Accuracy Requirements for Steady and Transient Aerodynamics.

Ken Mattsson[*], Magnus Svärd[†], Mark Carpenter[‡]
and
Jan Nordström[§]

25th October 2002

Abstract

A numerical study on the benefit of high order methods is performed. Numerical computations of solutions governed by the Euler equations are performed. The propagation of a vortex convected through an empty domain, computations of steady state solutions around a NACA0012 airfoil and vortex-airfoil interaction are considered. These computations show that high order methods are often required in order to capture the significant flow features, especially for transient problems.

1 Introduction.

In computational fluid dynamics 2:nd order accurate codes are the most common choice for production aerodynamic calculations. These codes usually have rather large inherent dissipation and dispersion errors making it hard to accurately compute fine structures in the flow, see for example [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. There is always the possibility to increase the number of grid points to obtain a better solution but often the computational cost becomes

[*]Department of Information Technology, Uppsala University, Uppsala, Sweden.
[†]Department of Information Technology, Uppsala University, Uppsala, Sweden.
[‡]Computational Modeling and Simulation Branch, NASA Langley Research Center, Hampton, VA 23681
[§]Computational Aerodynamics Department, Aeronautics Division, The Swedish Defence Research Agency, SE-172 90 Stockholm, Sweden, and Department of Information Technology, Uppsala University, Uppsala, Sweden.

too large. High order methods does not have this drawback. However, the theory for stability and boundary conditions (see [11]) were, until recently, less developed making these codes less robust.

In this paper we will use high order central finite difference operators obeying an SBP property, developed in [12, 13]. These operators, together with an SAT boundary procedure first introduced in [14], makes it possible to prove stability for linear problems by using the energy method (see for example [15, 16, 17, 18, 3]). The SAT boundary procedure can be used both for external and internal boundaries, making the scheme stable for grids composed of blocks.

For nonlinear problems, stability by energy estimates can not be obtained in general. One usually distinguish between two different phenomena, shocks and aliasing (see [19]), which make central difference schemes unstable when applied to nonlinear problems. However, instabilities can appear already for linear problems with variable coefficients, where it can be shown that numerical dissipation can stabilize the method without ruining the accuracy. Strang [20] showed that if the solution of the nonlinear differential equation is sufficiently smooth, convergence follows if the linearized approximation is stable. A thorough survey of nonlinear stability can be found in [11]. This motivates the use of numerical dissipation in practical flow simulations. To prevent aliasing errors, which are present also in smooth regions of the solution, one usually add artificial dissipation constructed by sufficiently high order undivided differences (see [21] and [19]).

To preserve the accuracy at the boundaries and the SBP property of the difference method, the artificial dissipation must be supplied with a specific boundary treatment. Such SBP conserving dissipation operators were developed in [4]. These are of the same order of accuracy as the central finite difference scheme itself, but can also be used to construct upwind schemes with a proper scaling. In the upwind case, one order of accuracy less is obtained.

The methodology mentioned above was used to construct 3:rd and 5:th order accurate upwind schemes, which were used to compute solutions to the Euler equations. The cases studied are: vortex convected through an empty domain; steady state computations for the flow around a NACA0012 airfoil and a vortex-airfoil interaction.

The contents of this report are divided as follows: In section 2 equations and analytic solutions are given; section 3 displays the discretization techniques and also an explanation of the parallel implementation. The numerical experiments are discussed in section 4; section 5 contains a discussion on the parallel performance. In section 6 conclusions are drawn.

2 The continuous problem.

The two-dimensional Euler equations in curvilinear coordinates, $x = x(\xi, \eta)$ and $y = y(\xi, \eta)$, can be written,

$$u_t + F_\xi + G_\eta = 0 \ .$$

The contravariant fluxes are given by

$$u = \begin{bmatrix} J\rho \\ J\rho u \\ J\rho v \\ J\rho E \end{bmatrix}, F = \begin{bmatrix} J\rho(\xi_x u + \xi_y v) \\ J\rho u(\xi_x u + \xi_y v) + J\xi_x p \\ J\rho v(\xi_x u + \xi_y v) + J\xi_y p \\ J(\rho E + p)(\xi_x u + \xi_y v) \end{bmatrix}, G = \begin{bmatrix} J\rho(\eta_x u + \eta_y v) \\ J\rho u(\eta_x u + \eta_y v) + J\eta_x p \\ J\rho v(\eta_x u + \eta_y v) + J\eta_y p \\ J(\rho E + p)(\eta_x u + \eta_y v) \end{bmatrix}$$

where $J = (\xi_x \eta_y - \xi_y \eta_x)^{-1}$, ρ is the density, u, v are the velocity components in the x, y directions respectively, $E = e + 1/2(u^2 + v^2)$ the total energy, where e is the internal energy and p the pressure.

In some of the applications below we consider the propagation of a vortex. The vortex model is presented in [22]. It satisfies the two-dimensional Euler equations, under the assumption of isentropy. In [23] it is shown that the solution is steady in the frame of reference moving with the freestream. The scaled vortex has the velocity field

$$v_\Theta = \frac{\epsilon r}{2\pi} exp\left(\frac{1 - r^2}{2}\right) \ , \tag{1}$$

where ϵ is the nondimensional circulation, (r, Θ) are the polar coordinates. The analytic solution in a fixed frame of reference (x, y, t) becomes

$$\begin{aligned} u &= 1 - \tfrac{\epsilon y}{2\pi} exp\left(\tfrac{f(x,y,t)}{2}\right) \ , \qquad & v &= \tfrac{\epsilon(x-(t-T))}{2\pi} exp\left(\tfrac{f(x,y,t)}{2}\right) \\ \varrho &= \left(1 - \tfrac{\epsilon^2(\gamma-1)M^2}{8\pi^2} exp\left(f(x,y,t)\right)\right)^{\frac{1}{\gamma-1}} \ , & p &= \tfrac{\varrho^\gamma}{\gamma M^2} \end{aligned} \tag{2}$$

where $f(x, y, t) = 1 - (((x - x_0) - t)^2 + y^2)$, M is the Mach number, $\gamma = c_p/c_v$ and x_0 is the initial position of the vortex (in the x-direction). The vortex is introduced into the computational domain by using the analytic solution as boundary data and initial data.

3 Computational procedure.

As mentioned above, we use numerical schemes of a specific form for stability reasons. Let h denote the spacing between two grid points. Further, let the grid consist of $N+1$ points numbered from 0 to N. Let u denote a grid function and u_x denote its exact derivative projected onto the grid. $H^{-1}Qu$ is an m:th order accurate approximation of u_x, if the internal discretization error is of order $\mathcal{O}(h^m)$ with a boundary closure of order $\mathcal{O}(h^{m-1})$. This is due to the fact that one can lower the accuracy by one order at a finite number of points and still produce m:th order of accuracy, see [24]. The matrix H is symmetric and positive definite and defines a norm $\|u\|_H^2 = u^T H u$. $H^{-1}Q$ is called an SBP operator if $Q + Q^T = B$, where $B = diag[-1, 0, .., 0, 1]$.

The simultaneous approximation term method (SAT), required for a stable implementation of boundary and interface conditions (see [17]), adds one term for each condition. Each of these terms contain a free parameter that is determined by stability requirements. This technique allows us to split the domain into different blocks and connecting them using SAT and hence allows the building of a stable, efficient and highly accurate scheme.

3.1 Flux splitting

We will briefly describe the upwinding or flux splitting procedure in this paper. Consider the following one-dimensional system with m unknowns,

$$u_t + (Au)_x = 0 \ , \quad a \le x \le b \ , \tag{3}$$

where u is the solution vector. We start by splitting the flux vector (Au) into two parts with positive and negative running characteristics respectively. We have

$$A\,u = \frac{1}{2}\,(A + \alpha\, I_m)\,u + \frac{1}{2}\,(A - \alpha\, I_m)\,u = A_+ u + A_- u \ ,$$

where α is the magnitude of the largest eigenvalue to A and I_m is the $m \times m$ identity matrix. The system (3) can now be written as

$$u_t + [A_+ u]_x + [A_- u]_x = 0 \ . \tag{4}$$

We now turn to the semidiscrete approximation of (4). The domain ($a \le x \le b$) is discretized using N equidistant grid points. The numerical approximation at grid point x_j is denoted v_j. We define a discrete solution vector $v^T = [v_1, v_2, \cdots, v_N]$. The derivative u_x is approximated with a nondissipative and p:th order accurate finite difference approximation $H^{-1}Q\,v$, see the description above. We also need the notation

$$C \otimes D = \begin{bmatrix} c_{0,0}\,D & \cdots & c_{0,q-1}\,D \\ \vdots & & \vdots \\ c_{p-1,0}\,D & \cdots & c_{p-1,q-1}\,D \end{bmatrix}$$

where C is a $p \times q$ matrix and D a $m \times n$ matrix. The $p \times q$ block matrix $C \otimes D$ is called a *Kronecker product.* There are some useful rules for Kronecker products. In this paper we will use $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$.

For nonlinear convection problems it is well known that centered difference schemes require the addition of artificial dissipation to absorb the energy of the unresolved modes. Hence, we add an artificial dissipation term $H^{-1}Rv$, with the stability and accuracy properties that $R = R^T \geq 0$ and that $H^{-1}Rv$ is small enough. The artificial dissipation operator for a p:th order accurate method is based on a second order accurate difference approximation of $d^p u/dx^p$, with a specific boundary treatment, see [4].

The SBP operator and the dissipation operator can now be combined to an upwind and a downwind difference operator. We introduce an upwind operator $D_+ = H^{-1}(Q + R)$, and a downwind operator $D_- = H^{-1}(Q - R)$.

A corresponding semidiscrete approximation to (4) can be written

$$\begin{aligned} v_t &= -(D_+ \otimes A_+)v - (D_- \otimes A_-)v \\ &= -(H^{-1} Q \otimes A)v - \alpha (H^{-1} R \otimes I_m)v \; . \end{aligned} \tag{5}$$

Obviously, the upwind formulation is a conservative formulation, with the addition of an artificial dissipation term scaled by α. This procedure can be generalized to multidimensional problems.

3.2 Stable interface treatment.

As an example of the block interface procedure we will consider,

$$\begin{aligned} u_t + Au_x &= 0, \qquad -1 \leq x \leq 0, \\ v_t + Av_x &= 0, \qquad 0 \leq x \leq 1, \end{aligned}$$

with the interface condition $u = v$ at $x = 0$. A is a constant symmetric matrix with m rows and columns. Discretize the domains with n and l points and, for simplicity, denote the $n \times 1$, $l \times 1$ vectors u and v respectively. The difference operators might also be different in the left and right domain and are denoted D_-^L, D_+^L, D_-^R and D_+^R. The semidiscrete system becomes,

$$u_t + (D_+^L \otimes A_+)u + (D_-^L \otimes A_-)u = (H_L^{-1} \otimes \Sigma_L)e_N \otimes (u_N - v_0), \tag{6}$$

$$v_t + (D_+^R \otimes A_+)u + (D_-^R \otimes A_-)u = (H_R^{-1} \otimes \Sigma_R)e_0 \otimes (v_0 - u_N), \tag{7}$$

where the right hand sides are the SAT penalty terms. Σ_R and Σ_L are unknown $m \times m$ matrices to be determined below. $e_0 = (1, 0, ..., 0)^T$, $e_N = (0, ..., 0, 1)$ are $l \times 1$, $n \times 1$, vectors respectively while u_N and v_0 are $m \times 1$

vectors with components located at the boundary points. By multiplying (6) and (7) by $u^T(H_L \otimes I_m)$ and $v^T(H_R \otimes I_m)$ respectively, adding the transpose and making use of $Q + Q^T = B$ we obtain,

$$\frac{d}{dt}(\|u\|^2_{H_L} + \|v\|^2_{H_L}) = (u_N, v_0)\, M\, (u_N, v_0)^T - 2\alpha\, (u, v)\, R\, (u, v)^T \,. \tag{8}$$

In (8), $R = diag(R_L \otimes I_m, R_R \otimes I_m)$ is symmetric and positive semidefinite and

$$M = \begin{bmatrix} -A + 2\Sigma_L & -\Sigma_L - \Sigma_R \\ -\Sigma_L - \Sigma_R & A + 2\Sigma_R \end{bmatrix} .$$

In (8) we have neglected the terms from the far field boundaries, at $x = -1, 1$. To obtain stability M, has to be negative semidefinite which is achieved by choosing Σ_L and Σ_R properly. Consider,

$$C^T M C = \begin{bmatrix} -A + 2\Sigma_L - (\Sigma_L + \Sigma_R)(A + 2\Sigma_R)^{-1}(\Sigma_L + \Sigma_R) & 0 \\ 0 & A + 2\Sigma_R \end{bmatrix} \tag{9}$$

where

$$C = \begin{bmatrix} I_m & (\Sigma_L + \Sigma_R)(A + 2\Sigma_R)^{-1} \\ 0 & I_m \end{bmatrix} ,$$

assuming that $(A + 2\Sigma_R)$ is invertible. The stability condition then reduces to make $C^T M C$ negative semidefinite, i.e. the blocks on the diagonal of (9) each have to be negative semidefinite. In fact, $A + 2\Sigma_R$ has to be strictly negative since it is invertible by assumption. Since A is a symmetric matrix we can diagonalize it by $X^T A X = \Lambda$, where X is a matrix consisting of the eigenvectors of A. Further, consider penalty parameters Σ_L and Σ_R, of the form $X^T \Sigma_L X = \Lambda_L$ and $X^T \Sigma_R X = \Lambda_R$. The upper left block of (9) yields the stability condition,

$$X^T (C^T M C)_{11} X = -\Lambda + 2\Lambda_L - (\Lambda_L + \Lambda_R)(\Lambda + 2\Lambda_R)^{-1}(\Lambda_L + \Lambda_R) \leq 0 \,, \tag{10}$$

and the lower right block,

$$X^T (C^T M C)_{22} X = \Lambda + 2\Lambda_R < 0 \,. \tag{11}$$

Denote by λ^i the $i : th$ diagonal component of Λ and similarly λ^i_L and λ^i_R for Λ_L and Λ_R. The i:th rows of (10) and (11) constitute a system of two equations with unknowns λ^i_R and λ^i_L. Thus for $i = 1...m$ we obtain from (10) and (11),

$$\lambda^i_L < \frac{\lambda^i}{2} \quad , \lambda^i_R = \lambda^i_L - \lambda^i \,. \tag{12}$$

Equation (12) correspond to the stability and conservation conditions found in for example [15]. With Λ_L and Λ_R determined, inserting $\Sigma_L = X \Lambda_L X^T$ and $\Sigma_R = X \Lambda_R X^T$ in (6) and (7) yields a stable scheme.

3.3 Parallelization.

It was shown above how to split the domain of the PDE into two computational domains connected with the SAT method. In one dimension one value must be shared between the connecting domains. In two dimensions, the computational blocks have to share the points constituting the internal boundaries at each block interface. This makes the problem particularly easy to parallelize. If each block corresponds to one process on a parallel computer, this process only has to communicate with other processes at the block interfaces, sending and receiving the boundary data at each computational step in time and then computing the solution within the block. With fairly large blocks, the amount of communication will be a small part of the total amount of work. If the grid is designed such that each block has approximately the same size, a proper load balance will be achieved.

4 Numerical computations.

4.1 Vortex in free space

In this section we compare the numerical results of a vortex calculation in free space, using 3:rd and 5:th order upwind methods. For the sake of comparison, computations are also done using a 2:nd order central scheme with a 4:th order difference as artificial dissipation, scaled as in the 3:rd order upwind case. The scaling preserve the accuracy of the scheme and match the $1/h$-dependence of the SBP operator. In the first test the numerical computations are done on a rectangle divided into two blocks. The vortex is imposed as initial data at the center of the internal boundary. In Figure 1 the solution at $t = 1$, using 100×100 grid points, for the 5:th order case is shown.

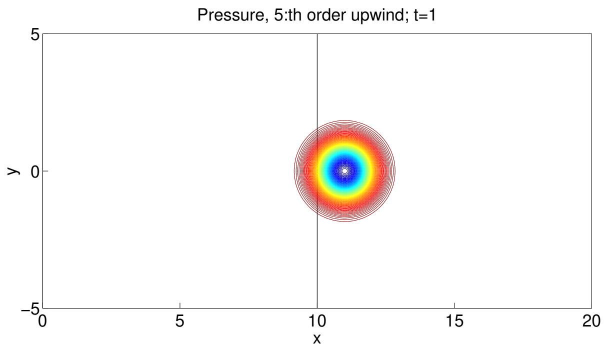


Figure 1: Pressure contour, 5:th order upwind at t=1, 100×100 grid points.

We are interested to see if the boundary treatment of the dissipation operator preserve the order of accuracy. We induce a vortex with strength $\epsilon = 1$, see (1). The convergence rate is calculated as,

$$q = \log_{10}\left(\frac{\|u - v^{h1}\|_h}{\|u - v^{h2}\|_h}\right) / \log_{10}\left(\frac{h_1}{h_2}\right) , \tag{13}$$

where u is the analytic solution and v^{h1} the corresponding numerical solution with grid size h_1. $\|u - v^{h1}\|_h$ is the $l_2 - error$. The results are presented in Table 1.

N	$2:nd$	q	$3:rd$	q	$5:th$	q
50	8.32e-6		1.52e-6		6.75e-7	
100	2.58e-6	1.68	2.24e-7	2.77	1.06e-8	6.00
150	1.24e-6	1.80	7.30e-8	2.76	1.12e-9	5.53
200	7.20e-7	1.90	3.24e-8	2.83	2.49e-10	5.24

Table 1: l_2-error and convergence rate q, tested for a vortex in free space.

For a pure convection problem, the major source of error is the dispersion error. The vortex field consists of several Fourier modes which all advect at different speeds. The dispersion error (the difference between the correct and the computed location of a Fourier mode) varies with both grid size ($h = 1/N$), frequency and time. With the artificial dissipation added, we also have dissipation error which will decrease the vortex strength. The dissipation error also varies with the grid size, frequency and time.

A calculation with a 6:th order accurate nondissipative method using N grid points in each of the spatial directions, result in a dispersion error proportional to $(1/N)^6$. In [25] it is shown that a 2:nd order accurate method require approximately N^3 grid points in each of the spatial directions to reproduce the same dispersion error.

A (2p-1):th order accurate upwind method for a smooth problem have approximately the same dispersion error as a nondissipative 2p:th order accurate method. An upwind method also have a dissipation error proportional to h^{2p-1}. Hence, as we grid-refine the dissipation error will eventually dominate the dispersion error for an upwind method. As a numerical test of the dispersion and dissipation errors, we do a long time integration. The vortex is introduced at the left boundary (x=0), by using the analytic solution (see (2)) as boundary and initial data. We induce a rather strong vortex, $\epsilon = 5$. The solution is advanced to t=15, implying that the vortex have reached x=15. In Figure 2 a comparison between the 2:nd and the 5:th order methods at $t = 15$ is shown. The computations are done at different grid resolutions.

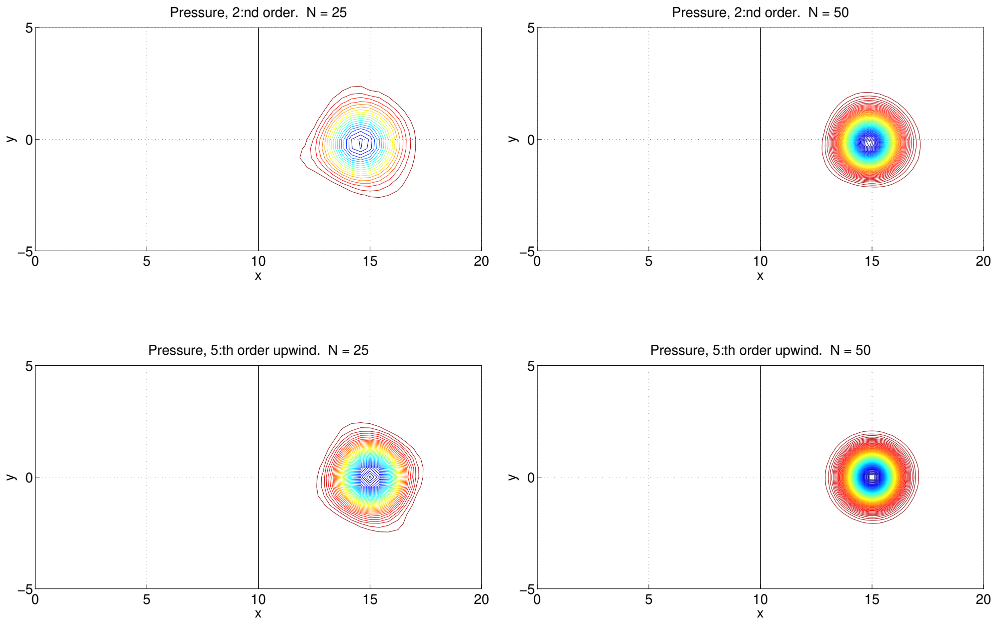


Figure 2: Pressure contour at t=15, using 25 × 25, 50 × 50 grid points respectively. Comparing 2:nd and 5:th order methods.

We introduce a measure to quantify the strength of a vortex, $\Omega = \frac{|p_{inf} - p_{min}|}{p_{inf}}$, where p_{inf} is the pressure far from the vortex and p_{min} is the pressure minimum at the center of the vortex. The exact pressure minimum can be found from the analytic solution of the pressure. The error of the vortex strength $\Delta\Omega$, is a measure of the dissipation error. We also introduce a measure to quantify the dispersion error for the vortex, $\Delta\Phi = \|(x,y)^a_{min} - (x,y)^c_{min}\|_2$, where $(x,y)^a_{min}$ is the analytic location of p_{min} and $(x,y)^c_{min}$ is the corresponding computed value found by interpolation between grid points. The results are plotted in Figures 3 for different methods and different grid resolutions.

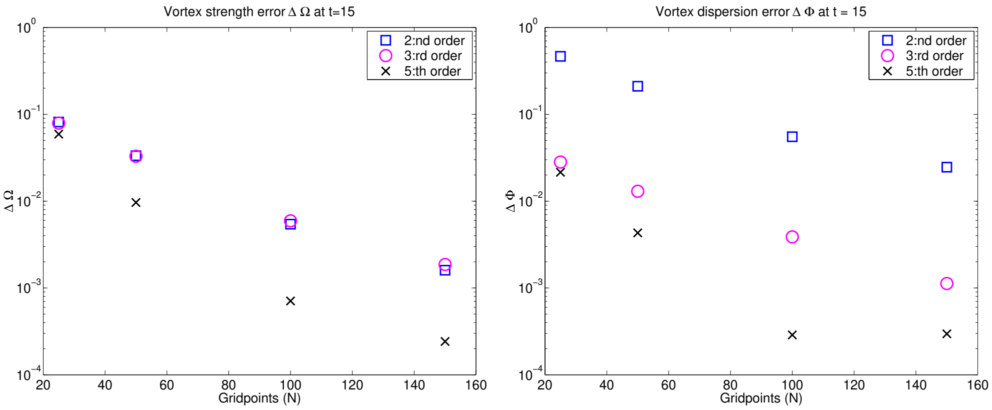


Figure 3: The dissipation ($\Delta\Omega$) and dispersion ($\Delta\Phi$) errors.

The vortex strength for the 3:rd order upwind and the second order method are almost identical, since they have the same amount of artificial dissipation added. The 2:nd order method have rather large dispersion error. The 5:th order upwind method is clearly superior.

As a last test, an even longer time integration on a rectangle is performed. The calculations are done using 500×100 points. The spatial resolution correspond to the computations using 50 grid points, see Figure 3. The dispersion and the dissipation errors are plotted as a functions of time in Figure 4.

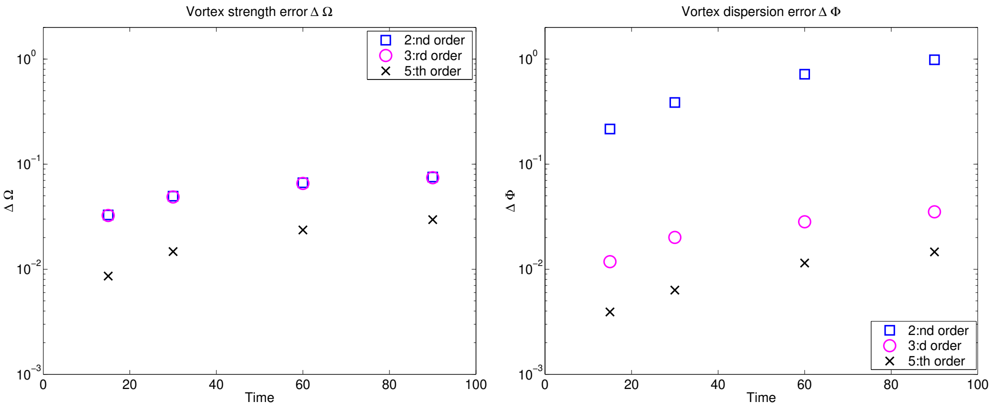


Figure 4: The dissipation ($\Delta\Omega$) and dispersion ($\Delta\Phi$) errors at different times.

The vortex strengths for the 3:rd order upwind and the 2:nd order method, as expected, are almost identical. However, the dispersion error in the 2:nd order case is rather large as compared to the 3:rd and 5:th order upwind case. This was also indicated in the previous results, see Figure 3. Again the 5:th order upwind method is superior compared to the less accurate methods regarding both dispersion and dissipation errors.

4.2 Steady state calculations

In the next test we consider steady state calculations around a NACA0012 airfoil. The nondimensional lift and drag are defined as follows:

$$c_l = \frac{L}{q_\infty c},\ c_d = \frac{D}{q_\infty c}, \tag{14}$$

where L, D are the lift and drag forces respectively. q_∞ is the dynamic pressure of the free stream, i.e. $q_\infty = \frac{1}{2}\rho_\infty U_\infty^2$, and c is the cord length.

The computational domain is split into 12 blocks, see Figure 5. Numerical calculations are done on a sequence of successively finer meshes (in this paper denoted: coarse, fine and finest). The number of grid points on successively finer meshes are doubled in each direction.

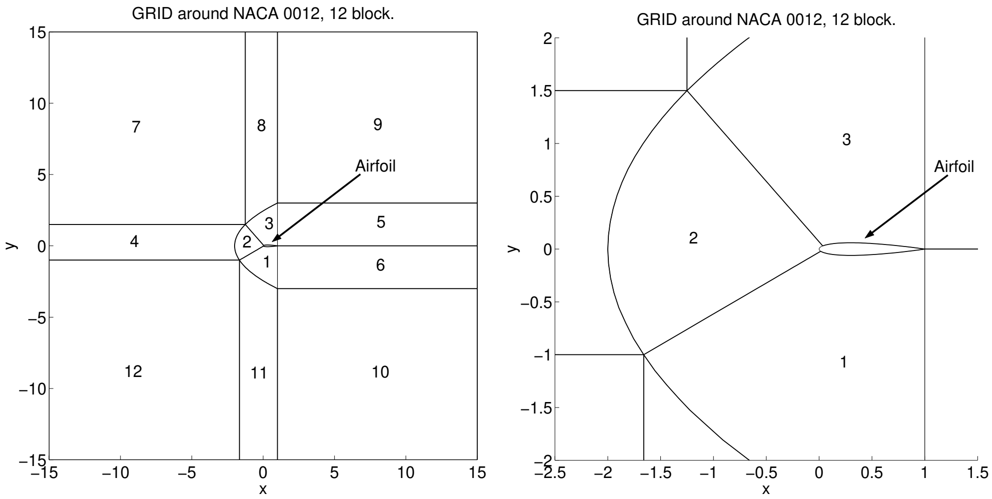


Figure 5: The computational domain, divided into 12 blocks, around a NACA0012 airfoil. The right subfigure is a close up.

The first case was run at a Mach number of 0.63 and an angle of attack of 2 degrees. At these conditions, the flow is completely subsonic. In Figure 6, the surface pressure distributions on the finest mesh are shown for the 3:rd order case, including a comparison with the result obtained using the node centered finite volume solver (EDGE [26]). EDGE is developed and extensively tested at the division of Aeronautics (FFA) at the Swedish Defense Research Agency. The results agree well.

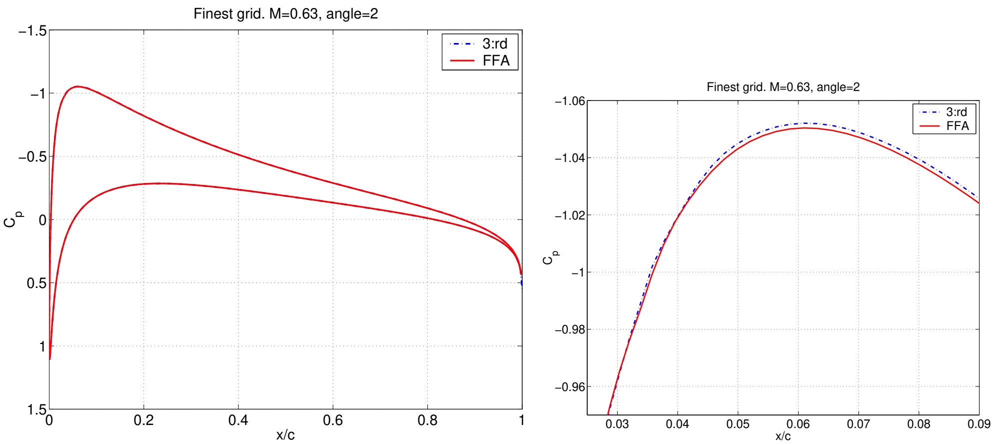


Figure 6: Steady state computations around a NACA0012 airfoil, 2 degree angle of attack, Mach 0.63. Surface pressure distributions on the finest grid.

The computed values for C_l and C_d for the different methods on the fine and the coarse grid is presented in Table 2. As a reference solution we have included the result of FFA finite volume solver on the finest grid (the 3:rd order method give almost identical results on this grid).

Method	$2:nd$	$3:rd$	$5:th$	FFA
coarse (lift)	0.3230	0.3258	0.3264	
fine (lift)	0.3227	0.3238	0.3238	0.3213
coarse (drag)	0.0020	0.0017	0.0022	
fine (drag)	0.0008	0.0007	0.0011	0.0005

Table 2: Steady state computation around a NACA0012 airfoil, 2 degree angle of attack, Mach 0.63. Lift and drag for different methods and grids.

The mesh refinement shows that finer meshes clearly improve the accuracy. That the 5:th order scheme has a higher drag than the 2:nd and 3:d order methods on the fine grid is probably due to a larger error constant for this particular problem.

The second case was run at a Mach number of 0.8 and 1.25 degree angle of attack, where the flow is transonic and shocks exist on both the upper and lower surfaces of the airfoil. In Figure 7, the surface pressure distributions on the finest mesh are shown for the 3:rd order case, including a comparison with the result obtained using the finite volume solver. Again, the results agree well. Since no shock treatment is included for the 3:rd order method, slight oscillations are seen close to the shock.

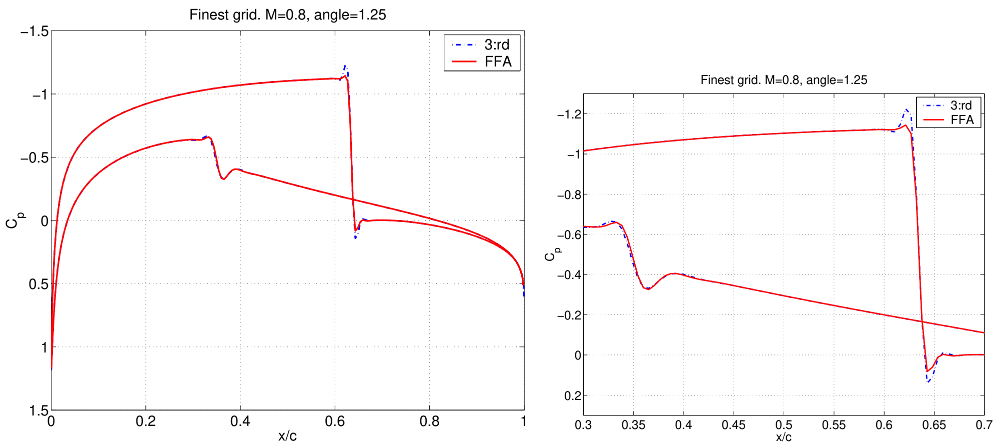


Figure 7: Steady state computations around a NACA0012 airfoil, 1.25 degree angle of attack, Mach 0.8. Surface pressure distributions on the finest grid.

In Figure 8, the results using the fine and the coarse mesh are shown and as a reference solution we have included the result from the finite volume method on the finest mesh. On the coarse mesh the weak shock on the lower surface is not resolved for any method.

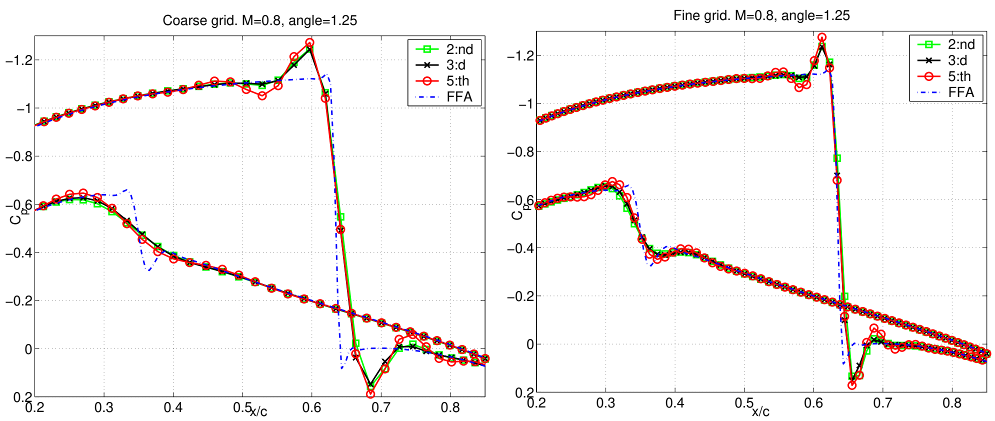


Figure 8: Steady state computations around a NACA0012 airfoil, 1.25 degree angle of attack, Mach 0.8. Surface pressure distributions on different grids.

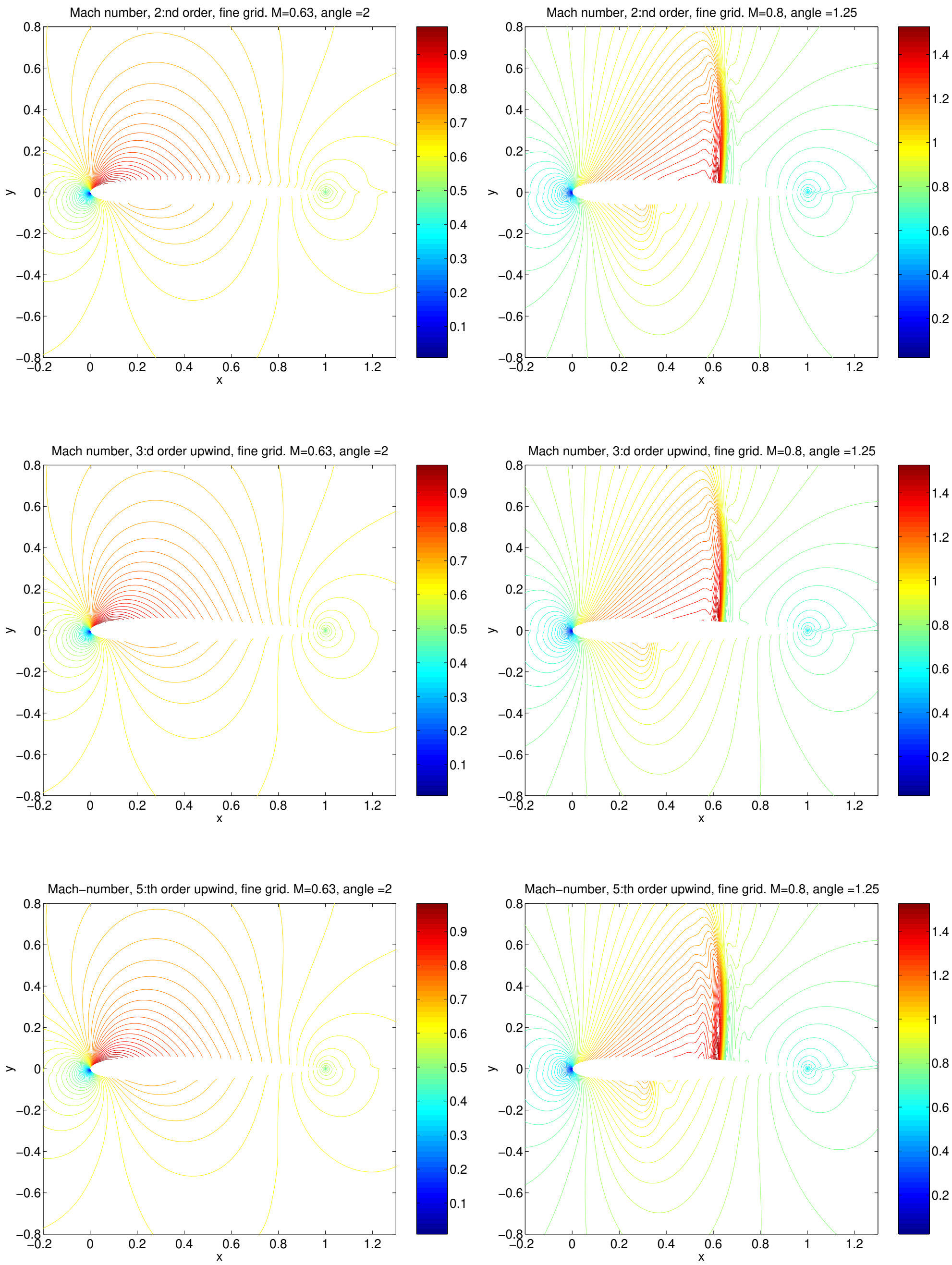


Figure 9: Steady state, NACA0012. Left column: Mach 0.63 at 2 degree angle. Right column: Mach 0.8 at 1.25 degree angle. Comparing 2:nd, 3:rd and 5:th order upwind on the fine grid.

In Figure 9, the Mach number contours for both cases are shown. The computed values for C_l and C_d are presented in Table 3. As a reference solution we have again included the result using the FFA finite volume solver on the finest grid.

Method	$2:nd$	$3:rd$	$5:th$	FFA
coarse (lift)	0.3509	0.3465	0.3457	
fine (lift)	0.3444	0.3408	0.3403	0.3352
coarse (drag)	0.0238	0.0234	0.0239	
fine (drag)	0.0224	0.0222	0.0225	0.0217

Table 3: Steady state computation around a NACA0012 airfoil, 1.25 degree angle of attack, Mach 0.8. Lift and drag for different methods and grids.

In section 4.1 we concluded that low order methods are not good enough for wave propagating problems due to the large dispersion errors, a well known fact. The results of this section indicate that low order methods are sufficiently accurate for steady state problems.

4.3 Vortex-wing interaction

In this section we consider the interaction of a vortex hitting an airfoil. As initial data we use steady state solutions of the different methods and introduce the vortex as boundary data on the left boundary, see Figure 5. The vortex is swept downstream towards the airfoil and eventually hits it, see Figures 10-11. In order to elucidate the quality of the results we also include the result using the 3:rd order upwind on the finest grid, see Figure 12. In this flow case, the difficulties encountered in section 4.1 and 4.2 are combined. The method must be able to handle both the wave (vortex) propagation problems (of section 4.1) and the non-linear problems (in section 4.2).

The computed values for C_l and C_d for the unsteady vortex-wing interaction for different times on the coarse and the fine grid are shown in Figure 13. The 2:nd order method gives reasonable result on the fine mesh, comparing the computed values for lift and drag, but on the coarse mesh the result differs a lot from the computed values using the higher order upwind methods.

If we compare the solutions, see Figures 10-12, the 2:nd order method, although it gives reasonable result for the integrated properties such as lift and drag, gives a completely different solution compared to the 3:rd and 5:th order upwind methods on the fine grid. This shows that a reasonable agreement in the integrated properties does not necessarily imply good agreement of the fine structures in the flowfield.

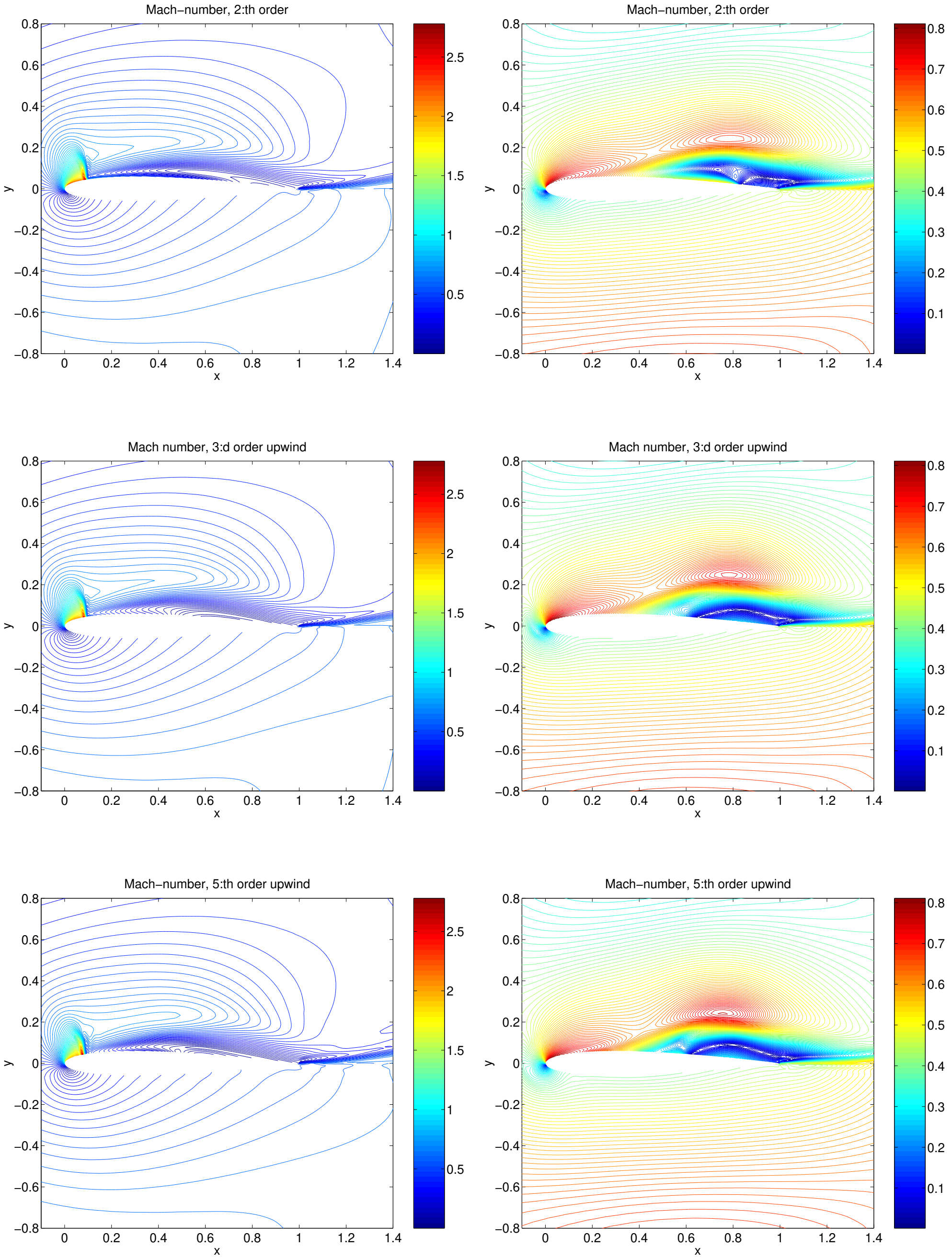


Figure 10: Vortex hitting a NACA0012. Left column shows the vortex at x=0 and right column shows the vortex at x=0.5. Comparing 3:rd and 5:th order upwind to a second order method.

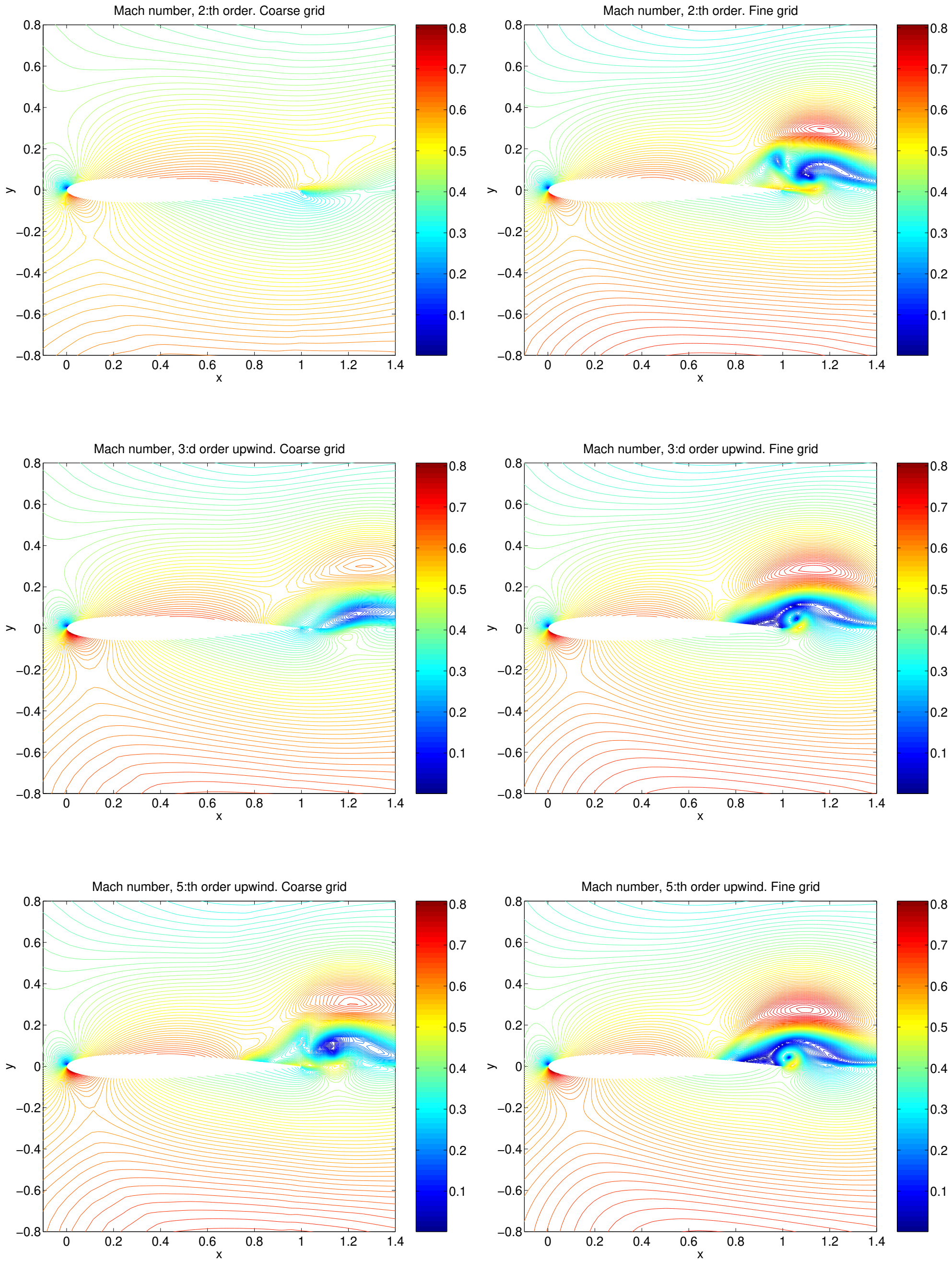
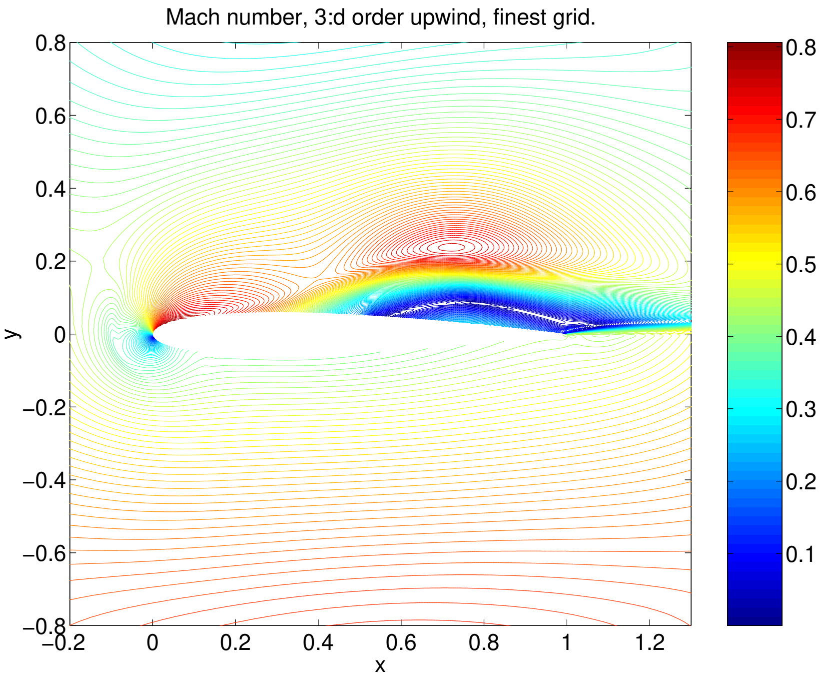


Figure 11: Vortex hitting a NACA0012 at x=1. Right column using a fine mesh and left column using a coarse mesh. Comparing 3:rd and 5:th order upwind to a second order method. Notice the similarity in the flowfields for the 3:rd and 5:th order methods, on the fine grid.



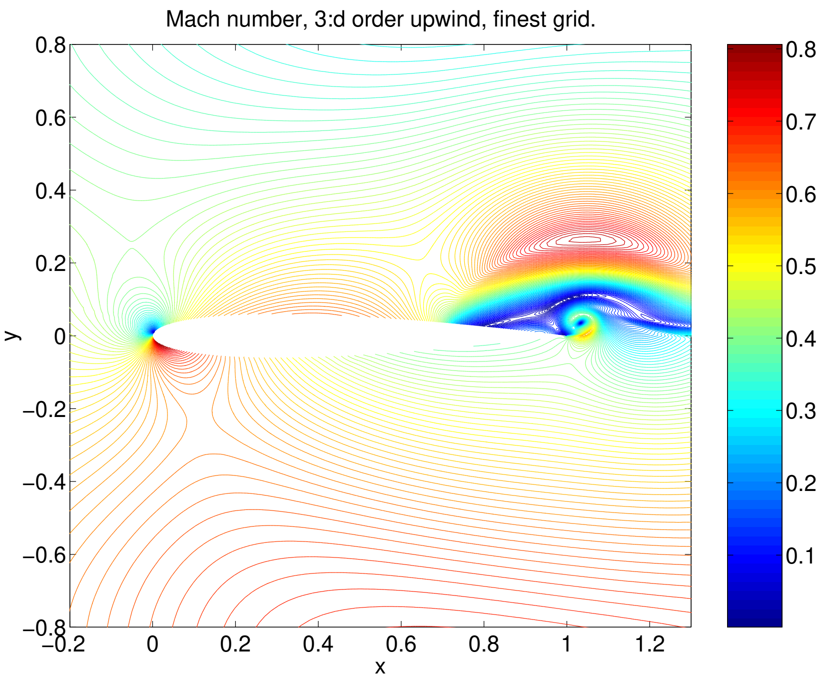


Figure 12: Vortex hitting a NACA0012. Left column shows the vortex at x=0.5 and right column shows the vortex at x=1. Computed with the 3:rd order upwind method on the finest grid.

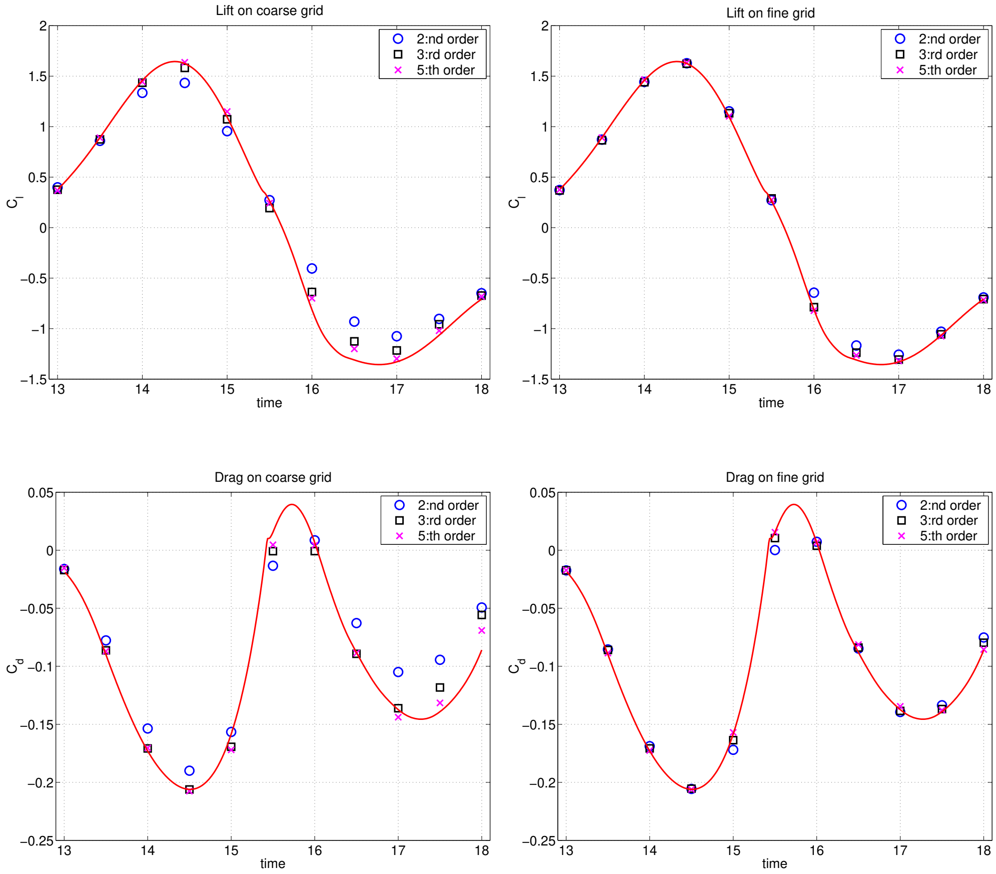


Figure 13: C_l and C_d, for the vortex-wing interaction. The left column using the coarse mesh and the right column the fine grid. $t = 15$ corresponds to the time when the vortex hits the airfoil, at $x = 0$. The solid lines are the 3:rd order results on the finest mesh.

5 Parallel performance

The calculations were done on a SunFire 15k system, with 48 UltraSPARC III+ CPUs at 900 MHz with 8 MB of L2 Cache memory and 48 GByte of primary RAM. Theoretical peak performance is 86.4 Gflop/s.

Let w denote the problem size and p the number of processors used to solve the problem. The run time to solve a problem of size w using p identical processors is denoted T_p^w. We define two different measures of parallel performance, *speedup* and *scaled speedup*. Speedup is defined as $S = T_1^w / T_p^w$ and is a measure of the efficiency of the parallel method, i.e. how much time is spent in processor communication compared to computations. Scaled speedup is defined as $S_s = pT_1^w/T_p^{p\,w}$ and is a measure of the scalability of the system, i.e. its capacity to increase speedup in proportion to the number of processors.

Theoretically, speedup can never exceed the number of processors, p. In practice, a speedup greater than p is sometimes observed for some problem sizes, when the data for one processor no longer fits the cache memory (a phenomenon called *super linear speedup*).

The numerical computations are done on a square. The square is divided into smaller blocks, one for each processor. The calculations are done using 1,4,8 and 16 processors. The result is presented in Figure 14. When the blocks are of size 240^2 or larger the data no longer fits into the cache memory, and we measure a speedup that exceeds the theoretical limit.

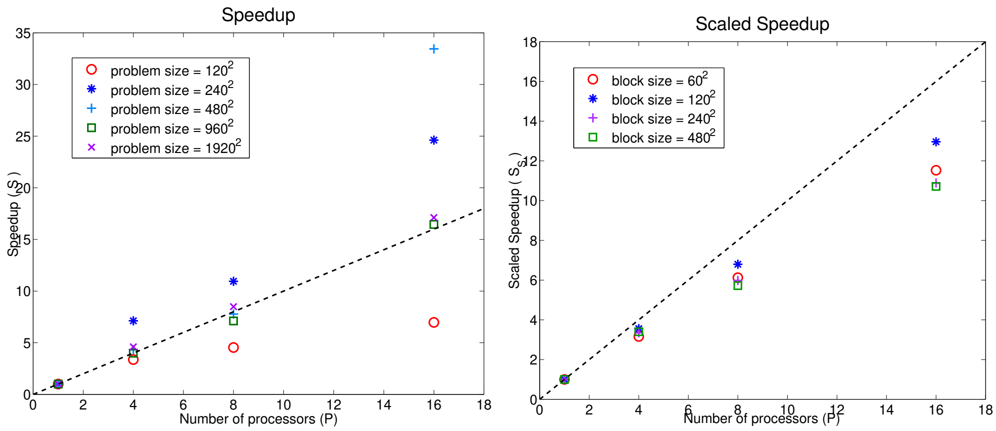


Figure 14: Speedup and Scaled speedup.

We also measure the speedup for an Euler calculation around a NACA0012 airfoil. The computational domain is split into 12 blocks, see Figure 5. Numerical calculations are done on a fine grid and a coarse grid. The speedup measured on the coarse grid is 9.7 and on the fine grid 9.5. The theoretical limit is 12 and the discrepancy is due to a nonoptimal load balance.

6 Conclusions.

The use of SBP operators and the SAT boundary procedure leads to stable and very accurate schemes. The grid can be split into blocks to fit complex geometries without loosing stability. Also, these kind of schemes lend themselves very naturally to parallelization allowing efficient large scale computations.

With the building blocks mentioned above stable 3:rd and 5:th order accurate upwind schemes have been constructed, in order to compute solutions to the 2-D Euler equations. This machinery can of course be generalized to handle more general 3-D problems.

Several computations have been done to show the efficiency of high order methods. We have shown that for time dependent problems and fine structures, high order methods are necessary to accurately compute the solution, on a reasonable fine grid. However, if the interest are in steady state solutions and integrated properties such as lift, drag and moment, lower order methods will suffice.

The computational cost (both the number of arithmetic operations and the memory requirement) scales with the dimension of the problem. As we turn to 3-D problems, the gain by using higher order methods will be even more pronounced. Also the computational gain by parallelizing the code will be reinforced as we turn to 3-D problems.

References

[1] B. Gustafsson and P. Olsson. Fourth-order difference methods for hyperbolic IBVPs. *J. Comput. Physics*, 117(1), 1995.

[2] D. W. Zingg, S. De Rango, M. Nemec, and T. H. Pulliam. Comparison of several spatial discretizations for the navier-stokes equations. *AIAA Paper*, 99-3260, 1999.

[3] Ken Mattsson. Boundary procedures for summation-by-parts operators. *Journal of Scientific Computing*, to appear, 2003.

[4] Ken Mattsson, Magnus Svärd, and Jan Nordström. Stable and accurate artificial dissipation. Technical report 2002-003, Dept. of Information Technology, Uppsala University, 2002.

[5] A. Harten. The artificial compression method for computation of shocks and contact discontinuities: III. self -adjusting hybrid schemes. *Math. Comp.*, 32:363–389, 1978.

[6] H.C. Yee, N.D. Sandham, and M.J. Djomehri. Low-dissipative high-order shock-capturing methods using characteristic-based filters. *Journal of Computational Physics*, 150:199–238, 1999.

[7] H.C. Yee, M. Vinokur, and M.J. Djomehri. Entropy splitting and numerical dissipation. NASA/TM-1999-208793 and 8th International Symposium on CFD, 5-10 September 1999, Bremen, Germany.

[8] B. Müller and Yee H.C. High order numerical simulation of sound generated by the Kirchhoff vortex. *Computing and Visualization in Science*, 4:197–204, 2002.

[9] M. Gerritsen and P. Olsson. Designing an efficient solution strategy for fluid flows. *J. Comput. Physics*, 129:245–262, 1996.

[10] B. Sjogreen. Multiresolution wavelet based adaptive numerical dissipation control for shock-turbulence computations. Technical report 01.01, RIACS, NASA Ames Research Center, 2000.

[11] Bertil Gustafsson, Heinz-Otto Kreiss, and Joseph Oliger. *Time dependent problems and difference methods.* Wiley, New York, 1995.

[12] H.-O. Kreiss and G. Scherer. Finite element and finite difference methods for hyperbolic partial differential equations. *Mathematical Aspects of Finite Elements in Partial Differential Equations., Academic Press, Inc.*, 1974.

[13] H.-O. Kreiss and G. Scherer. On the existence of energy estimates for difference approximations for hyperbolic systems. Technical report, Dept. of Scientific Computing, Uppsala University, 1977.

[14] Mark H. Carpenter, David Gottlieb, and Saul Abarbanel. The stability of numerical boundary treatments for compact high–order finite–difference schemes. *J. Comput. Phys.*, 108(2), 1994.

[15] Mark H. Carpenter, Jan Nordström, and David Gottlieb. A stable and conservative interface treatment of arbitrary spatial accuracy. *J. Comput. Phys.*, 148, 1999.

[16] Jan Nordström and Mark H. Carpenter. Boundary and interface conditions for high order finite difference methods applied to the euler and navier–stokes equations. *J. Comput. Phys.*, 148, 1999.

[17] Jan Nordström and Mark H. Carpenter. High-order finite difference methods, multidimensional linear problems and curvilinear coordinates. *J. Comput. Phys.*, 173, 2001.

[18] Bo Strand. *High–Order difference approximations for hyperbolic initial boundary value problems*. PhD thesis, Uppsala University, Dep. of Scientific Computing, Uppsala Univ., Uppsala, Sweden, 1996.

[19] L.-E. Eriksson. Boundary conditions for artificial dissipation operators. Technical Report FFA TN 1984-53, The Aeronautical Research Institute of Sweden, Aerodynamics Depertment, Stockholm, Sweden, 1984.

[20] G. Strang. Accurate partial difference methods II. non-linear problems. *Num. Math.*, 6:37–46, 1964.

[21] Pelle Olsson. *The Numerical Behavior of stable High-Order Finite Diffrence Methods*. PhD thesis, Uppsala University, Dep. of Scientific Computing, Uppsala Univ., Uppsala, Sweden, 1992.

[22] G. Erlebacher, M. Y. Hussaini, and C.-W. Shu. Interaction of a shock with a longitudinal vortex. *J. Fluid. Mech.*, 337:129–153, 1997.

[23] E. Petrini, G. Efraimsson, and J. Nordström. A numerical study of the introduction and propagation of a 2–d vortex. Technical Report FFA TN 1998–66, The Aeronautical Research Institute of Sweden, Bromma, Sweden, 1998.

[24] Bertil Gustafsson. The convergence rate for differnce approximations to general mixed initial boundary value problems. *SIAM J. Numerical Analysis*, 18(2):179–190, 1981.

[25] Heinz-Otto Kreiss and Joseph Oliger. Comparison of accurate methods for the integration of hyperbolic equations. *Tellus XXIV*, 3, 1972.

[26] Peter Eliasson. A Navier-Stokes solver for unstructured grids. Scientific report FOI-R–0298–SE, The Swedish Defense Research Agency, Stockholm, 2001.