33RD CONGRESS
OF THE INTERNATIONAL COUNCIL
OF THE AERONAUTICAL SCIENCES
STOCKHOLM, SWEDEN, 4-9 SEPTEMBER, 2022

ICAS 2022 SWEDEN

ENHANCING CFD PREDICTIONS FOR THE GRIPEN AIRCRAFT

Peter Eliasson[1], Per Weinerfelt[1] & Frank Bramkamp[2]

[1]Saab, SE-581 88 Linköping, Sweden
[2]National Supercomputer Centre (NSC), SE-581 83 Linköping, Sweden

Abstract

We present recent developments and improvements of the efficiency carrying out aerodynamic CFD calculations at Saab. The enhancement comes from an implicit discretization in time of the governing equations where the linear system of equations that arise from the linearization is solved iteratively and efficiently with the help of an external numerical library. The new approach is implemented in Saab's in-house CFD solver M-Edge for unstructured grids. We show comparative results for several test cases with focus on Gripen. Most calculations lead to a factor of three or more in computational speed-up. This holds for both steady state and time accurate calculations. Furthermore, we can also get improved steady state convergence with less or no remaining oscillations in the computed results, hence leading to improved accuracy as well.

Keywords: CFD, implicit time integration, convergence acceleration, Gripen

1. Introduction

Aerodynamic prediction and design at Saab rely more and more on large scale CFD simulations. For external aerodynamics, we typically perform calculations in the entire flight envelope where we vary Mach number, angle of attack, side slip angle, altitude, control surface deflections etc. Most aerodynamic calculations of the external shape of an aircraft are steady state viscous calculations using the Reynolds Averaged Navier-Stokes (RANS) equations with turbulent closure from a turbulence model. Our main product is the Gripen aircraft. A typical RANS unstructured mesh with a good resolution of the boundary layer ($y^+ \sim 1$) has about 60×10^6 nodes and about a factor three more elements. Although a single RANS calculation on such a grid can be done in hours, thousands of calculations are required to cover all necessary combinations of ingoing parameters which requires major computing resources.

Unsteady calculations are also becoming more and more common as computational resources increase together with improved modelling capability and efficiency of the underlying CFD software. Typical applications are control surface deflections, calculations of dynamic derivatives, hysteresis effects at high angles of attacks etc. Also, the number of turbulent scale-resolving calculations (hybrid RANS-LES) increase due their improved prediction capability of large scale separated flows. Example of such simulations comprise computations over landing gears, engine inlet calculations, weapon bays etc.

Saab develops its own CFD software, M-Edge, that originates from the Edge flow solver [7]. The development is carried out together with a few collaborative partners at universities and research institutes under Saab lead. Saab has a close cooperation with the National Supercomputer Centre (NSC) in Linköping for many years. In a previous work, NSC has analyzed the efficiency of the code, improved its parallel scaling and identified and removed bottlenecks in the code to speed-up computations. During recent years, Saab and NSC have been working on a new implicit time integration method to further accelerate the convergence of the computations. This work has now been validated internally at Saab and is already in use for various applications. Here we describe this new implicit approach, and we show a number of cases to demonstrate that the new scheme

works very well and achieves a substantial increase in computational efficiency compared to previous algorithms.
The paper is organized as follows: In the next section we present the finite volume scheme for unstructured grids. Then follows a section about steady state time integration. We outline explicit Runge-Kutta integration, followed by line-implicit integration, and finally we put emphasis on describing the new implicit scheme. Next, a section of how to extend the steady state solution technique to time accurate problems by dual time stepping follows. After that we present a short section on the M-Edge flow solver together with some practical experiences and recommendations on what we have found to be important for the implicit approach to be efficient and robust. Finally, we present a set of steady and unsteady flow cases demonstrating the improved efficiency of the new scheme, followed by some concluding remarks.

2. Finite Volume Method

We consider the governing equations of the Navier-Stokes equations in integral form

$$\int_V \frac{\partial Q}{\partial t} dV + \oint_{\partial V} F dS = 0 \tag{1}$$

with the conservative variables Q (density, momentum and total energy) and F the sum of the convective and viscous terms of the underlying continuity, momentum and energy equations.
Consider the discrete counterpart of Equation (1) with a control volume V_i for an arbitrary discrete node with subscript i. The spatial discretization of the Navier-Stokes equations in Equation (1) using a finite volume formulation on an unstructured grid for this node may be written in semi-discrete form as follows:

$$V_i \frac{dQ_i}{dt} + \sum_{k \in i} F_{ik} S_{ik} + F_{i_{bc}} S_{i_{bc}} = V_i \frac{dQ_i}{dt} + R_i(Q) = 0 \tag{2}$$

where Q_i contains the conservative variables for node i, $F_{ik}S_{ik}$ is the flux between two connected nodes i and k (denoted $k \in i$), $F_{i_{bc}}S_{i_{bc}}$ is the boundary flux at node i. The boundary flux is only included if the node is located on a boundary in order to close the control volume. If node i is an interior node this term is zero. The fluxes are summed up to the residual for node i, $R_i(Q)$. The formulation is a so-called dual grid formulation where a dual grid forms the control volume to a primary grid. As an example, Figure 1 illustrates the dual grid of a triangular grid. The dual grid is computed in a preprocessing step that sums up contributions from surrounding elements to a node such that a single control surface is obtained for each edge, e.g. S_{01} between nodes 0 and 1 in Figure 1. A more thorough description of the approach above can be found in e.g. [5].
We leave out sub index when we refer to an entire vector for all unknowns. Note that the vector Q_i may contain more than 5 unknowns, e.g. additional unknowns related to a turbulence model. Typically, turbulence models include additional source terms as well, which have been omitted in Equation (2) for the sake of brevity.

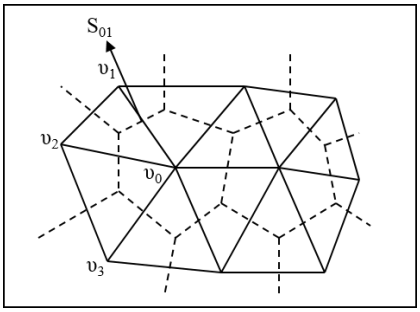

Figure 1 – Primary (solid line) triangular grid and its dual grid (dashed line).

3. Steady State Time Integration

First, we are looking for a steady state solution in time to Equation (2).

3.1 Explicit Runge-Kutta Time Integration

The governing equations may be integrated explicitly in time using a multi-stage Runge-Kutta method

$$\begin{aligned} Q^{(0)} &= Q^n \\ Q^{(s)} &= Q^{(0)} - \alpha_s \frac{\Delta t}{V} R(Q^{(s-1)}); \quad s = 1, ..., p \\ Q^{n+1} &= Q^{(p)} \end{aligned} \tag{3}$$

where α_s denote the stage coefficients of the p-stage scheme. To reach asymptotically steady state, the Runge-Kutta coefficients are chosen for optimal damping of high-frequency errors. We employ a 3-stage scheme with $\alpha_1 = 2/3, \alpha_2 = 2/3, \alpha_3 = 1$.

To accelerate the convergence to steady state, we use local time steps, which means that Δt varies from node to node. To further accelerate convergence, the explicit time integration scheme is combined with multigrid. The current geometric multigrid scheme is based on the Full Approximate Storage (FAS) approach for the non-linear problem. The coarser grids are generated as part of a preprocessing step where coarser grids are generated by agglomeration (fusing) of finer grid control volumes, see e.g. [3] for more details.

Furthermore, implicit residual smoothing is applied to extend the admissible CFL number.

3.2 Line-Implicit Runge-Kutta Time Integration

The explicit Runge-Kutta scheme can be combined with a line-implicit time integration method along selected grid lines where the mesh is highly stretched. This allows to increase the time step and to remove the restriction from the small length scales normal to the wall, which generally leads to a speed-up of the steady state convergence.

The residual is split into two parts, $R(Q) = R^E(Q) + R^I(Q)$, where the implicit residual $R^I(Q)$ contains the fluxes along the implicit lines only and $R^E(Q)$ contains all remaining fluxes treated explicitly. The multi-stage Runge-Kutta method can then be formulated as

$$\begin{aligned} Q^{(0)} &= Q^n \\ \left(I + \gamma_s \frac{\Delta t}{V} \frac{\partial R^I(Q)}{\partial Q}\right)(Q^{(s)} - Q^{(s-1)}) &= Q^{(0)} - Q^{(s-1)} - \alpha_s \frac{\Delta t}{V} R(Q^{(s-1)}); \quad s = 1, ..., p \\ Q^{n+1} &= Q^{(p)} \end{aligned} \tag{4}$$

where γ_s denote implicit stage coefficients. For the explicit 3-stage scheme we use $\gamma_1 = 1, \gamma_2 = 1, \gamma_3 = 0.8$. As in the case of the fully explicit scheme, we combine this method with multigrid, where we employ the line-implicit integration on the finest grid only. We refer to [8] for further details and numerical results.

3.3 Implicit Time Integration

For steady fluid flow, the time accurate description of the transient solution is immaterial. Time plays the role of an iteration parameter to achieve an asymptotic steady state solution in the computation. Therefore, the solution is advanced in time by an implicit backward Euler method.

The governing equations can then be expressed as:

$$(Q^{n+1} - Q^n)\frac{V}{\Delta t} + R(Q^{n+1}) = 0. \tag{5}$$

Linearizing the residual leads to the following linear system of equations:

$$J(Q^n)\,\Delta Q^n = -R(Q^n) \tag{6}$$

with

$$J(Q^n) = \frac{V}{\Delta t} I + \frac{\partial R}{\partial Q}(Q^n), \quad \Delta Q^n = Q^{n+1} - Q^n. \tag{7}$$

In practice, we employ an under-relaxation of the solution update:

$$Q^{n+1} = Q^n + \alpha \Delta Q^n, \quad \alpha \leq 1 \tag{8}$$

To our findings, $\alpha = 0.5$ offers a robust and efficient scheme.
The linear system of Equation (6) is solved by an iterative Krylov subspace method. In the present study, we employ the GMRES algorithm [15], preconditioned by an incomplete LU-factorization, namely ILU(0). For parallel processing, the ILU preconditioner is embedded in an additive Schwarz method. The implementation of the Krylov method is based on the PETSc software library [2] developed at Argonne National Laboratory.

3.3.1 Jacobian Evaluation

The exact linearization of the spatial discretization is very complicated. Amongst others, it requires the full linearization of the artificial dissipation operator which is highly non-linear in nature and not readily accessible in a closed form. Furthermore, the memory requirement of the exact Jacobian is extremely high. In order to alleviate these problems, the Jacobian of the spatial discretization is based on a simplified, lower order approximation. It solely uses solution information provided by compact stencils, which only couple direct neighbors. Hence, the assembled Jacobian is a blocked matrix where the number of non-zero entries is equivalent to twice the number of edges plus the number of nodes (diagonal entries). Each entry is an $N \times N$ matrix where $N \geq 5$ is the number of unknowns in a node.
We assume that all convective terms of the mean flow and turbulence equations are discretized by a central discretization with added numerical dissipation. The Jacobian of the central part of the convective fluxes in Equation (2) are computed as the exact derivatives of the discrete numerical fluxes since they only involve nearest neighbors. The Jacobians of the viscous fluxes only involve the normal derivatives, [5], and neglect the tangential derivatives. Further, the dynamic viscosity is considered constant, so is the turbulent dynamic viscosity. The spectral radius of the linearized turbulent source terms is estimated and added to the diagonal part of the Jacobian. It is also essential that the Jacobians of all boundary fluxes in Equation (2) are included. The formulation of these Jacobians depend on the specific boundary condition.
For the linearization of the artificial dissipation of the convective flux term, the dissipation operator is approximated by the upwind dissipation of the Roe scheme [14]. Assuming a first order method in space between nodes i and k this can be expressed as:

$$d^{\text{Roe}} = -\frac{1}{2} \left| A^{\text{Roe}} \right| (Q_k - Q_i). \tag{9}$$

A^{Roe} is the so-called Roe matrix and is considered constant when deriving the Jacobian. Then, the approximate Jacobian of the dissipation can be expressed by

$$\frac{\partial d^{\text{Roe}}}{\partial Q_i} = \frac{1}{2} \left| A^{\text{Roe}} \right|, \quad \frac{\partial d^{\text{Roe}}}{\partial Q_k} = -\frac{1}{2} \left| A^{\text{Roe}} \right|. \tag{10}$$

The absolute Roe matrix is expressed in terms of the associated left and right eigenvector matrix and its corresponding eigenvalues:

$$\left| A^{\text{Roe}} \right| = G |\Lambda| G^{-1} \tag{11}$$

with the diagonal matrix of the absolute eigenvalues

$$|\Lambda| = diag(|\lambda_{1-3}|, |\lambda_4|, |\lambda_5|). \tag{12}$$

The eigenvalues of the convective fluxes can be determined as

$$\lambda_{1-3} = u_n, \lambda_4 = u_n + a, \lambda_5 = u_n - a \tag{13}$$

where u_n is the normal velocity and a is the speed of sound. The immediate use of these eigenvalues generally leads to an unstable discretization, particularly when an eigenvalue vanishes, i.e. $\lambda_i \approx 0$.

These instabilities typically occur near stagnation points and in the vicinity of shock waves. To avoid such problems, a so-called entropy fix is introduced. The absolute eigenvalues are replaced by some effective values

$$|\Lambda|_{\text{ef}} = diag(|u_n|_{\text{ef}}, |u_n + a|_{\text{ef}}, |u_n - a|_{\text{ef}}) \tag{14}$$

so that the effective Roe matrix reads as following:

$$\left|A^{\text{Roe}}\right|_{\text{ef}} = G\,|\Lambda|_{\text{ef}}\,G^{-1}. \tag{15}$$

Here, the entropy fix is based on a simple cutoff value, that is expressed in terms of the local spectral radius $(|u_n| + a)$:

$$|u_n|_{\text{ef}} = \max\{|u_n|, \delta^{\text{IMP}}(|u_n| + a)\}, \quad |u_n \pm a|_{\text{ef}} = \max\{|u_n \pm a|, \delta^{\text{IMP}}(|u_n| + a)\} \tag{16}$$

To our experience, a value of $\delta^{\text{IMP}} = 0.3$ is a good compromise between robustness and efficient convergence. The Jacobian of the convective upwind dissipation for a turbulence model is expressed similarly to the Roe scheme. For the Spalart-Allmaras turbulence model [18] we use the following expression:

$$d_{tu}^{\text{Roe}} = -\frac{1}{2}\,|\lambda_{tu}|_{\text{ef}}((\rho\tilde{\nu})_k - (\rho\tilde{\nu})_i). \tag{17}$$

where $\rho\tilde{\nu}$ is the turbulence conservative variable. Then, the corresponding Jacobians are given by

$$\frac{\partial d_{tu}^{\text{Roe}}}{\partial(\rho\tilde{\nu})_i} = \frac{1}{2}\,|\lambda_{tu}|_{\text{ef}}, \quad \frac{\partial d_{tu}^{\text{Roe}}}{\partial(\rho\tilde{\nu})_k} = -\frac{1}{2}\,|\lambda_{tu}|_{\text{ef}}. \tag{18}$$

The absolute eigenvalue is assumed to be constant with respect to the linearization. We choose

$$|\lambda_{tu}|_{\text{ef}} = |u_n| + 0.1\,\|u\|_2^2 + \delta^{\text{IMP}}(|u_n| + a) \tag{19}$$

3.3.2 CFL Evolution Strategy

To accelerate convergence to steady state, the numerical solution is advanced using local time steps. In case of explicit time integration, the CFL number is set to a constant value for all non-linear iterations (time steps). The time step is proportional to the smallest length scale of a computational cell and typically CFL ~ 1 for reasons of numerical stability. With the line-implicit time integration the length scale limitation from the small edges along the lines is removed leading to locally larger time steps.

For the fully implicit time integration there is theoretically no upper bound on the time step and CFL number. However, a high CFL number and a large time step can only be used if the solution is relatively close to its converged steady state solution. Therefore, the CFL number is varied between a minimum and a maximum value during the iteration process:

$$CFL^{min} \leq CFL(t^n) \leq CFL^{max}$$

The choice of the CFL number has to balance the requirements for a robust numerical scheme and fast convergence. In the current approach, we increase the CFL number by a certain factor at each time step, until a prescribed maximum value CFL^{max} is reached

$$CFL(t^n) = \beta \cdot CFL(t^{n-1}),\ \beta \geq 1 \tag{20}$$

β is typically selected in the range $\beta \in [1.05, 1.5]$, where $\beta = 1.2$ is our choice of preference. Usually, we employ $CFL^{min} = 1.5$ and $CFL^{max} = 10^4$.

The CFL evolution strategy is employed for steady state calculations mainly. For unsteady calculations, described below, with an initial solution sufficiently developed in time, we typically employ a constant and high CFL number from the beginning.

4. Unsteady Time Integration

The steady state time integration schemes above can be extended to unsteady time accurate calculations by a so-called dual time stepping approach introducing a fictitious time derivative in dual time. We demonstrate this for the 2^{nd} order backward difference scheme (BDF2) [11] that is employed in this paper. We stress that this approach can also be used for diagonally implicit multi-stage Runge-Kutta schemes [9] and fully implicit Runge-Kutta schemes [12]. A requirement is that the implicit scheme is both A- and L-stable.

We discretize Equation (2) with the BDF2 scheme, assuming that the geometry is fixed in time with a constant volume V,

$$V\frac{3Q^{n+1}-4Q^{n}+Q^{n-1}}{2\Delta t}+R(Q^{n+1})=0. \tag{21}$$

We introduce a new time derivative in fictitious time τ and denote the unknown Q^{n+1} with Q^{*}

$$V\frac{dQ^{*}}{d\tau}+V\frac{3Q^{*}-4Q^{n}+Q^{n-1}}{2\Delta t}+R(Q^{*})=V\frac{dQ^{*}}{d\tau}+\tilde{R}(Q^{*})=0 \tag{22}$$

where $\tilde{R}(Q^{*})=R(Q^{*})+V(\frac{3}{2}Q^{*}-2Q^{n}+\frac{1}{2}Q^{n-1})/\Delta t$ and $Q^{*}\rightarrow Q^{n+1}$ as $\partial Q^{*}/\partial\tau\rightarrow 0$. For each physical time step Δt, we can apply the various steady state time integrators described above to Equation (22) to be iterated towards steady state in dual time τ. It should be noted that for the implicit approaches described above, an additional contribution $\frac{3V}{2\Delta t}I$ is added to the Jacobian.

5. The Flow Solver M-Edge

The CFD code employed for the present simulations and used internally at Saab is the M-Edge flow solver, which originates from the Edge flow solver [7]. The discretization in space is a finite volume formulation for unstructured grids as described above where a median dual grid forms the control volumes with the unknowns allocated in the grid nodes. The various available time integration methods are also described above. A large number of turbulence models are available. Throughout this paper, a central discretization is used for the convection to which a small amount of numerical dissipation is added. This applies to the mean flow and for the turbulence equations. The viscous terms are discretized by a compact discretization of the normal derivates and remaining tangential derivates are obtained from Gauss-Green integral formulation of gradients to have a full viscous operator. There are numerous boundary conditions available in M-Edge for walls, external boundaries and periodic boundaries. All of these boundary conditions are specified weakly, which means, that the unknowns on nodes located on a boundary are unknowns like any other unknown in the interior [6], [13]. The boundary conditions are specified through the boundary flux in Equation (2).

A preprocessor creates the dual grid, coarser grids for multigrid, stretched lines for the line-implicit approach, wall distances and other quantities required by the flow solver. Last but not least, it splits the computational grid by domain decomposition. The splitting ensures that the implicit lines are not broken, so that entire lines belong to a specific partition.

5.1 Some Specifics for the Implicit Scheme

Several aspects were found to be important to make the new implicit scheme efficient and robust. We list some of these findings below.

- We do not combine the implicit scheme with multigrid, although it is possible. Multigrid has very small or no effect in implicit time advancement using large time steps. It has larger effect on explicit time integration schemes that have good high frequency smoothing and where the coarser grids in practice serve to increase the relatively small explicit time steps.
- Although the implicit lines are not used for the implicit scheme, it was found important for the robustness not to partition the domain along the lines where the grid is highly stretched. Removing this requirement can lead to divergence or deterioration in the rate of convergence.
- Some Jacobians for the boundary condition fluxes were challenging to derive. We have used the external software *Maple* to derive closed expressions for some boundary fluxes, e.g. for external

characteristic boundary conditions and for subsonic inflow conditions where total states are specified. We also found that some boundary conditions required a small amount of additional local numerical dissipation which was added to the Jacobian, e.g. the external characteristic boundary conditions.

- The linear system in Equation (6) should be solved "accurate enough" by the iterative GMRES process. An excessively accurate solution of the linear system will cost additional time and not increase the convergence of the non-linear iteration drastically. On the other hand, the solution should be accurate enough to avoid divergence. We typically reduce the error of the linear system to 10% of its initial value, which offers a good compromise between efficiency and robustness.

- We separate the mean flow and turbulence equations and treat their Jacobians and iterative solutions separately. Treating all equations together did not results in a computational speed-up, on the contrary it often resulted in computational overhead due to larger Jacobians. The implicit scheme is implemented for general turbulence models, that are based on one or two transport equations. It is most efficient for models based on the model by Spalart-Allmaras, [18] [17]. Turbulence models of $k-\omega$ type often show somewhat slower convergence rates and may require lower CFL numbers as well.

- We found that time dependent calculations with small time steps, e.g. hybrid RANS-LES calculations, benefit in terms of steady state convergence in dual time from a smaller and scalar dissipation. More specifically, where the elements of the eigenvalues in $\left|A^{\mathrm{Roe}}\right|$ in Equation (9) are replaced by its largest eigenvalue (spectral radius) and where the factor $1/2$ is replaced by a smaller value. A contributing reason why a smaller numerical dissipation is beneficial is that the additional source term from BDF2 in Equation (22) contribute to the stability and diagonal dominance of the assembled Jacobian matrix.

- The price to pay for the new implicit scheme is a substantially increased computer memory consumption. The increase is about a factor of four compared to the explicit multigrid approach with or without line-implicit integration. The additional memory requirement is due to the storage of the assembled Jacobian, the ILU preconditioning matrix and the vast number of additional unknowns required by GMRES. The increased memory consumption is usually not a problem for the computer architectures currently in use, but a user should consider that the computational domain has to be split in a sufficient number of partitions to fit into available memory.

6. Numerical Results

Numerical results are presented for turbulent viscous calculations on a number of test cases. Most of the cases involve steady state calculations, some are unsteady calculations using dual time stepping with the BDF2 method. All calculations employ no-slip adiabatic wall conditions and characteristic far field boundary conditions.

Unless otherwise stated, the turbulence model for the RANS calculations all use the Spalart-Allmaras turbulence model [18]. Comparisons are made between the computations obtained with the new implicit scheme and with computations using 3 levels of multigrid with line-implicit time integration on the finest grid that up to now has been the standard scheme used at Saab. All steady state calculations are started from free stream flow conditions. For steady state calculations the CFL numbers start from a low value of $CFL = 1.5$ and ramped up to a maximum value of $CFL = 10^4$ with a factor of 1.2 per iteration. Unsteady calculations with a given initial flow solution use the high CFL already from the beginning whereas the explicit calculations employ $CFL = 1.0$. All calculations use a central discretization with added numerical dissipation for the mean flow and turbulence equations. The convergence of the density residual is displayed for most cases. The density residual is defined such that it corresponds to the time derivative of the density. Usually, the logarithm of the L2 norm of the residual is displayed.

6.1 Transonic Flow over RAE 2822 Airfoil

The first test case is the two-dimensional flow over the RAE 2822 airfoil, Case 10 [4]. The free stream conditions are $M_\infty = 0.754$, $\alpha = 2.57°$, $Re = 6.2 \times 10^6$. A structured grid with 22×10^3 nodes is employed. Figure 2 shows a detail of the grid and the Mach number contours of the converged solution. Computations are carried out using 1 and 4 partitions, i.e. serial and parallel calculations with 4 cores.

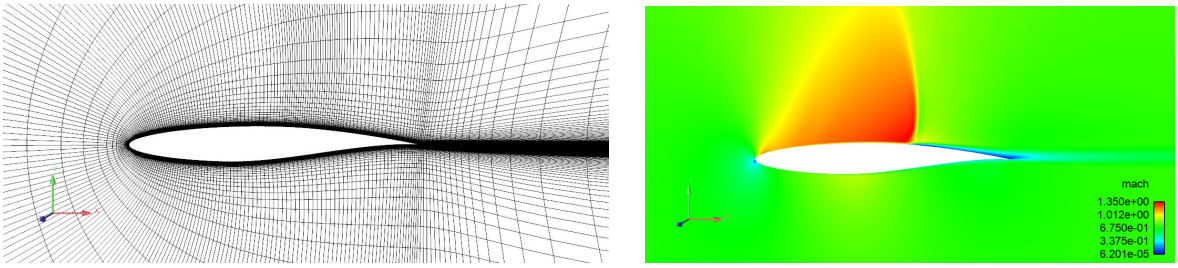

Figure 2 – RAE 2822 airfoil. Grid (left), Mach number contours (right).

The convergence of the density residual and the lift coefficient as function of computational time are depicted in Figure 3. The implicit scheme is more than 20 times faster than the M-Edge standard scheme, using explicit/line-implicit time integration with multigrid. The explicit/line-implicit calculations suffer from the sharp trailing edge and the C-type grid, which limits the CFL number in this region.

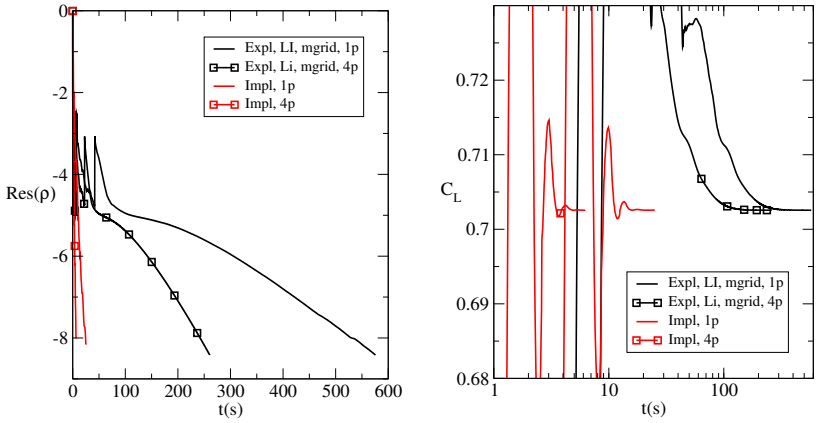

Figure 3 – RAE2822 steady state convergence of density residual as function of elapsed time (left). Convergence of lift coefficient as function of elapsed time (right).

6.2 Transonic Flow over NACA0012 Airfoil

Next test case is the two-dimensional flow over the NACA0012 airfoil, using unstructured hybrid grids with quadrilateral cells close to the airfoil and triangular cells further away from the airfoil. Three different grids are available with 51×10^3, 54×10^3 and 57×10^3 nodes with normal distance between the first two nodes being 10^{-5}, 10^{-6} and 10^{-7} chord lengths, respectively. The number of quadrilateral layers is 35, 45 and 55, respectively. For all three grids, the airfoil surface grid (315 nodes) as well as the triangular part of the mesh remain the same. The maximum aspect ratio, i.e. the ratio between the largest and smallest edge of a cell, is 1.5×10^3, 1.5×10^4 and 1.5×10^5. The test case has been used in a previous study to verify that grid independent convergence can be achieved using a line-implicit formulation [8].

The flow conditions are the same as for the RAE 2822 airfoil, $M_\infty = 0.754$, $\alpha = 2.57°$, $Re = 6.2 \times 10^6$. All calculations are performed in serial mode using a single core. Figure 4 shows the convergence history for the explicit time integration scheme with multigrid and with/without line-implicit integration. Without the line-implicit integration, the convergence slows down considerably as the near wall grid is refined and the aspect ratio increases. Introducing the line-implicit integration, the convergence is nearly independent of the stretching of the mesh. The implicit scheme converges fastest for all grids.

There is a minor degradation of the convergence wih the implicit scheme for the grid with the highest aspect ratio. The speed-up of the implicit scheme is in a range from a factor of 4.5 to 9 compared to the explicit multigrid scheme with line-implicit time integration.

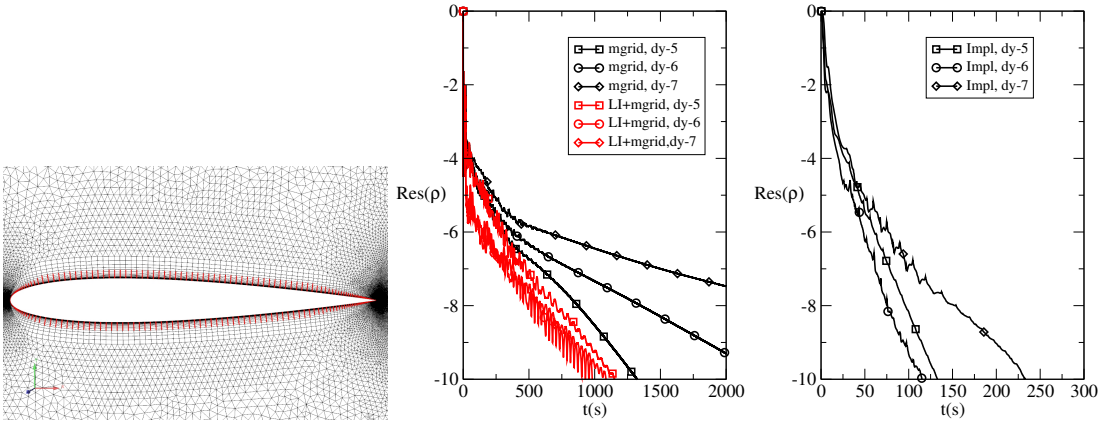

Figure 4 – Unstructured grid near NACA0012 airfoil, implicit lines in red (left). Convergence of density residual with explicit scheme with multigrid (black) and multigrid and line implicit (red) as function of elapsed time (mid). Convergence of density residual as function of elapsed time with implicit scheme (right).

6.3 Internal Flow in a Nozzle

Next, we consider the internal flow in a two-dimensional nozzle. The test case is used to validate the implicit scheme for different types of boundary conditions, as these are non-trivial to derive. On the three left inflow boundaries, the total pressure and the total temperature are specified together with the direction of the flow in the positive x-direction. On the top boundary, characteristic boundary conditions are used, the static pressure is specified on the right outflow boundary and a symmetry boundary condition on the bottom boundary. There is a supersonic region in the contraction, the flow is otherwise subsonic. The computational grid has about 38×10^3 nodes, consisting of 12×10^3 quadrilateral elements and 49×10^3 triangular elements. Figure 5 shows the computational grid, the Mach number distribution and the convergence of the density residual. The calculations are executed in serial mode using a single core. The implicit scheme is about a factor of 5.5 faster compared to the explicit scheme with line-implicit integration.

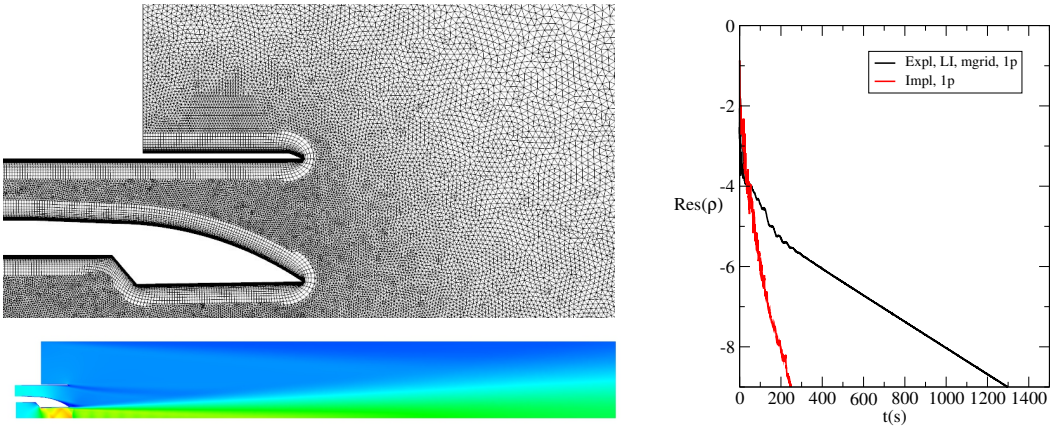

Figure 5 – Unstructured grid in left part of nozzle (left top). Mach number distribution (left bottom). Convergence of density residual as function of elapsed time (right).

6.4 3D Flow over the ONERA M6 Wing

As first 3D test case, we consider turbulent flow over the ONERA M6 wing [16]. The free stream flow conditions are $M_\infty = 0.84$, $\alpha = 3.06^\circ$, $Re = 14.6 \times 10^6$. The computational grid contains 0.92×10^6

nodes with 1.18×10^6 prismatic elements near the wing and 1.87×10^6 tetrahedral elements outside the prismatic region. There are about 40×10^3 triangular elements on the surface of the wing and about 11×10^3 triangular and quadrilateral elements on the plane of symmetry. For the line-implicit integration, each line contains about 27 nodes on average, with 19.8×10^3 lines in total, which corresponds to the total number of nodes on the wing surface. The maximum aspect ratio of the grid is 12×10^3.

All calculations were carried out in parallel using 32 partitions and processors (cores). The convergence history of the density residual and the lift coefficient are displayed in Figure 6. Comparisons are made with the explicit scheme using multigrid, with and without line-implicit time integrations. We can see that the calculation is accelerated by about a factor of two using the line-implicit time integration, compared to the standard explicit scheme. Another factor of 4 is obtained with the new implicit scheme. A well converged steady state solution is here obtained in about one minute computing time.

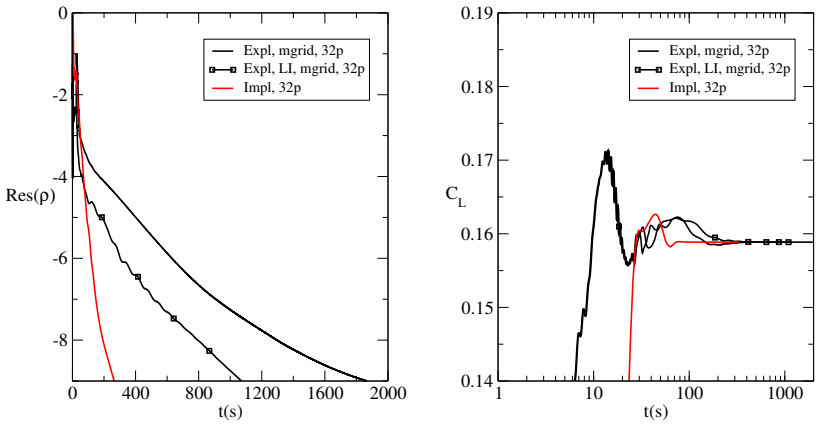

Figure 6 – ONERA M6 wing, steady state convergence of density residual as function of elapsed time (left). Convergence of lift as function of elapsed time (right).

6.5 Steady State Gripen Calculations

One of the main purposes for the new implicit scheme is to make CFD calculations for the Gripen aircraft more efficient. In the following, we present results using a representative grid for the Gripen with about 60×10^6 nodes, partitioned in 256 domains for the parallel computation. The grid contains 113×10^6 prismatic and 28×10^6 tetrahedral elements with about 55 layers of prismatic cells.

We consider transonic flow conditions with $M_\infty = 0.8, \alpha = 4^\circ$ at an altitude of 6000 meters. All simulations are started from free stream conditions and with turbulence effects modeled by the Spalart-Allmaras model. In addition to wall and far field boundary conditions, the static pressure is specified at the engine inlet which is a subsonic outflow boundary. The engine outlet is modeled as a subsonic inflow boundary, which requires the total temperature, total pressure and flow direction to be prescribed. The explicit calculations, combined with line-implicit integration, employ full multigrid with 3 multigrid levels which up to now has been the default setting. In the past, calculations for such a complex configuration at transonic conditions did not converge entirely to machine accuracy.

The convergence of the residual as function of elapsed time and non-linear iterations can be seen in Figure 7, which also displays the convergence of the normal force.

For the explicit/line-implicit scheme, the density residual converges about 6 orders of magnitude before it goes into a limit cycle oscillation. The implicit scheme shows a different convergence behaviour. The method is able to converge the residual to machine accuracy. The difference in the residual history is also reflected in the evolution of the normal force. The explicit scheme shows remaining oscillations of the normal force, while the implicit schemes rapidly converges to a constant value. Worth noting is that this constant value is not the same as an averaged value from the last iterations with the explicit scheme. The implicit scheme delivers fully converged aerodynamic forces and moments after 6-7 orders of magnitude reduction of the density residual, which corresponds to about one wall clock hour of computing time on 256 cores.

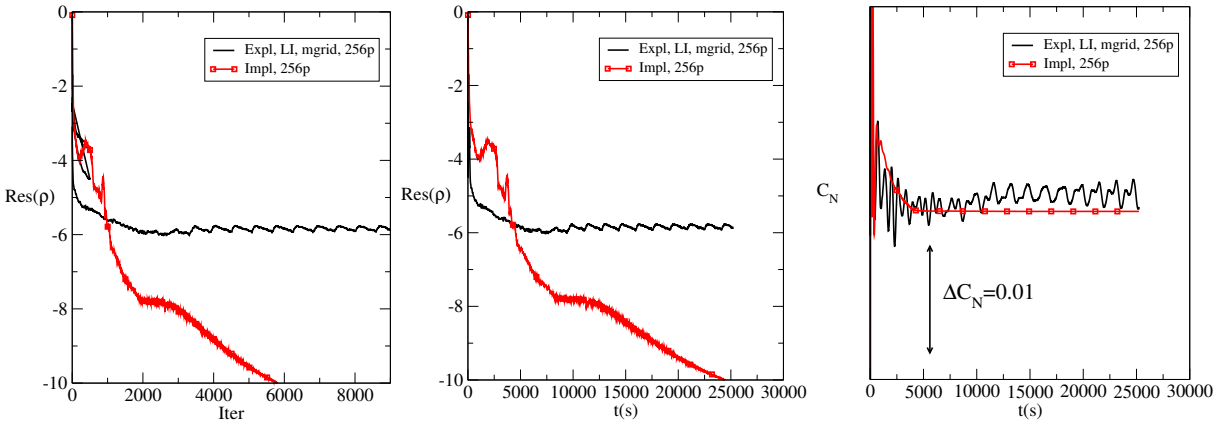

Figure 7 – Gripen density residual steady state convergence as function of iterations (left) and elapsed time (mid). Convergence of normal force as function of elapsed time (right).

For the explicit calculations, the remaining oscillations in the normal force originate primarily from vortices emanating from the gaps between the control surfaces and the fuselage. This effect is mainly visible in the temperature distribution as depicted in Figure 8. Streaks of high and low temperatures are clearly visible in this area at the aft, whereas no such temperature variations can be observed in the solution from the implicit scheme.

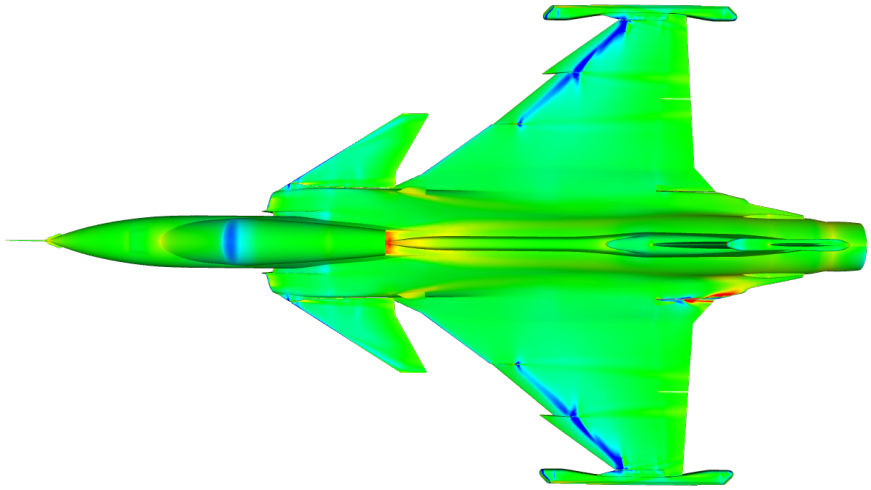

Figure 8 – Temperature distribution on upper side of Gripen. Upper half shows the implicit and converged solution. Lower half shows the not fully converged explicit/line-implicit solution.

6.6 Unsteady Gripen Calculations

Although there exists a steady state solution as shown above, the solution does not necessarily need to be stable in time. To investigate if this is the case, we have performed unsteady flow simulations starting from the converged steady state solution of the implicit scheme as initial condition. The mesh and the flow conditions are the same as mentioned above. In addition, we also perform unsteady calculations using the not fully converged steady state solution from the explicit scheme as initial condition. The goal is to investigate, if the difference in the initial conditions has an impact on the result of the unsteady simulation. Apart from the initial solution, everything else is the same. Within each time step, we converge the maximum density residual to the same level. A time step of $\Delta t = 10^{-4}s$ is used that provides good resolution in physical time with satisfactory steady state convergence in dual time. We proceed with the unsteady calculations for about $1s$ in time.

The time histories of the lift force coefficient reveal significant differences as shown in left and mid plots of Figure 9. Namely, the unsteady simulations result in two different cyclic time dependent solutions. Starting from the well converged steady state solution, the flow remains steady for about

half a second. After that, the lift starts to oscillate with a small amplitude around its initial value. Starting from the not fully converged steady state solution, the flow field ends in a different time dependent cyclic solution. Both the average value and the amplitude of the lift oscillation are larger than in the case, starting from the well converged solution. The oscillations are still substantially smaller than those of the non-converged steady state solution. The included time histories for the steady state solutions are fictitious, the iterations have been scaled to fit in the same plots. The mid plot in Figure 9 also reveals that the main part of the oscillations occur due to vortices in the vicinity of the gap between the fuselage and elevon as shown in Figure 8.

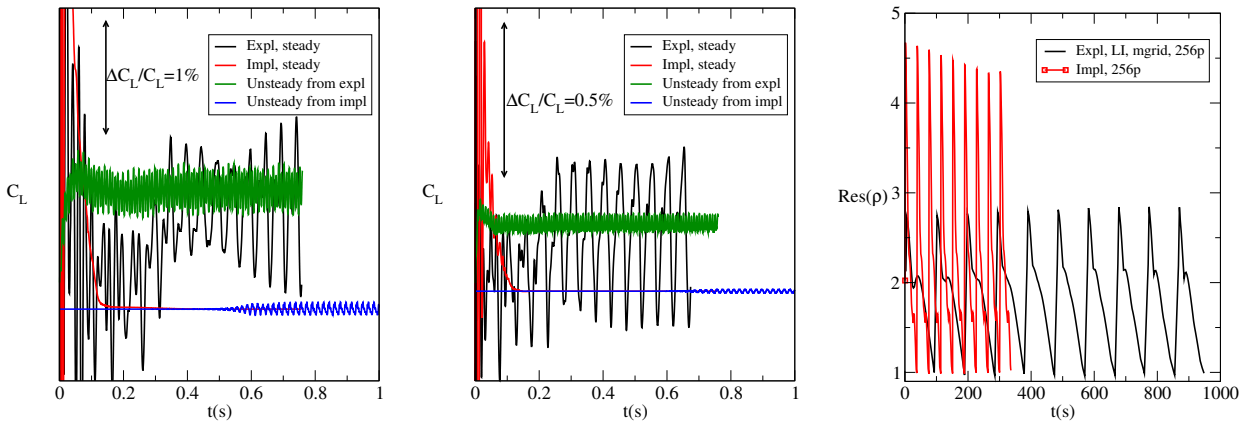

Figure 9 – Gripen time history of total lift (left) starting from a converged and a not fully converged steady state solution. Time history of lift on inner elevon (mid) Steady state convergence history included, in fictitious time. Convergence of maximum density residual in dual time vs. elapsed time during 10 time steps starting from the implicit steady state solution (right).

The unsteady calculations were carried out with the implicit time integration in dual time due to its lower computational cost. A calculation was also carried out with explicit/line-implicit time integrator in dual time to compare the efficiencies and to ensure that the dual time integrator did not affect the time dependent solution. The right plot in Figure 9 shows a comparison of the convergence of the maximum density residual versus elapsed time during 10 time steps starting from the implicit steady state solution when the solution has become oscillatory in time. The plot reveals that the implicit time integration in dual time is about a factor of 2.5 faster. It also reveals that the maximum density residual increases to a higher level with the implicit scheme when going from one time step to the next. The residual quickly reduces though.

6.6.1 Explanation to Multiple Solutions

The space discretization of the time dependent Navier-Stokes, Equation (2), leads to a large non-linear system of ODE:s. If we reformulate the equations in terms of the discrete primitive variable U, instead of Q, we get another non-linear system of ODE:s where the residual $\bar{R}(U)$ can be approximated by polynomials of U with high accuracy

$$\bar{R}_i(U) \approx \sum_{|\alpha| \le m \le 4} c_{i,\alpha} U^{\alpha}, \quad \alpha = (\alpha_1, \alpha_2, ..., \alpha_N), \quad |\alpha| = \sum_{k=1}^{N} \alpha_k, \quad \alpha_k \in \{0, 1, ...4\}. \tag{23}$$

The coefficients $c_{i,\alpha}$ are independent of time. In order to give an explanation to multiple solutions to the Navier-Stokes equations we study a simpler non-linear system of ODE:s, having a similar structure as the discrete Navier-Stokes equations, namely the van der Pol equation

$$\frac{d^2u}{dt^2} - \mu\left(1 - u^2\right)\frac{du}{dt} + u = 0. \tag{24}$$

Equation (24) can be rewritten as a first order system of two equations

$$\begin{aligned} \frac{du_1}{dt} &= u_2 \\ \frac{du_2}{dt} &= \mu\left(1 - u_1^2\right)u_2 - u_1 \end{aligned} \tag{25}$$

where $u_1 = u$ and $u_2 = \frac{du}{dt}$. One of the features of Equation (25) is that the solution approaches asymptotically a periodic solution for large t, which depends on μ but not on the initial solution (see Figures 10 and 11). This is in contrast to linear systems. The steady state solution to Equation (25) is $(u_1, u_2) = (0,0)$. The initial solution in Figure 10 (mid) and Figure 11 is very close to the steady state solution, however the asymptotic solution is far from that one.

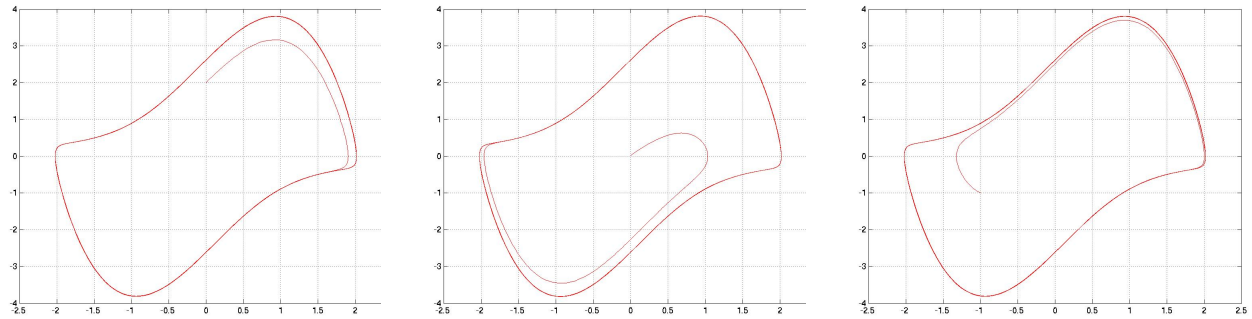

Figure 10 – Phase space plot, van der Pol's equation $\mu = 2$, different initial solutions.

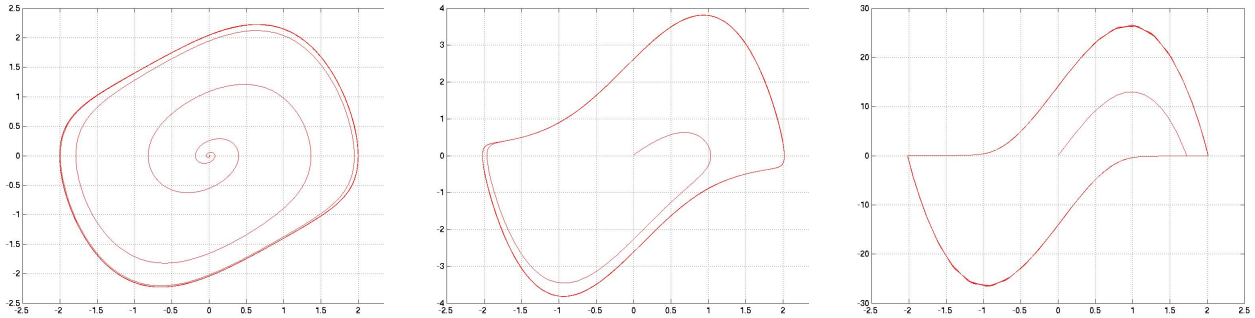

Figure 11 – Phase space plot, van der Pol's equation, different values of μ.

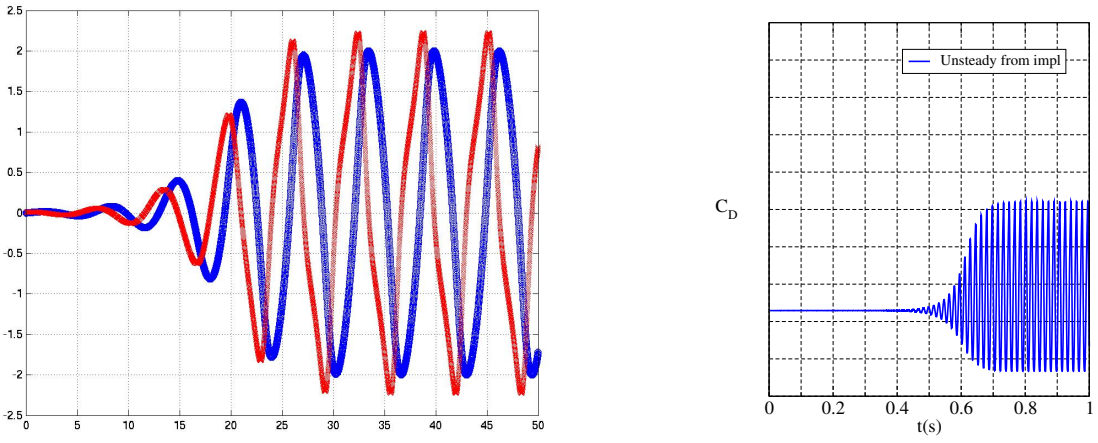

Figure 12 – Comparison of solutions to van der Pol's equation (left) and N-S equation for Gripen displaying total drag (right).

The system in Equation (25) can be extended to a system consisting of K independent systems of type Equation (25) according to

$$\begin{aligned} \frac{du_{2k-1}}{dt} &= u_{2k} \\ \frac{du_{2k}}{dt} &= \mu_k \left(1 - u_{2k-1}^2\right) u_{2k} - u_{2k-1}, \quad 1 \le k \le K \end{aligned} \tag{26}$$

Different time dependent solutions to Equation (26) can be obtained by starting from different initial solutions such as $(u_{1,0}, u_{2,0}, 0, 0, ..., 0, 0)$, $(0, 0, u_{3,0}, u_{4,0}, ..., 0, 0)$ or $(0, 0, 0, 0, ..., u_{2K-1,0}, u_{2K,0})$ $(u_{k,0} \neq 0)$. We saw in the previous section that Gripen calculations exhibit a similar behaviour (see Figure 12). Starting from a solution close to steady state we obtained asymptotically a periodic solution with finite amplitude whereas other start solutions resulted in different periodic solutions. An explanation for this is that the Navier-Stokes equations have the same structure as van der Pol's equation and can be expressed as a system similar to Equation (26).

6.7 Hybrid RANS-LES Calculations over a Cavity

The last case reported here involves scale-resolved turbulent calculations over the M219 cavity where we employ dual time stepping for each physical time step and compare convergence rates in dual time. The test case is well known and several references exist with a vast amount of numerical results, see e.g. [1], [10]. The flow conditions are $M_\infty = 0.85, Re = 6.8 \times 10^6$ where the Reynolds number is based on the cavity length. An unstructured grid with 6.2×10^6 nodes is used with 9.6×10^6 tetrahedral, 9.0×10^6 prismatic and a few pyramid elements. The geometry in the grid follows the geometry used of the wind tunnel where experimental results have been generated. The grid of the cavity and the device in the wind tunnel where the cavity is embedded can be seen in Figure 13. The figure also displays an instantaneous Mach number on a cut at the center of the cavity.

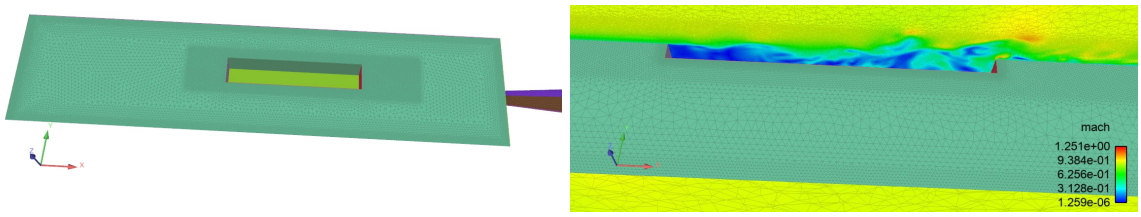

Figure 13 – Surface grid of the cavity inside the wind tunnel (left). Instantaneous Mach number on a cut in the cavity (right).

Comparative calculations are carried out for 10 time steps only, starting from a fully developed solution using the IDDES model [17]. The parallel computations use 32 cores and partitions. For the unsteady flow simulations, we use a time step of $\Delta t = 10^{-5} s$. In each physical time step, we perform a constant number of 200 dual iterations for the explicit/line-implicit scheme as well as for the fully implicit scheme. We have deliberately chosen a high number of sub iterations to demonstrate that we can push the convergence in dual time to very low values. We stress that 200 dual iterations is far beyond required engineering accuracy. The convergence in dual time can be seen in Figure 14. The density residual with the implicit scheme reaches slightly lower values when converged in dual time. As for the unsteady RANS calculation with Gripen above, the residual increases to a higher level with the implicit scheme when going from one time step to the next. The corresponding mid plot of the convergence of the maximum temperature reveals that this quantity converges more quickly with the implicit scheme at each time step. The right plot displays the maximum density residual after the 200 dual iterations versus CPU (elapsed) time during the 10 time steps. The gain in efficiency is about a factor of 2.5.

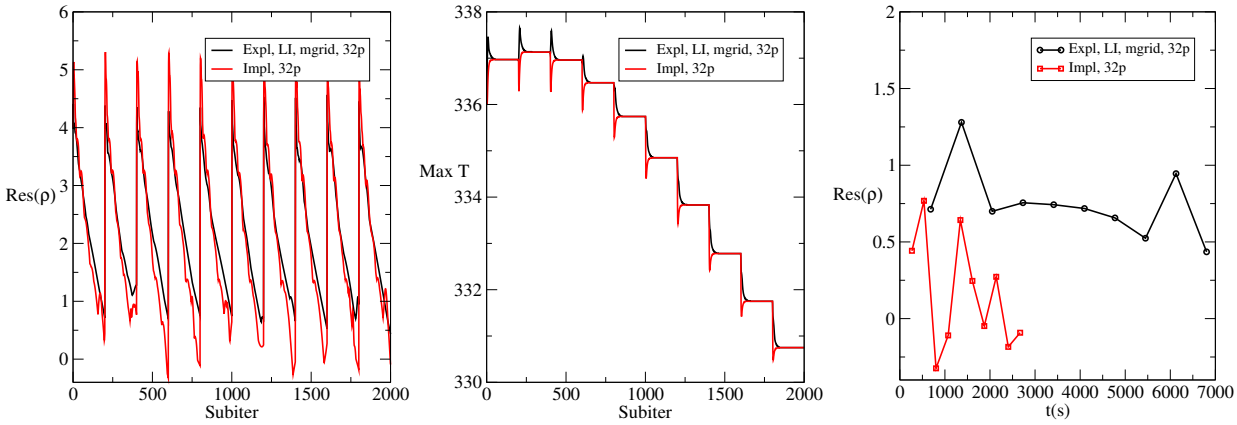

Figure 14 – Convergence in dual time for the M219 cavity. Maximum density residual convergence as function of sub iterations (left). Maximum temperature convergence as function of sub iterations (mid). Converged maximum density residual as function of elapsed time (right).

7. Summary and Conclusions

We present a numerical approach based on an implicit discretization of the governing Navier-Stokes equations. The implicit approach is employed for steady and time dependent problems. In the latter a dual time stepping algorithm is introduced where a steady state problem is solved in dual time for each physical time step.

The implicit approach relies on the discretization of the non-linear governing equations using an Euler backward method. This is an iterative procedure that is used to drive the residual vector towards zero to obtain a steady state. For each non-linear iteration, the residual is linearized leading to a linear system of equations to be inverted. An external library, namely PETSc, is used to solve the linear system iteratively with an ILU preconditioned a GMRES method to make the iterative solution efficient. The approximate solution of the linear system is used to update the flow solution. Beyond the description of the new algorithm, we also give some details on its implementation and usage that we have found to be important for the efficient and robust use of the scheme.

We present numerical results for a number of test cases in two and three space dimensions with comparative calculations using explicit multigrid calculations combined with line-implicit integration. There is a substantial speed-up in the steady state convergence for all cases. For the steady state Gripen calculation we manage to obtain a fully and machine converged steady state solution whereas the corresponding calculations with the explicit/line-implicit approach leave remaining oscillations in forces and moments. Also, unsteady calculations benefit in efficiency. All calculations show a speed-up of at least a factor of three. We also demonstrate that unsteady calculations over Gripen can end up in different time dependent solutions depending only on the initial solution. We have given an explanation to why this can happen. The price to pay for the new method is an increased computer memory requirement by about a factor of four.

8. Contact Author Email Address

Mailto: peter.eliasson2@saabgroup.com

9. Copyright Statement

References

[1] Arvidson, S., Carlsson, M. and Nilsson, S. Effect of LES length scale and numerical scheme in hybrid RANS-LES of free shear layer flows. *International Council of the Aeronautical Sciences, ICAS*, Stockholm, 2022.

[2] Balay, S., Gropp, W. D., McInnes, L. C. and Smith, B. F. Efficient management of parallelism in object oriented numerical software libraries, *Modern Software Tools in Scientific Computing*, Birkhäuser Press, pp. 163-202, 1997.

[3] Berglind, T. An agglomeration algorithm for Navier-Stokes grids. *AIAA Paper*, No. 2000-2254; 2000.

[4] Cook, P., Donald, M., Formin, M. Aerofoil Rae 2822 - Pressure distributions and boundary layer and wake measurements, *AGARD Report AR*, Vol. 138, pp. A6.1-A6.77, 1979.

[5] Eliasson P. EDGE, a Navier-Stokes solver for unstructured grids. FOI report, FOI-R-0298-SE, Swedish Defence Research Agency (FOI). Stockholm (Sweden), 2001.

[6] Eliasson, P., Eriksson, S. and Nordström, J. The influence of weak and strong solid wall boundary conditions on the convergence to steady-state of the Navier-Stokes equations. *AIAA Paper*, No. 2009-3551, 2009.

[7] Eliasson, P. and Weinerfelt, P. Recent applications of the flow solver Edge, Proc. to 7th Asian CFD Conference, Bangalore, India, 2007.

[8] Eliasson, P., Weinerfelt, P. and Nordström, J. Application of a line-implicit scheme on stretched unstructured grids. *AIAA Paper*, No. 2009-016, 2009.

[9] Eliasson, P. and Weinerfelt, P. High-order implicit time integration for unsteady turbulent flow simulations, *Computer & Fluids*, Vol. 112, pp. 35-49, 2015.

[10] Haase, W., Braza, M. and Revell, A. DESider - A European effort on hybrid RANS-LES modelling, *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, Vol. 103, Springer, 2009.

[11] Jameson, A. Time dependent calculations using multigrid, with applications to unsteady flow past airfoils and wings. *AIAA Paper*, 91-1596, 1991.

[12] Jameson, A. Evaluation of fully implicit Runge-Kutta schemes for unsteady flow calculations, *Journal of Scientific Computing*, DOI 10.1007/s10915-017-0476-x, 2017.

[13] Nordström, J., Eriksson, S. and Eliasson, P. Weak and strong wall boundary procedures and convergence to steady state of the Navier-Stokes equations, *Journal of Computational Physics*, Vol. 231, pp. 4867-4884, 2012.

[14] Roe, P. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, Vol. 43, No. 2, pp. 357-372, 1981.

[15] Saad, Y. and Schultz, M. H. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM Journal of Scientific and Statistical Computing*, Vol. 7, No. 3, pp. 856-869, 1986.

[16] Schmitt, V. and Charpin, F. Pressure distribution on the ONERA-M6-wing at transonic Mach numbers, *AGARD Report AR*, Vol. 138, pp. B1.1-B1.44, 1979.

[17] Shur, M. L., Spalart, P. R., Strelets, M. K., et al. A hybrid RANS-LES approach with delayed-DES and wall-modelled LES capabilities. *International Journal of Heat and Fluid Flow*, 29, pp. 1638-1649, 2008.

[18] Spalart P. R., Allmaras, S. R. A one-equation turbulence model for aerodynamic flows. *AIAA Paper*, No. 92-0439, 1992.