Control Errors in CFD!

Bernhard Müller

Department of Scientific Computing,
Information Technology, Uppsala University,
Box 337, S-751 05 Uppsala, Sweden,
E-mail `bernd@tdb.uu.se`, URL
`http://www.tdb.uu.se/~bernd/`

Abstract
Error control in computational fluid dynamics (CFD) has been crucial for reliability and efficiency of numerical flow simulations. The roles of truncation and rounding errors in difference approximations are discussed. Truncation error control is reviewed for ODEs. For difference approximations of PDEs, discretization error control by Richardson extrapolation is outlined. Applications to anisotropic grid adaptation in CFD are shown. Alternative approaches of error control in CFD are mentioned.

1 Introduction

Computational fluid dynamics (CFD) has been maturing using wind tunnel measurements for validation. However, experiments are often not available, in particular in new multidisciplinary applications. Therefore, the assessment of mathematical models has become crucial. Thus, numerical errors in the discretization of mathematical models have to be minimized. Error control is essential for the reliability of CFD. Moreover, error control is the prerequisite for adaptive methods, which are decisive for efficiency.

Eriksson et al. [2] cite the philosopher Ludwig Wittgenstein (1889 - 1951):

"On what you cannot compute with error control, you must be silent."

Since Wittgenstein was an aeronautical engineer before turning to philosophy, he was well aware of the need for reliable computations in aeronautics.

2 Truncation and Rounding Errors

Let us first consider the approximation of the first derivative $\frac{df(x)}{dx}$ of a given function $f(x)$ by finite differences. As an example, we take the second-order central difference approximation

$$D_h f(x) \; = \; \frac{f(x+h) - f(x-h)}{2h} \qquad (1)$$

where h is the step size. Inserting the Taylor series expansions around x into (1), we obtain the truncation error

$$T_h(x) \; := \; D_h f(x) - \frac{df(x)}{dx} \; = \; \frac{f'''(x)}{6} h^2 + \mathcal{O}(h^4) \qquad (2)$$

showing the second-order accuracy.

The truncation error indicates how to get higher accuracy:

- h-refinement
  The grid size h is reduced. For example, if the grid size is halved, i.e. $\frac{h}{2}$ is used instead of h, the truncation error of the second-order central difference approximation is reduced by a factor of $\frac{1}{2^2} = \frac{1}{4}$, i.e. $T_{\frac{h}{2}}(x) \approx 2^{-2} T_h(x)$.

- p-refinement
  The order of accuracy p is increased. E.g. instead of the second-order central difference approximation with $p = 2$, we can use a higher order method, e.g. $p = 4$, for which the truncation error is lower.

For a difference method of order p, the truncation error will be reduced by a factor of 2^{-p}, when the grid size is halved. Thus, the accuracy is the more increased by grid refinement the larger p is. However, no matter what the order of p is, we cannot see the expected error reduction as $h \to 0$. Then, we even observe an increase of the error. The reason for the error increase as $h \to 0$ is caused by rounding errors, which lead to cancellation errors. The errors of second-, fourth- and sixth-order central difference approximations of $\frac{dexp(0)}{dx}$ are shown in

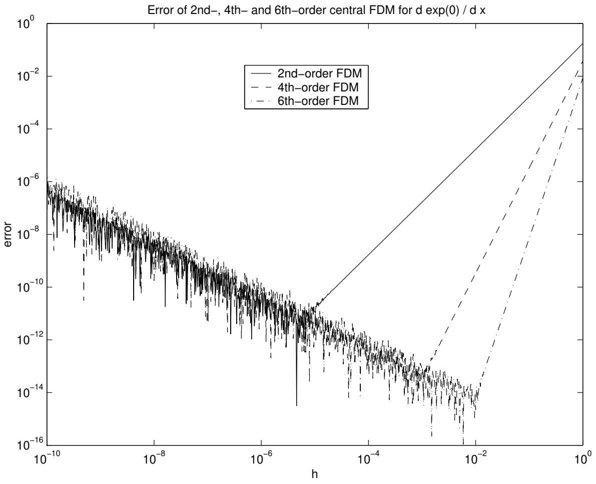

Figure 1: Error of second-, fourth- and sixth-order central difference approximations of $\frac{dexp(0)}{dx}$ in double precision.

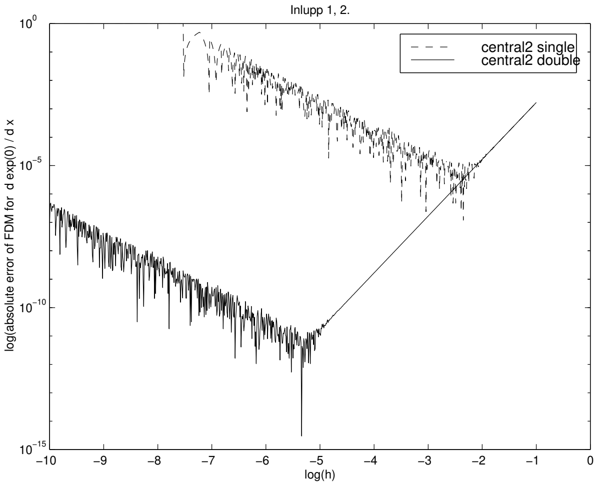

Figure 2: Error of second-order central difference approximation of $\frac{dexp(0)}{dx}$ with single and double precision.

Fig. 1. The finite differences were computed in double precision.

The error of $D_h exp(0)$, the second-order central difference approximation of $\frac{dexp(0)}{dx}$, is shown in Fig. 2. The influence of single and double precision on the rounding error is clearly visible. Any real number u is represented as a floating point number $fl(u)$ on the computer. The following relation holds:

$$fl(u) = u(1+\epsilon)$$

where

$$|\epsilon| \leq \epsilon_m \approx \begin{cases} 10^{-7} & \text{for single precision} \\ 10^{-16} & \text{for double precision} \end{cases}$$

where ϵ_m is machine epsilon. When computing the second-order central difference approximation, the dominant rounding error is the cancellation error caused by computing the difference of almost equal numbers. Neglecting the error caused by the floating point division by $2h$, we get on the computer instead of (1)

$$\begin{aligned} D_h fl(f(x)) &= \frac{fl(f(x+h)) - fl(f(x-h))}{2h} \\ = (D_h f(x))[1 &+ \frac{f(x+h)}{\Delta f}\epsilon^+ - \frac{f(x-h)}{\Delta f}\epsilon^-] \end{aligned}$$

where $\Delta f = f(x+h) - f(x-h)$ and $\epsilon^\pm$ the relative error of $fl(f(x \pm h))$. We see that the relative cancellation error is large, if $\frac{f(x\pm h)}{\Delta f}$ are large and cannot be reduced sufficiently by multiplication by $\epsilon^\pm$, for which $|\epsilon^\pm| \leq \epsilon_m$ holds. Thus, cancellation errors and rounding errors in general can be better kept under control by using double precision than by single precision. However, even with double precision, we should be aware of rounding errors, which can pollute numerical results, cf. Fig.s 1 and 2.

In CFD, cancellation errors occur, when we are dealing with flows, for which the changes are minute compared with the mean flow. An example is the computation of compressible low Mach number flow. When solving the compressible Euler and Navier-Stokes equations for Mach numbers $M \ll 1$, the thermodynamic quantities like density ρ, pressure p and temperature T only change very slightly compared with their stagnation values. The consequence is that the computation of e.g. ∇p can lead to large cancellation errors, which make the results useless. Fig. 3 shows the expansion wave in a tube computed with the axisymmetric Navier-Stokes equations at low Mach numbers [12], [9]. The re-

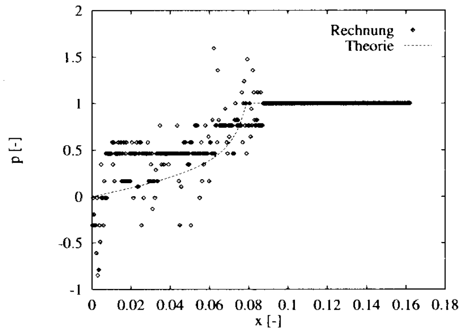

Figure 3: Erroneous pressure distribution of an expansion fan in a tube computed with conventional formulation of the Navier-Stokes equations leading to cancellation errors in single precision at $M = O(10^{-6})$ [12], [9], $\diamond$: computed result, $---$: analytical solution.

sults with single precision are completely erroneous owing to cancellation errors. However, if the Navier-Stokes equations are formulated in terms of the perturbation variables with respect to stagnation variables like $\rho' = \rho - \rho_0$, the cancellation errors can be minimized [12], [9]. Computations at $M = O(10^{-10})$ can be performed with single precision using the perturbation formulation, which is mathematically equivalent to the original Navier-Stokes equations but formulated in a form better suited for computation, cf. [12], [9]. You simply solve for e.g. density change instead of density with the same conservation laws.

3 Truncation Error Control for ODEs

Let us consider an initial value ordinary differential equation (ODE)

$$\begin{cases} y' = f(x,y) \,, & x > 0 \\ y(0) = y_0 \end{cases} \tag{3}$$

where x denotes time and $y = y(x)$ is the unknown function to be determined. We want to control the step size h of a numerical method to solve ODE (3) based on the local truncation error. We shall consider Richardson extrapolation and Runge-Kutta-Fehlberg methods.

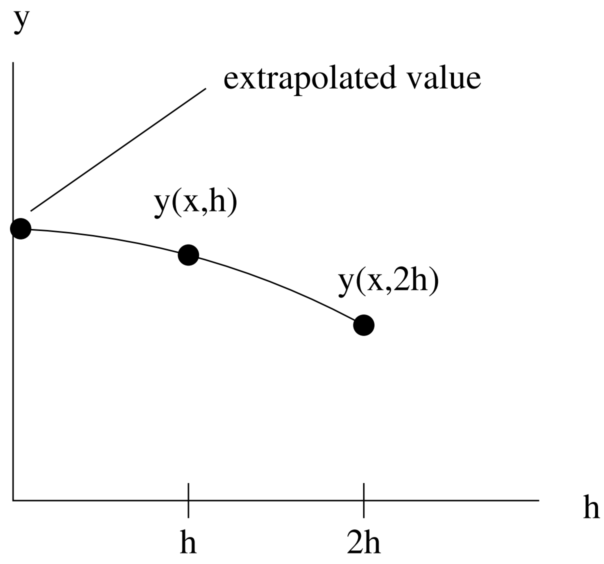

Figure 4: Basic idea of Richardson extrapolation.

3.1 Richardson Extrapolation

The Richardson extrapolation is a general method to improve the accuracy of numerical results. The basic idea of extrapolating results with step sizes h and $2h$ to $h = 0$ can be summarized as follows, cf. Fig. 4:

- A method of order p is used, meaning that we know the theoretical behaviour of the truncation error as $h \to 0$.
- Compute a solution in x_k using step size h: $y(x_k, h)$
- Compute a solution in x_k using step size $2h$: $y(x_k, 2h)$
- Given two values with different error and knowing that the truncation error behaves as $\mathcal{O}(h^p)$ it is possible to extrapolate an approximation of the exact value $y(x_k, 0)$.

For a method of order p, the exact value (solution) can be expressed as

$$y(x_k) = y(x_k, h) + ch^p + \mathcal{O}(h^q) \,,$$

where c is a constant and p and q are integers with $p < q$. Solving the problem with step sizes h and $2h$ yields

$$
\begin{aligned}
y(x_k) &= y(x_k, h) + ch^p + \mathcal{O}(h^q) \\
y(x_k) &= y(x_k, 2h) + c(2h)^p + \mathcal{O}(h^q)
\end{aligned}
$$

Subtracting these equations, we obtain the leading term of the truncation error

$$
ch^p = \frac{y(x_k, h) - y(x_k, 2h)}{2^p - 1} + \mathcal{O}(h^q) \ . \tag{4}
$$

Thus, adding the leading error term to $y(x_k, h)$, we obtain a better approximation of the exact solution $y(x_k)$, because

$$
y(x_k) = y(x_k, h) + \frac{y(x_k, h) - y(x_k, 2h)}{2^p - 1} + \mathcal{O}(h^q)
$$

The algorithm for truncation error control based on Richardson extrapolation becomes:

1. Compute y_k using a method of order p with step sizes h and $2h$.

2. Estimate the truncation error by the leading term ch^p with Richardson extrapolation (4).

3. If error $ch^p < tol$ where tol is a prescribed tolerance, accept y_k. Compute new step size h_{k+1}, where $h_{k+1} > h_k$.

4. If error $ch^p > tol$, discard y_k and compute new h_k less then the previous one. Compute y_k. Continue until error $ch^p < tol$.

If the truncation error is $T_h(x_k) \approx ch^p$, we choose the new step size h_{k+1} such that the estimated truncation error is approximately equal to the desired tolerance tol, i.e. $T_h(x_{k+1}) \approx tol$. Thus, the new step size is computed from

$$
h_{k+1} = \left(\frac{\Theta\, tol}{T_h(x_k)} \right)^{1/p} h_k \ ,
$$

where $\Theta < 1$ is a safety factor.

3.2 Runge-Kutta-Fehlberg Methods

Instead of computing $y(x_k, h)$ with two different step sizes, one can compute $y(x_k, h)$ with two methods having different orders of accuracy. The difference of the two results can be used to estimate the truncation error. The computational work is not doubled, if Runge-Kutta-Fehlberg methods are used. For example, 7 stages are performed and two different weightings of the stages yield two methods of orders 4 and 5, respectively. Then, the truncation error can be estimated by the difference of the two results. This error control is e.g. used in the MATLAB ODE-routine `ode45`.

4 Discretization Error Control for PDEs

Consider an initial boundary value problem for a partial differential equation (PDE) like the Navier-Stokes equations

$$
Lu = 0 \ , \tag{5}
$$

where L denotes the differential operator and $u = u(x, t)$ is the unknown function to be determined. Here, x and t denote the spatial coordinate and time, respectively. As an example, we consider the scalar linear 1D advection equation with initial and boundary conditions

$$
\begin{aligned}
u_t + au_x &= 0 \ , \qquad x \in [0, 1], t > 0, \quad (6) \\
u(x, 0) &= f(x) \ , \quad x \in [0, 1], \\
u(0, t) &= g(t) \ , \qquad t \geq 0,
\end{aligned}
$$

where $a > 0$ is a constant advection velocity.

We choose a finite difference method to discretize the PDE (5)

$$
L_h u_h = 0 \ , \tag{7}
$$

e.g. the first-order explicit upwind scheme with $v = u_h$ for (6)

$$
\frac{v_j^{n+1} - v_j^n}{\Delta t} + a \frac{v_j^n - v_{j-1}^n}{\Delta x} = 0 \ , \tag{8}
$$

where $j = 1, ..., N = \frac{1}{\Delta x}$, $n = 0, ..., \frac{t}{\Delta t} - 1$. Insert the exact solution u of the PDE (5) into the FDM (7) to obtain the truncation error

$$T_h = L_h u_h = L_h u_h - Lu \; . \tag{9}$$

Inserting $u(x_j, t_n)$ for v_j^n into (8), we obtain

$$\begin{aligned} T_h(x_j, t_n) &= \frac{u_{tt}(x_j, t_n)}{2}\Delta t + \mathcal{O}(\Delta t^2) - \\ &\quad -a\frac{u_{xx}(x_j, t_n)}{2}\Delta x + \mathcal{O}(\Delta x^2) \quad (10) \\ &= c\Delta x + \mathcal{O}(\Delta x^2) + d\Delta t + \mathcal{O}(\Delta t^2) \end{aligned}$$

The task is to estimate the leading error terms, i.e. in our example $c\Delta x$ and $d\Delta t$. Following Ferm and Lötstedt [3] and Lötstedt et al. [8], we can use Richardson extrapolation to approximate the truncation error by the discretization error:

1. Compute v_h^{n+1} by solving $L_h v_h^n = 0$ on fine grid with step size $h = \Delta x$ and time step $k = \Delta t$.

2. Restrict the fine grid solution v_h to the coarse grid with step size $2h$, i.e. use restriction operator (N assumed even) $I_h^{2h} v_h^n = [v_0^n, v_2^n, v_4^n, ..., v_N^n]^T$. Compute discretization error

$$\tau_{2h} = L_{2h} I_h^{2h} v_h^n \tag{11}$$

   on the coarse grid with time step k, i.e. in the example

$$\tau_{2h}(x_j, t_n) = \frac{v_j^{n+1} - v_j^n}{\Delta t} + a\frac{v_j^n - v_{j-2}^n}{2\Delta x} \; , \tag{12}$$

   $j = 2, 4, ..., N$.

3. Approximate the difference of the **truncation errors** on the fine and coarse grids by the difference of the **discretization errors** on the fine and coarse grids (same $k = \Delta t$). Then, the leading spatial truncation error is approximated by Richardson extrapolation:

$$ch^p \approx \tau_h = \frac{L_{2h} I_h^{2h} v_h^n - I_h^{2h} L_h v_h^n}{2^p - 1} \; . \tag{13}$$

   In our example, we have $p = 1$. Since $L_h v_h^n = 0$, the right hand side of (13) is the difference of the spatial discretizations on the coarse and fine grids, i.e. for our example

$$ch \approx \tau_h(x_j, t_n) = a\frac{v_j^n - v_{j-2}^n}{2\Delta x} - a\frac{v_j^n - v_{j-1}^n}{\Delta x} \; .$$

4. Check discretization error τ_h and refine or coarsen grid, if needed. Recalculate starting from 1. until the tolerances tol are met. Thus:
   While $\tau_h \geq tol_{refine}$, refine grid, i.e. use $\frac{h}{2}$.
   While $\tau_h \leq tol_{coarsen}$, coarsen grid, i.e. use $2h$.

5. After completing the grid adaptation in computing v_h^{n+1} with the required level of the spatial discretization error, control temporal truncation error as for ODEs, cf. section 3.

5 Anisotropic Grid Adaptation

In multi dimensions, the discretization errors in each coordinate direction are computed by

1. coarsening in x_l-direction, $l = 1, 2, 3$ in 3D,

2. computing the difference approximations of the x_l-derivatives, $l = 1, 2, 3$, on coarse grids using the fine grid solution,

3. comparing with fine grid discretizations of the respective derivatives.

If the error tolerance for refinement (coarsening) is not met, check whether refinement (coarsening) in one, two or all three x_l-directions is needed. Then, an anisotropic grid is obtained.

The procedure can be easily extended to finite difference methods on general, i.e. non-Cartesian, structured grids. The extension to structured finite volume methods is outlined

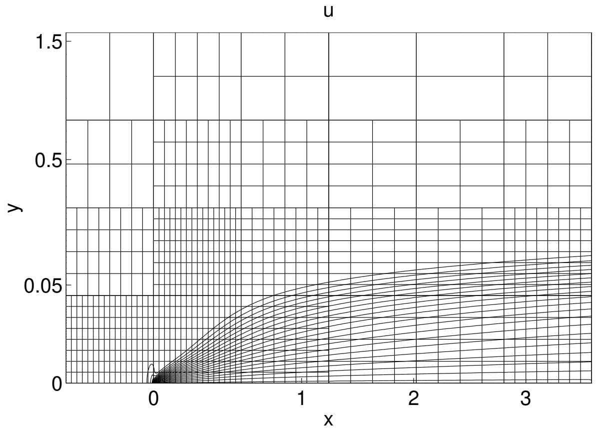

Figure 5: Anisotropic adapted grid with u velocity contours in the transformed domain, adaptation based on discretization error control for incompressible flow over a flat plate at $Re = 5000$ by Ferm and Lötstedt [3].

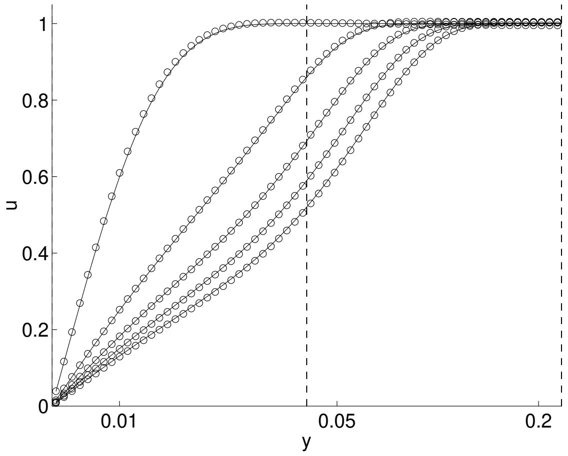

Figure 6: u velocity profiles (symbols $\circ$) in transformed y-coordinate compared with the Blasius solution (solid lines), adaptation based on discretization error control for incompressible flow over a flat plate at $Re = 5000$ by Ferm and Lötstedt [3].

in detail by Ferm and Lötstedt [3]. Defining coarse grid cells by the union of fine grid cells allows to generalize Ferm and Lötstedt's approach to unstructured finite volume methods. The data structure is considerably simplified, if the refinement and coarsening is not done locally, but blockwise. Therefore, Ferm and Lötstedt use anisotropic grid adaptation with a block-structured finite volume method for the 2D compressible and incompressible Navier-Stokes equations [3].

Examples by Ferm and Lötstedt [3] are presented here. Steady incompressible flow over a flat plate at Reynolds number $Re = 5000$ is computed using the artificial compressibility method. While the initial grid had 2304 cells, the adapted grid with 14496 cells is shown in Fig. 5 together with the u-velocity contours. The refinement tolerance was $tol_{refine} = 0.0017$. Note that the y-coordinate is stretched in Fig. 5. The u-velocities and the corresponding errors are compared at different stations in Fig.s 6 and 7, respectively. Compared to the initial grid, the error is considerably reduced with the adapted anisotropic grid.

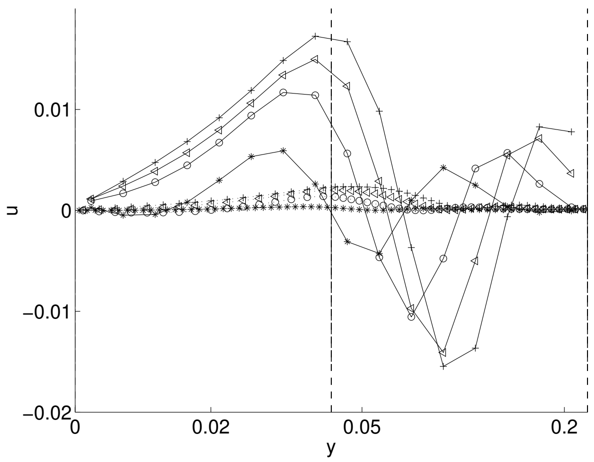

Figure 7: Error of u velocity contours in the transformed domain with initial grid (solid lines) and adapted grid (dotted lines), adaptation based on discretization error control for incompressible flow over a flat plate at $Re = 5000$ by Ferm and Lötstedt [3].

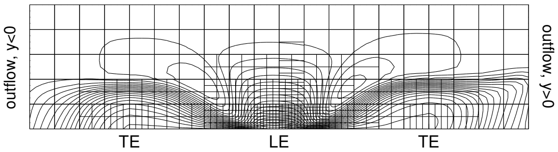

Figure 8: Anisotropic adapted grid with u velocity contours in the transformed domain, adaptation based on discretization error control for laminar subsonic flow over a NACA0012 airfoil by Ferm and Lötstedt [3] LE and TE denote leading and trailing edges, respectively.

Fig. 8 shows the adapted grid based on discretization error control for laminar subsonic flow over a NACA0012 airfoil at 1.25° angle of attack, Mach number $M_\infty = 0.5$ and Reynolds number $Re_\infty = 3000$ by Ferm and Lötstedt [3]. The contour plots in Fig.s 9 and 10 were obtained by that approach with the refinement tolerance $tol_{refine} = 0.058$, 1600 cells in the initial grid and 9600 cells in the adapted grid (Fig. 8). The analysis of the dependence of the solution error $u_h - u$ on the discretization error τ_h and the results by Ferm and Lötstedt [3] indicate that the discretization error is a useful error indicator.

6 Conclusions

A subtle analysis of certain flows, e.g. boundary layer flow, reveals the relation between the discretization errors and the errors of the flow variables [3]. The conclusion is that it often suffices to control the discretization errors. Other *a posteriori* error estimates are based on Taylor expansion [4], [5], [7].

$$u(\mathbf{x}) = u(0) + \mathbf{x}^T \nabla u(0) + \frac{1}{2}\mathbf{x}^T(\nabla\nabla^T u(0))\mathbf{x} + ... \tag{14}$$

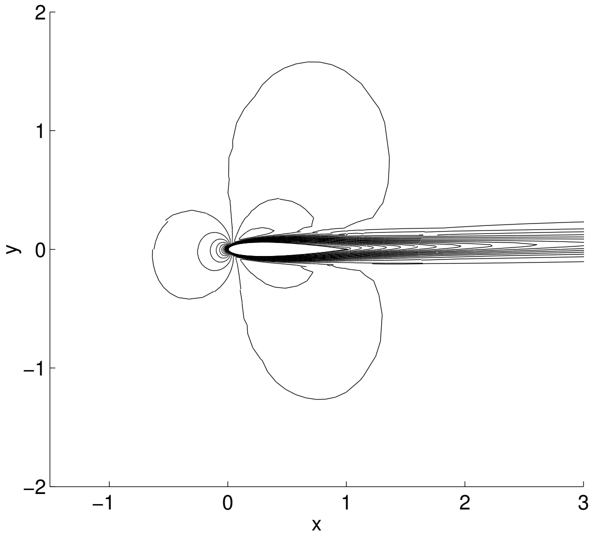

Figure 9: u velocity contours with anisotropic grid adaptation based on discretization error control for laminar subsonic flow over a NACA0012 airfoil by Ferm and Lötstedt [3].

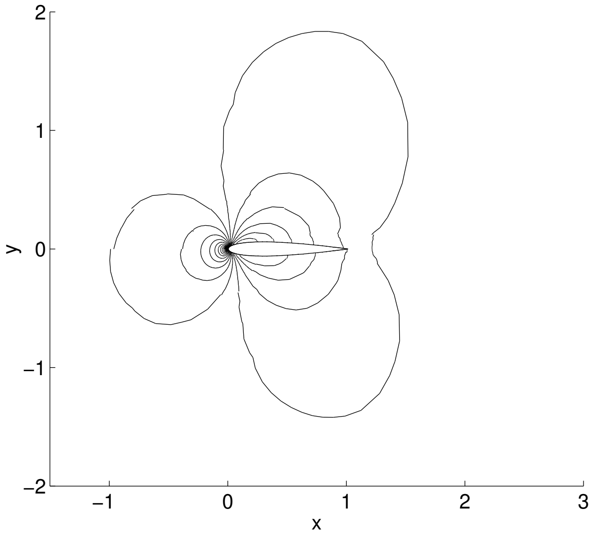

Figure 10: Pressure contours with anisotropic grid adaptation based on discretization error control for laminar subsonic flow over a NACA0012 airfoil by Ferm and Lötstedt [3].

The adjoint method optimizes the grid for solving a PDE by solving the adjoint problem. This approach is used with the Galerkin finite element method in [6], [2], [11], [1].

Other errors like oscillations at moving fluid interfaces in gas mixtures [9], flux discretization errors due to source terms [9], numerical instabilities due to the 'carbuncle' phenomenon [10], etc. are fundamental to the numerical methods used and require numerical analysis to understand the reasons for the errors and devise cures.

The most important recommendation is to be critical to the results produced by a CFD code and to use not only physical insight but also all the available knowledge on numerical analysis to get reliable results efficiently. Therefore, control errors in CFD!

ACKNOWLEDGEMENTS

The contributions by my colleagues Lars Ferm and Per Lötstedt are gratefully acknowledged. I thank Hideaki Aiso, NAL, Japan, for the invitation to the 20th ANSS and Yoko Takakura for the hospitality at Tokyo Noko University.

References

[1] Becker, R., Rannacher, R.: An Optimal Control Approach to a posteriori Error Estimation in Finite Element Methods. Preprint 2001-03, IWR Heidelberg, 2001.

[2] Eriksson, K., Estep, D., Hansbo, P., Johnson, C.: Introduction to Adaptive Methods for Differential Equations. Acta Numerica (1995), pp. 105-158.

[3] Ferm, L., Lötstedt, P.: Anisotropic Grid Adaptation for Navier-Stokes' Equations. Technical Report 2002-023, Department of Information Technology, Uppsala University, June 2002.

[4] Jasak, H., Gosman, A.D.: Automatic Resolution Control for the Finite Volume Method. Part 1: *A-posteriori* Error Estimates. Submitted to Numerical Heat Transfer, 2000.

[5] Jasak, H., Gosman, A.D.: Automatic Resolution Control for the Finite Volume Method. Part 2: Adaptive Mesh Refinement and Coarsening. Submitted to Numerical Heat Transfer, 2000.

[6] Johnson, C.: On Computability and Error Control in CFD. International J. for Numerical Methods in Fluids, Vol. 20 (1995), pp. 777-788.

[7] Lepage, C.Y., Suerich-Gulick, F., Habashi, W.G.: Anisotropic 3-D Mesh Adaptation on Unstructured Hybrid Meshes. AIAA Paper 2002-0859, 40th ASM&E, 14-17 Jan. 2002, Reno, Nevada.

[8] Lötstedt, P., Söderberg, S., Ramage, A., Hemmingsson-Fränden,L.: Implicit Solution of Hyperbolic Equations with Space-Time Adaptivity. BIT, Vol. 42 (2002), pp. 134-158.

[9] Müller, B., Jenny, P.: Improving the Reliability of Low Mach Number Flow Computations. CFD J., Vol. 9, No. 1 (2001), pp. 515-528.

[10] Pandolfi, M., D'Ambrosio, D.: Numerical Instabilities in Upwind Methods: Analysis and Cures for the "Carbuncle" Phenomenon. J. Comput. Physics, Vol. 166 (2001), pp. 217-301.

[11] Rannacher, R.: Error Control in Finite Element Computations. In Bulgak, H. and Zenger, C. (eds), "Proc. Summer School Error Control and Adaptivity in Scientific Computing", Kluwer Academic Publishers (1998), pp. 247-278.

[12] Sesterhenn, J., Müller, B., Thomann, H.: On the Cancellation Problem in Calculating Compressible Low Mach Number Flows. J. Comput. Physics, Vol. 151 (1999), pp. 597-615.