylinder as in Figs. 3 and 4. The cylinder surface is marked by a thick black line. The error is somewhat underestimated in P.

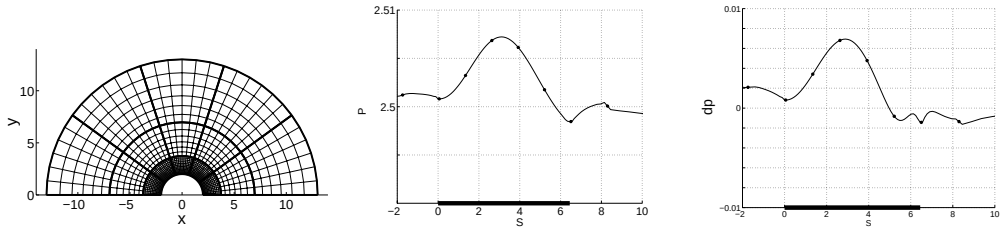


Figure 5: Grid for control of P and the measured values of P and δp.

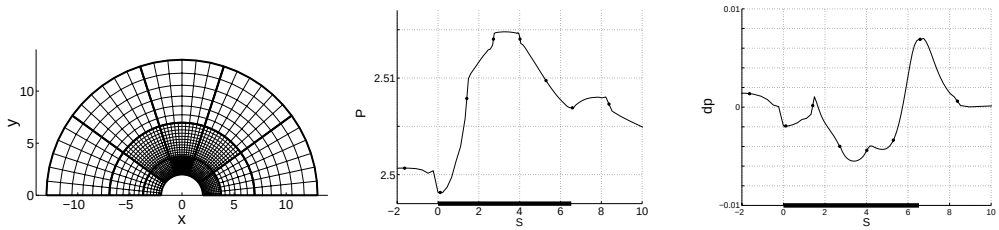


Figure 6: Grid for control of p and the measured values of P and δp.

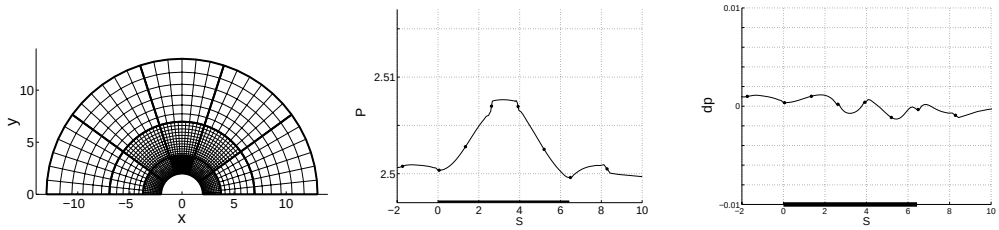
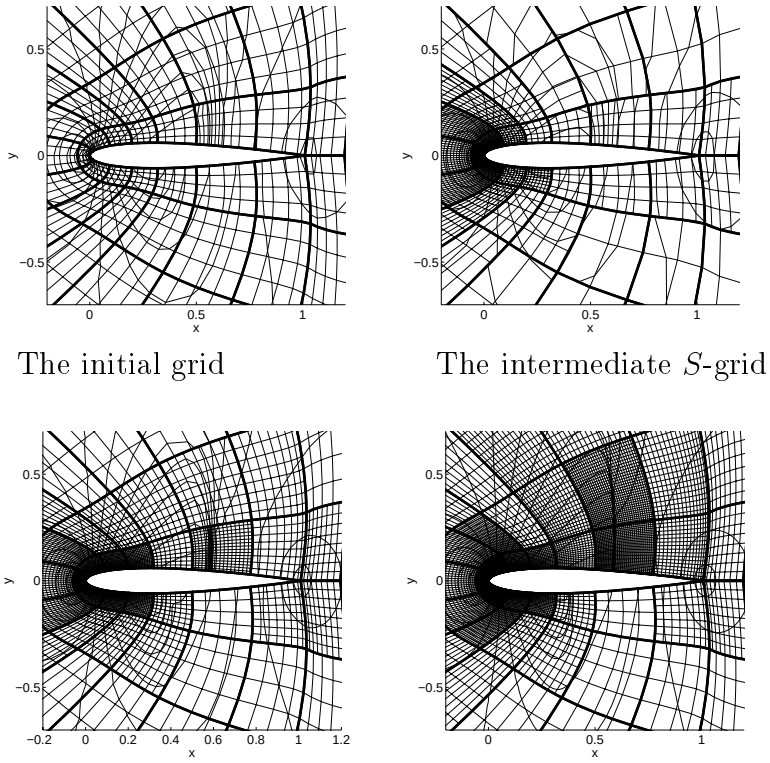


Figure 7: Grid for control of p and P and the measured values of P and δp.

The grid for control of the error of p is generated by (60) and integration of (48b) and is plotted in Fig. 6. The coarse block in the left part of the cylinder gives rise to a large error in P, which is convected downstream resulting in a peak in δp at the right stagnation point. A reliable estimate of δp requires control of P, see Fig. 7, where the goal is $\max(\delta P, \delta p) \leq \epsilon$. To reduce δP, it does not help to refine the grid in the top block on the cylinder, where δP has its maximum. Instead, the grid should be finer, where δP grows most rapidly.

5.2 Compressible flow

The steady state solution of the compressible Euler equations ((22) with $\tau = 0$) is calculated around the NACA 0012 airfoil. An original C-grid is partitioned into blocks and refined with the adaptive algorithm in Sect. 4.2. The equations are discretized with the Osher scheme [29]. The error equations in Sect. 3 break down at discontinuities such as shocks but the adaptive algorithm (60) still works well.



The initial grid | The intermediate S-grid

The intermediate H-grid | The fine H-grid

Figure 8: Isobars for transonic flow on the initial, intermediate and fine grids. Every second grid line is omitted in the plot of the fine H-grid.

In the first example, the flow is transonic with $M = 0.8$ and the angle of attack $\alpha = 1.25^o$ and $\epsilon = 5 \cdot 10^{-2}$ in (60). The adaptation algorithm (60) is applied twice: first on the original grid to generate an intermediate grid and then again to generate the final grid. Either the error in the enthalpy H or the error in the entropy S is monitored. The equations (28) and (30) are integrated for the errors in H and S along the streamlines on the upper and lower surfaces of the airfoil. Then the change of H or S in a block determines the grid refinement in (60). It is important that the refinement is based on the difference of H or S between the block boundaries and not the peak values of them in the blocks (cf. Figs. 5,6,7).

The initial grid and the intermediate grids generated from δH and δS are shown in Fig. 8 together with the isobars of the solution. The estimated error in S is not sufficiently large to motivate a refinement at the shock on the upper part of the profile. On the contrary, δH captures the shock on the upper and lower side and the algorithm (60) introduces grid refinement both at the leading edge and the shocks. The importance of a good resolution at a shock is discussed in [15] and [16]. Errors of low order are created in a shock and convected downstream.

Fig. 9 and Fig. 10 display the computed δH and S on the upper and lower side of the airfoil with the initial (dotted), intermediate (dashed) and fine (solid) H-grids. The plots are extended to the outer boundary, and the wing is indicated by a thick line. The entropy has a jump at the shock on the upper side. In the wake, an average is taken between S above and below the trailing edge.

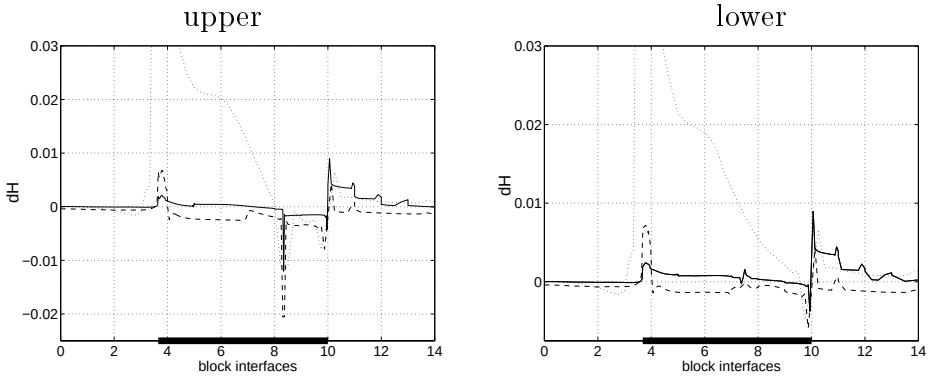


Figure 9: Error in the enthalpy along the streamlines at the surface of the wing profile.

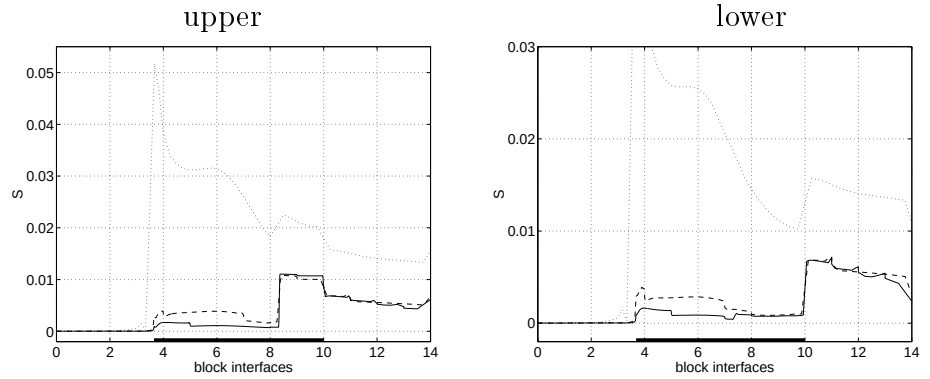
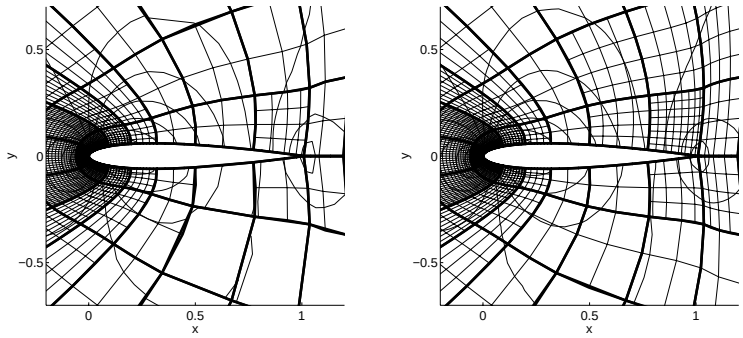


Figure 10: The entropy along the streamlines at the surface of the wing profile.

Subsonic flow is computed in the second example with $M = 0.5$, $\alpha = 1.25^o$, and $\epsilon = 2 \cdot 10^{-3}$. The initial grid is the same as in Fig. 8. The results after applying the algorithm (60) twice are found in Fig. 11. The grids are generated using the enthalpy or the entropy as a measure of the error. The enthalpy is more sensitive to the grid density at the trailing edge. The errors are compared on the two grids in Fig. 12, where the solid line corresponds to the H-grid and the dashed line to the S-grid.

Acknowledgment

Financial support has been obtained from the Swedish Research Council for Engineering Science and the Swedish Foundation for Strategic Research.



S-grid H-grid

Figure 11: Isobars for subsonic flow on the final grids. Every second grid line is omitted.

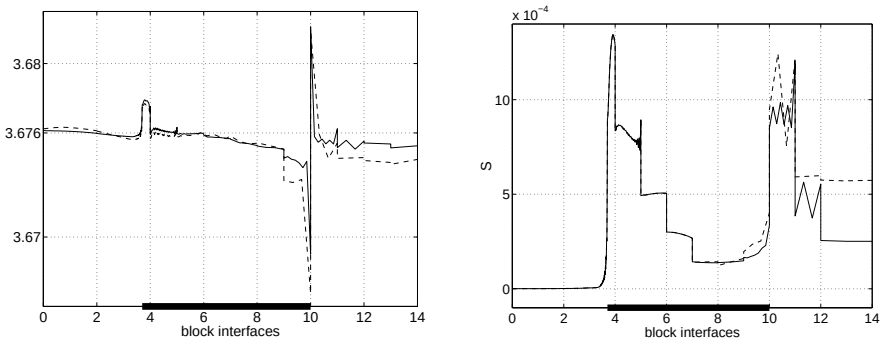


Figure 12: The enthalpy (left) and entropy (right) along the lower side of the wing profile.

References

[1] R. Abgrall, On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation, *J. Comput. Phys.*, 114 (1994), p. 45–58.

[2] J. D. Anderson, Jr, *Modern Compressible Flow*, 2nd ed., McGraw-Hill, New York, 1990.

[3] J. Bell, M. Berger, J. Saltzman, M. Welcome, Three-dimensional adaptive mesh refinement for hyperbolic conservation laws, *SIAM J. Sci. Comput.*, 15 (1994), p. 127–138.

[4] M. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics, *J. Comput. Phys.*, 82 (1989), p. 64–84.

[5] M. Berger, A. Jameson, Automatic adaptive grid refinement for the Euler equations, *AIAA J.*, 23 (1985), p. 561–568.

[6] C. J. Budd, G. P. Koomullil, A. M. Stuart, On the solution of convection-diffusion boundary value problems using equidistributed grids, *SIAM J. Sci. Comput.*, 20 (1998), p. 591–618.

[7] J. Casper, H. L. Atkins, A finite-volume high-order ENO scheme for two-dimensional hyperbolic systems, *J. Comput. Phys.*, 106 (1993), p. 62–76.

[8] A. J. Chorin, A numerical method for solving incompressible viscous flow problems, *J. Comput. Phys.*, 2 (1967), p. 12–36.

[9] S. D. Connell, D. G. Holmes, Three-dimensional unstructured adaptive multigrid scheme for the Euler equations, *AIAA J.*, 32 (1994), p. 1626–1632.

[10] P. Colella, M. R. Dorr, D. D. Wake, Numerical solution of plasma fluid equations using locally refined grids, *J. Comput. Phys.*, 152 (1999), p. 550–583.

[11] A. Dadone, B. Grossman, Surface boundary conditions for the numerical solution of the Euler equations, *AIAA J.*, 32 (1994), p. 285–293.

[12] R. L. Davis, J. F. Dannenhoffer, III, 3-D adaptive grid-embedding Euler technique, AIAA paper 93-0330, 1993.

[13] D. De Zeeuw, K. G. Powell, An adaptively refined Cartesian mesh solver for the Euler equations, *J. Comput. Phys.*, 104 (1993), p. 56–68.

[14] J. De Keyser, K. Lust, D. Roose, Run-time balancing support for a parallel multiblock Euler/Navier-Stokes code with adaptive refinement on distributed memory computers, *Parallel Comput.*, 20 (1994), p. 1069–1088.

[15] G. Efraimsson, G. Kreiss, A remark on numerical errors downstream of slightly viscous shocks, *SIAM J. Numer. Anal.*, 36 (1999), p. 853–863.

[16] B. Engquist, B. Sjögreen, The convergence of finite difference schemes in the presence of shocks, *SIAM J. Numer. Anal.*, 35 (1998), p. 2464–2485.

[17] K. Eriksson, D. Estep, P. Hansbo, C. Johnson, Introduction to adaptive methods for differential equations, *Acta Numerica*, (1995), p. 121–123.

[18] L. Ferm, P. Lötstedt, Blockwise adaptive grids with multigrid acceleration for compressible flow, *AIAA J.*, 37 (1999), p. 121–123.

[19] M. Gerritsen, P. Olsson, Designing an efficient solution strategy for fluid flows II. Stable high-order central finite difference schemes on composite adaptive grids with sharp shock resolution, *J. Comput. Phys.*, 147 (1998), p. 293–317.

[20] B. Gustafsson, The convergence rate for difference approximations to mixed initial boundary value problems, *Math. Comput.*, 29 (1975), p. 396–406.

[21] E. Hairer, S. P. Nørsett, G. Wanner, *Solving ordinary differential equations*, 2nd ed., Springer-Verlag, Berlin, 1993.

[22] R. D. Hornung, J. A. Trangenstein, Adaptive mesh refinement and multilevel iteration for flow in porous media, *J. Comput. Phys.*, 136 (1997), p. 522–545.

[23] A. Jameson, W. Schmidt, E. Turkel, Numerical solutions of the Euler equations by finite volume methods using Runge-Kutta time-stepping schemes, AIAA-paper 81-1259, 1981.

[24] S. Li, L. Petzold, Y. Ren, Stability of moving mesh systems of partial differential equations, *SIAM J. Sci. Comput.*, 20 (1998), p. 719–738.

[25] F. Liu, S. Ji, G. Lao, An adaptive grid method and its application to steady Euler flow calculations, *SIAM J. Sci. Comput.*, 20 (1998), p. 811–825.

[26] P. Lötstedt, S. Söderberg, A. Ramage, L. Hemmingsson-Frändén, Implicit solution of hyperbolic equations with space-time adaptivity, in preparation.

[27] S. Muzaferija, D. Gosman, Finite-volume CFD procedure and adaptive error control strategy for grids of arbitrary topology, *J. Comput. Phys.*, 138 (1997), p. 776–787.

[28] W. L. Oberkampf, F. G. Blottner, Issues in computational fluid dynamics code verification and validation, *AIAA J.*, 36 (1998), p. 687–695.

[29] S. Osher, F. Solomon, Upwind schemes for hyperbolic systems of conservation laws, *Math. Comp.*, 38 (1982), p. 339–377.

[30] N. G. Pantelelis, A. E. Kanarachos, The parallel block adaptive multigrid method for the implicit solution of the Euler equations, *Int. J. Numer. Meth. Fluids*, 22 (1996), p. 411–428.

[31] K. G. Powell, A tree-based adaptive scheme for solution of the equations of gas dynamics and magnetohydrodynamics, *Appl. Numer. Math.*, 14 (1994), p. 327–352.

[32] K. G. Powell, P. L. Roe, T. J. Linde, T. I. Gombosi, D. L. De Zeeuw, A solution-adaptive upwind scheme for ideal magnetohydrodynamics, *J. Comput. Phys.*, 154 (1999), p. 284–309.

[33] M. Renardy, R. C. Rogers, *An Introduction to Partial Differential Equations*, Springer-Verlag, New York, 1993.

[34] P. Roache, Quantification of uncertainty in computational fluid dynamics, *Annu. Rev. Fluid Mech.*, 29 (1997), p. 123–160.

[35] G. Söderlind, The automatic control of numerical integration, *CWI Quarterly*, 11 (1998), p. 55–74.

[36] K. Srinivasan, S. G. Rubin, Solution-based grid adaptation through segmented multigrid domain decomposition, *J. Comput. Phys.*, 136 (1997), p. 467–493.

[37] J. Wu, H. Ritzdorf, K. Oosterlee, B. Steckel, A. Schüller, Adaptive parallel multigrid solution of 2D incompressible Navier-Stokes equations, *Int. J. Num. Meth. Fluids*, 24 (1997), p. 875–892.