The 9th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2024
3–7 June 2024, Lisbon, Portugal

ON THE EXTENSION OF THE CFD SOLVER CODA FOR TURBOMACHINERY APPLICATIONS

Graham Ashcroft[1*], Alexander Bleh[1], Hakki Birliker[1] and Pascal Post[2]

[1] Institue of Propulsion Technology, German Aerospace Center (DLR), Linder Hoehe, 51147 Cologne, Germany, graham.ashcroft@dlr.de
[2] Airbus Operations GmbH, Airbus-Allee 1, 28199 Bremen, Germany, pascal.post@airbus.com

Keywords: *High Performance Computing, Boundary Conditions, Turbomachinery, Fluid Mechanics*

Abstract

The optimal integration of propulsion systems in the next generation of aircraft is a fascinating and challenging multidisciplinary design problem. To fully realise the combined potential of next generation aircraft and engine technologies an integrated multiphysics design approach is key. In this context the German Aerospace Center (DLR), the French Aerospace Lab (ONERA) and Airbus are currently developing CODA (**C**FD **O**NERA **D**LR **A**irbus), a next-generation CFD solver for aircraft and turbomachinery design, devised to be able to fully exploit current and future HPC architectures [1].

To efficiently simulate turbomachinery components a number of specific boundary conditions are necessary. In this paper we discuss the implementation of the so-called mixing-plane boundary condition in CODA. Due to the relative motion between adjacent blade rows in compressor or turbine stages, the flows within turbomachinery components are inherently unsteady. To nevertheless efficiently simulate such flows in the context of steady simulations, approximate artificial boundary conditions (mixing-planes), in combination with some form of non-reflecting boundary condition, are commonly used between blade rows [2]. The implementation of such mixing-plane boundary conditions (which are non-local in space) in an highly optimized HPC environment, such as CODA, is nontrivial. In this work we describe the mixing-plane boundary condition implementation in detail and outline the approach adopted to minimize its impact on code performance and scalability.

1 INTRODUCTION

Although aircraft fuel efficiency has improved significantly in recent years, the benefit to the environment has been offset by the growth in air traffic over the same period of time. Therefore, to meet the European Commission's ambitious goal of climate neutral aviation by 2050, disruptive technologies based on revolutionary engine concepts and improved aircraft technologies will be needed alongside the widespread adoption of sustainable aviation fuels and Hydrogen [3].

At present, several new engine technologies are under development by all major aircraft

engine manufacturers, including CFM's open rotor RISE concept, Rolls-Royce's UltraFan and MTU's Water Enhanced Turbofan. Although these technologies differ conceptually, they have in common the adoption of very-high bypass-ratio, large diameter ($\approx 3.5\,m$) fans. This represents a significant increase in comparison to today's modern aero-engines with diameters of approximately $2m$ and therefore presents new and significant challenges in the context of aircraft integration.

To meet these challenges, DLR, together with its partners ONERA and Airbus, initiated in 2012 [4] the development of CODA, a next-generation CFD solver for aircraft design, with extensions for the simulation of turbomachinery components, fully capable of exploiting current and future HPC architectures. To easily operate in the context of a multidisciplinary design setting, CODA has a Python control-layer and has been designed as a plugin to DLR's platform for multidisciplinary design analysis, FSDM [5]. To allow the generation of efficient machine code, whilst at the same time enabling a high level of abstraction, the code is written in C++11 and makes heavy use of templates to minimize run-time overhead. To optimize parallel efficiency, CODA has been explicitly designed to support the multiple levels of parallelism found in modern HPC architectures, i.e. node-to-node, multicore and SIMD (Single Instruction Multiple Data). In terms of numerics, CODA supports both the cell-centered Finite Volume and the Discontinuous Galerkin discretization methods in combination with implicit and explicit time-integration schemes.

In this work, we present the extensions implemented in CODA to facilitate the efficient simulation of turbomachinery components. The paper is organized as follows. Firstly, the extension of CODA to support arbitrary rotating frames of reference is presented. The next two sections then present the implementation of the periodic and mixing-plane boundary conditions. Finally, the combined methods are applied to simulate a transonic fan stage and comparisons with experimental data are presented.

2 TURBOMACHINERY EXTENSIONS

In compressors, turbines or fan stages the relative motion between adjacent blade rows ensures the flows in these components are inherently unsteady. Whilst it is possible today with modern HPC resources to numerically simulate the time-dependent interaction between blade rows, very good estimates of a machine's performance characteristics can be obtained using well established modelling assumptions. Firstly, considering an isolated rotating blade row, it is clearly advantageous to adopt a frame of reference attached to the rotating blade row as we can generally expect the time-mean flow to be stationary in such a system. Secondly, neglecting indexing effects and assuming identical blade geometries, the time-mean flow in all the passages of a given row will be identical, and it is therefore sufficient to compute the flow in a single passage with appropriate boundary conditions. Finally, although unsteady, interaction effects between blade rows can impact the flow in multistage turbomachines, these are generally second-order effects and it is therefore sufficient to model only the time-mean interaction between the blade rows.

2.1 ROTATING FRAME OF REFERENCE

When simulating turbomachinery components it is convenient to adopt a rotating frame of reference, stationary relative to the respective blade rows. To this end, CODA has been extended to support arbitrary axes of rotation. An axis of rotation is defined by the

combination of an origin of location $\bar{o} = [o_x, o_y, o_z]^T$ and a unit direction vector $\bar{d}$ (see Figure 1).

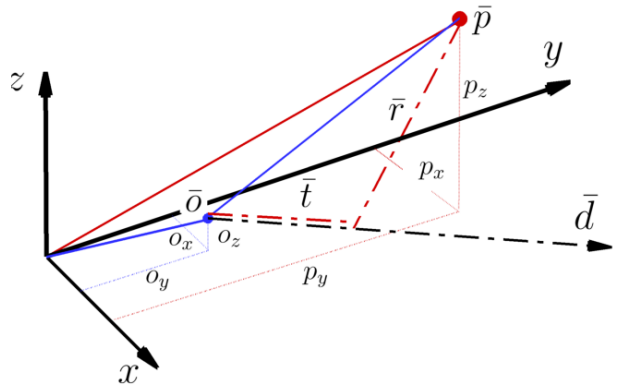


Figure 1: Axis of rotation and radius vector $\bar{r}$ of an arbitrary point $\bar{p} = [p_x, p_y, p_z]^T$.

The rate of rotation is specified separately by a scalar ω, with a positive value corresponding to a rotation in the direction given by the right-hand-rule applied to the unit direction vector $\bar{d}$. For such a system the relative velocity $\bar{u}$ at an arbitrary point $\bar{p}$ is

$$\begin{aligned} \bar{u} = \bar{v} - \bar{w} &= \bar{v} - \bar{\omega} \times \bar{r} \\ &= \bar{v} - \bar{\omega} \times (\bar{p} - \bar{o}) \end{aligned} \tag{1}$$

where $\bar{v}$ is the velocity in the absolute system, $\bar{\omega} = \omega \bar{d}$ and $\bar{r} = (\bar{p} - \bar{o}) - \bar{d}\left((\bar{p} - \bar{o}) \cdot \bar{d}\right)$ is the component of the position vector $\bar{p}$ perpendicular to the axis of rotation $\bar{d}$. The introduction of a (non-inertial) rotating frame of reference leads to the appearance of two fictitious forces in the governing equations: the *Coriolis* force and the *centrifugal* force. The Coriolis force is defined as $f_{cor} = -2\rho\left(\bar{\omega} \times \bar{u}\right)$ and the centrifugal force by $f_{cen} = -\rho\,\bar{\omega} \times (\bar{\omega} \times \bar{r})$ and appear as source terms in the momentum conservation equations. I.e. the Navier-Stokes equations, in the absence of external forces, in a rotating frame of reference take the form:

$$\frac{\partial}{\partial t}\int_\Omega \rho d\Omega + \oint_S \rho \bar{u} \cdot d\hat{S} = 0 \tag{2}$$

$$\frac{\partial}{\partial t}\int_\Omega \rho \bar{u} d\Omega + \oint_S \left(\rho \bar{u} \otimes \bar{u} + p\bar{\bar{I}} - \bar{\bar{\tau}}\right) \cdot d\hat{S} = -\int_\Omega \rho \left(2\left(\bar{\omega} \times \bar{u}\right) + \bar{\omega} \times (\bar{\omega} \times \bar{r})\right) d\Omega \tag{3}$$

$$\frac{\partial}{\partial t}\int_\Omega \rho E d\Omega + \oint_S \left(\rho \bar{u} I - k\bar{\nabla}T - \bar{\bar{\tau}} \cdot \bar{u}\right) \cdot d\bar{S} = 0 \tag{4}$$

where ρ is the fluid density, p is the fluid pressure, $\bar{\bar{I}}$ is the 3×3 unity matrix, $\bar{\bar{\tau}}$ is the viscous stress tensor, T is the fluid temperature, $E = e + \frac{1}{2}\left(\bar{u}^2 - (\bar{\omega} \times \bar{r})^2\right)$ is the relative total energy, $I = h + \frac{1}{2}\left(\bar{u}^2 - (\bar{\omega} \times \bar{r})^2\right)$ is the rothalpy, and $h = e + \frac{p}{\rho}$ is the enthalpy. Assuming a calorically perfect gas, the fluid pressure is related to the relative total energy by

$$p = (\gamma - 1)\left[\rho E - \frac{\rho}{2}\left(\bar{u}^2 - (\bar{\omega} \times \bar{r})^2\right)\right] \tag{5}$$

where $\gamma = \frac{c_p}{c_v}$ is the ratio of specific heats at constant pressure c_p and at constant volume c_v. The fluid temperature is determined from the ideal gas law, $p = \rho R T$. Alongside

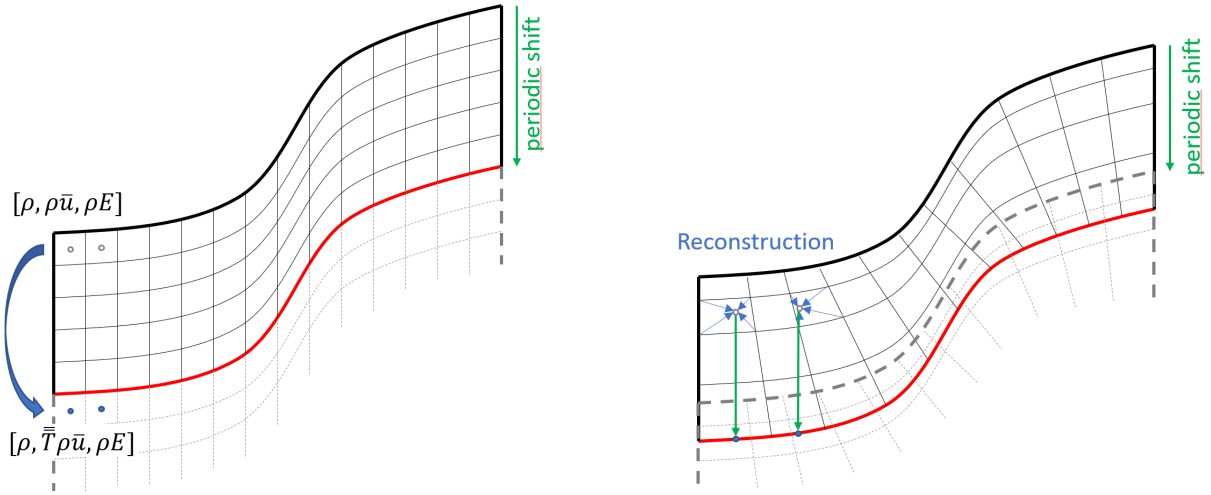
the source terms in the momentum equations, the rotating frame of reference leads to a modified form of the energy equation through...

Explain new terms in the energy equation!

2.2 PERIODIC BOUNDARY CONDITION

Periodic boundary conditions are a crucial solver acceleration technique for the simulation of turbomachinery components. They allow the consideration of only a single blade passage instead of the full annulus, hence roughly decreasing the computational cost by a factor equal to the number of blades of the current stage. For stationary simulations, in the context of the mixing-plane model, this comes without a loss in accuracy.

2.2.1 IMPLEMENTATION



(a) Matching one to one periodic boundary

(b) Overlapping periodic boundary

Figure 2: Different implementation concepts of periodic boundary conditions

Periodic boundaries enforce a condition, in which the periodic repetition of a domain merged with the original domain still yields a solution which fulfils the governing equations everywhere. The classic approach is to use a periodically matching mesh and use the opposite cells as neighboring cells of a one to one connection as shown in Figure 2a. CODA uses a more general formulation which does not require a periodically matching mesh and even allows the meshes to overlap, see Figure 2b. In this later case, the external states and gradients on the integration points of the periodic boundary faces are computed by reconstructing the corresponding states and gradients on the periodic shift of the integration point into the flow domain as shown in Figure 2. By integrating the functionality into the existing framework for *Overset* grids as described in [4], communication can be done together with all other boundary conditions in one step. While scalar quantities such as density or energy can simply be transferred without further care, vectors or tensors must be rotated by the periodic shift in the case of rotational periodicity. CODA features a generic engine which automatically identifies vectorial or tensor quantities at compile time and applies a rotation functor accordingly. E.g. in the case of a vectorial quantity,

the state components belonging to the vector are multiplied with a rotation matrix T, as shown in figure Figure 2a.

2.3 MIXING-PLANE BOUNDARY CONDITION

To facilitate the simulation of multi-row turbomachinery configurations artificial boundaries between the blade rows are introduced and the computational domain is then decomposed into a number of (non-overlapping) subdomains, one per blade row. The interfaces between the blade rows are typically located approximately midway between the adjacent blades. A (rotating) frame of reference can then be defined for each subdomain, in which the respective blade row is stationary. Finally, numerical boundary conditions are introduced at the interfaces to couple the adjacent subdomains. In unsteady simulations a boundary condition is required, that allows the conservative exchange of the instantaneous flow data between the adjacent subdomains. By contrast, if one is interested only in (an approximation of) the time-mean flow in each blade row, then it is sufficient to exchange only the circumferentially averaged flow variables between the adjacent blade rows using the so-called mixing-plane approach [6].

2.3.1 THEORY

The mixing-plane approach comprises two main components: (i) the definition and computation of average flow states, and (ii) the exchange of the average flow states between the neighboring subdomains. Whilst originally developed in the context of two-dimensional problems, the approach is easily extended to three-dimensions by the definition of radial bands, along which the two-dimensional theory can be independently applied. For each radial band it is possible, in general, to define several average states, e.g. mass-averaged, area-averaged or flux-averaged states. In turbomachinery applications, flux-averaged states are preferred since they ensure the conservation of mass, momentum and energy. However, as the implementation of flux-averaging is ongoing, in this work we employ simple, but robust, area-averaged states, i.e.

$$\bar{\phi}_A^F = \frac{\sum_i \phi_i^F A_i}{\sum_i A_i} \tag{6}$$

where the summation is across all cell faces in a given radial band, and ϕ^F denotes an arbitrary flow variable at face integration point, reconstructed from the local interior boundary element. Following this approach, area-averaged states are computed for both sides of the interface. In the converged state we require that both area-averaged states are consistent. To achieve this we formulate the differences between the up- and downstream state values in terms of characteristics variables and use one-dimensional non-reflecting boundary condition theory to update the state values, i.e.

$$\delta\bar{\phi}_A^F = \bar{\phi}_A^{F+} - \bar{\phi}_A^{F-} \tag{7}$$

where the superscripts $+$ and $-$ denote the up- and downstream sides of the mixing-plane interface, and $\delta\bar{\phi}_A^F = [\delta\bar{\rho},\ \delta\bar{u}_x,\ \delta\bar{u}_\theta,\ \delta\bar{u}_r,\ \delta\bar{p}]^T$. The band-average characteristic variables are given by the linear transform

$$\delta\bar{c} = M \cdot \delta\bar{\phi}_A^F \tag{8}$$

where $\delta\bar{c} = [\delta\bar{c}_1,\ \delta\bar{c}_2,\ \delta\bar{c}_3,\ \delta\bar{c}_4,\ \delta\bar{c}_5]^T$ and

$$M = \begin{bmatrix} \frac{-1}{\bar{\rho}} & 0 & 0 & 0 & \frac{1}{\bar{\rho}\bar{a}^2} \\ 0 & 0 & \frac{1}{\bar{a}} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\bar{a}} & 0 \\ 0 & \frac{1}{\bar{a}} & 0 & 0 & \frac{1}{\bar{\rho}\bar{a}^2} \\ 0 & \frac{-1}{\bar{a}} & 0 & 0 & \frac{1}{\bar{\rho}\bar{a}^2} \end{bmatrix} \tag{9}$$

Here, $\bar{\rho}$ and $\bar{a}$ are the local band-averaged density and speed of sound, respectively. In accordance with characteristic theory, the first mode c_1 corresponds to an entropy wave that travels with speed $\bar{u}_x$, the second and third modes c_2 and c_3 are vorticity waves which also travel with speed $\bar{u}_x$, and the fourth and fifth modes are acoustic waves which propagate with the speeds $\bar{u}_x + \bar{a}$ and $\bar{u}_x - \bar{a}$, respectively.

On the upstream side of the interface, only the upstream running acoustic mode ($\delta\bar{c}_5$) influences the face state. Similarly, on the downstream side of the interface, only the downstream running mean characteristics ($\delta\bar{c}_1, \ldots, \delta\bar{c}_4$) are used to update the face state. Following this approach the band-averaged state can be updated in an iterative manner as the simulation progresses, i.e.

$$\bar{\phi}_A^{F,n+1} = \bar{\phi}_A^{F,n} + \alpha M^{-1} \cdot \left[\delta\bar{c}^{\text{in}},\ \delta\bar{c}^{\text{out}}\right]^T \tag{10}$$

where α is a scalar relaxation parameter and the superscripts (in, out) denote incoming and out-going waves, respectively. The incoming waves are computed using Eqn. 8 and the out-going *mean* characteristics are set to zero.

In addition to the band-averaged states, the spatial variations (from the local side of the interface) in the flow state variables along each radial band need to be considered in the boundary treatment to avoid spurious numerical reflections. To accurately treat two-dimensional flows non-local boundary conditions are generally needed [2]. As a first step, however, we again use simple one-dimensional characteristic boundary conditions to demonstrate the basic mixing-plane functionality in this work. At each face the local perturbation in the primitive flow variables relative the band-average is computed and again used to define one-dimensional characteristic variables, i.e.

$$\delta c_i = M \cdot \delta\phi_i \tag{11}$$

where $\delta\phi_i = \phi(x_i) - \bar{\phi}_A$. To suppress reflections at each face the amplitudes of the local incoming characteristics δc_i^{in} are set to zero. The local out-going characteristics δc_i^{out} are computed according to Eqn. 11. The local flow state perturbation relative to the band-average value can then be updated, i.e.

$$\delta\phi_i^{n+1} = \delta\phi_i^n + M^{-1} \cdot \left[\delta c_i^{\text{in}},\ \delta c_i^{\text{out}}\right]^T \tag{12}$$

Finally, we combine Eqns. 10 and 12 to obtain updated values of the primitive flow state at each boundary face

$$\phi_i^{F,n+1} = \phi_i^{F,n} + M^{-1} \cdot \left[\alpha\delta\bar{c}^{\text{in}} + \delta c_i^{\text{in}},\ \alpha\delta\bar{c}^{\text{out}} + \delta c_i^{\text{out}}\right]^T \tag{13}$$

2.3.2 IMPLEMENTATION

Cylindrical Coordinates

We compute the area-averaged band states as the area-weighted summation across all integration points of every face associated with a given radial band using the primitive variables in cylindrical coordinates. To compute the band-averaged velocities the velocity vectors at each integration point are transformed to cylindrical coordinates using the orthonormal basis $(\bar{\eta}_d, \bar{\eta}_r, \bar{\eta}_\theta)$, where $\bar{\eta}_d = \frac{\bar{d}}{|\bar{d}|}$ is the unit vector parallel to the axis of rotation, $\bar{\eta}_r = \frac{\bar{r}}{|\bar{r}|}$ is the unit vector parallel to the radial vector $\bar{r}$, and $\bar{\eta}_\theta = \bar{\eta}_d \times \bar{\eta}_r$ is the unit vector in the direction of the radial band, such that

$$\begin{bmatrix} u_d \\ u_r \\ u_\theta \end{bmatrix} = \begin{bmatrix} \eta_{d_x} & \eta_{d_y} & \eta_{d_z} \\ \eta_{r_x} & \eta_{r_y} & \eta_{r_z} \\ \eta_{\theta_x} & \eta_{\theta_y} & \eta_{\theta_z} \end{bmatrix} \cdot \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \tag{14}$$

Radial Bands

To construct radial bands, the radial vectors $\bar{r}$, see Figure 1, of all element corner points $\bar{p}$ on the mixing-plane interface are first computed as

$$\bar{r} = (\bar{p} - \bar{o}) - \bar{d}\left((\bar{p} - \bar{o}) \cdot \bar{d}\right) \tag{15}$$

from which the radii $r = |\bar{r}|$ are determined. To construct the radial bands on each side of the mixing-plane interface an arbitrary element is first chosen and the radius of each corner point, r_i, computed. Using these data the first radial band is defined with lower and upper radial bounds given by $\min(r_i)$ and $\max(r_i)$, respectively. For the next element an average radius is computed from the average of the radii at each corner point and compared with the radial bounds of the first band. If the average radius lies within the radial bounds of the first band the element is added to the radial band. If not, a new radial band is created with lower and upper radial bounds determined from the values at each corner point of the element. By repeating this process for each interface element, radial bands are created across the whole interface. In Figure 3 a simple radial band is visualized in a computational domain comprising three domains. In this basic configuration a mixing-plane interface couples a rotating block row (red) with a downstream stationary block row (blue). For visualization purposes the radial band is displayed at the inflow of the computational domain. The downstream, stationary block row comprises two domains (or blocks).

In its current implementation the mixing-plane boundary condition requires the mesh to have an implicit band structure (see Figure 3). In future work, however, it will be extended to support arbitrary mesh topologies. As indicated in Figure 3, at the interface between domains I and II, the number of radial bands on each side of the mixing-plane interface do not need to be equal. To achieve this the (band-averaged) contributions from all radially overlapping partner bands are integrated radially using a simple area-weighted approach, i.e.

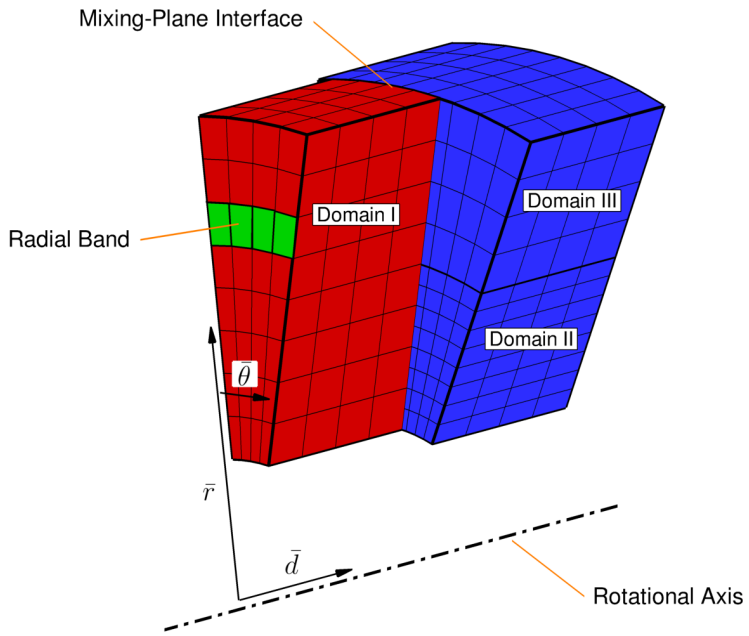


Figure 3: Mixing-Plane interface between three computational domains.

$$\bar{\phi}_A^{\pm}(r_i) = \sum_{j=1}^{N_{Bands}} w^{\mp}(r_j)\bar{\phi}_A^{\pm,donor}(r_j) \tag{16}$$

where the weights $w^{\mp}(r_j)$ of the down- and upstream sides of the interface are only non-zero if the donor bands overlap with the local (i-th) band. For the case in which the bands overlap, the area weight is given by

$$w(r_j) = \frac{r_{\max} - r_{\min}}{\Delta r_i} \tag{17}$$

where $r_{\max} = \min(r_{upper}^{own}, r_{upper}^{opp})$, $r_{\min} = \max(r_{lower}^{own}, r_{lower}^{opp})$, and $\Delta r_j = r_{upper}^{own} - r_{lower}^{own}$.

Exterior States

The mean characteristics $\delta\bar{c}$ are calculated from the differences between the up- and downstream transformed states as outlined earlier. The element local characteristics δc_i are computed as the difference between the transformed interior states of the local face and the band-averages. With these values, Equation 12 is used to update the individual face states. In a final step, the exterior boundary states are extrapolated from the local interior states using the updated face states, i.e.

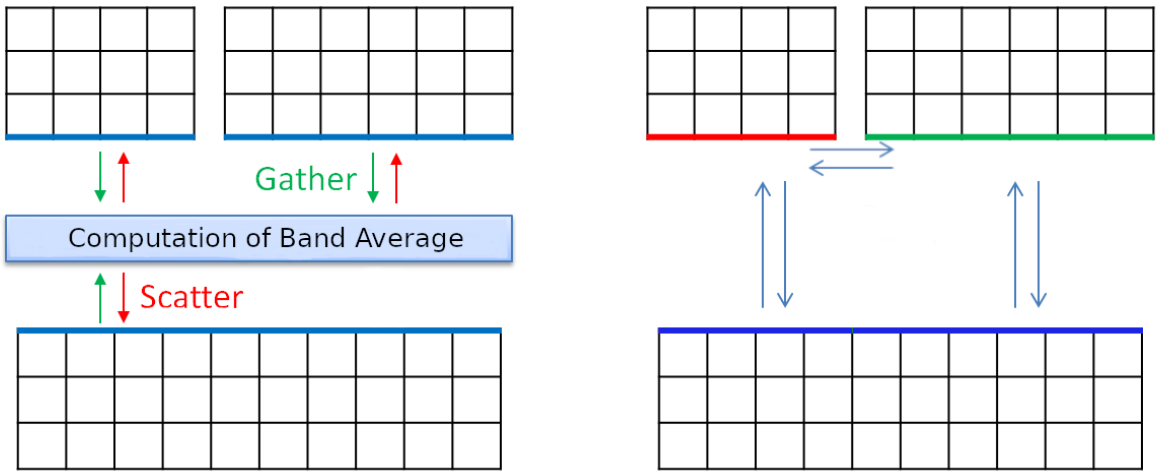
$$\phi_i^{ext} = 2\phi_i^F - \phi_i^{int} \tag{18}$$

where the superscripts *int* and *ext* denote the interior and exterior elements, respectively.

Parallel Communication

In simulations performed on multiple CPUs it is necessary to communicate the flow field data between processors. Within the context of a mixing-plane interface the data from

both sides of the interface must be communicated to all processors with boundary faces on the mixing-plane. As in general, it must be possible to split the domain in an arbitrary fashion to optimize load balancing, the number of processors with contributions to a given mixing-plane interface is essentially arbitrary. Various strategies exist to optimize parallel efficiency in large scale numerical simulations. Figure 4 highlights two possible strategies for computing and using band-averages in the context of a three block (processor) problem.



(a) Multistep communication (b) Singe-step communication

Figure 4: Comparison of communication patterns for the computation of band averages.

In the multistep approach sketched in Figure 4a all participating processors first communicate their data to a single worker processor that first gathers all necessary data for a given band, then (alone) computes the band-averages and applies the mixing-plane boundary condition, before finally communicating (scattering) the computed updated boundary values to all participating processors. This approach has the advantage that the work can be distributed among the participating processors. However, it requires numerous communication steps which can be bottleneck in large scale simulations. Therefore, in the context of CODA, the mixing-plane boundary condition is implemented such that only a single communication step is required, as sketched in Figure 4b. In this approach each participating processor communicates its data to all other participating processors. Following this step, each processor can then work independently of the other processors without the need for further communication. The disadvantage with this approach is that the computational work associated with the mixing-plane boundary condition is duplicated across the participating processors.

3 APPLICATION

3.1 FAN STAGE

To demonstrate the turbomachinery extensions described in the previous sections, CODA is applied to compute the NASA/GE Source Diagnostic Test (SDT) test case [7, 8]. The SDT test case is a high bypass ratio fan stage with a supersonic design tip speed tested by NASA in the Glenn Research Center 9- by 15-Foot Low Speed Wind Tunnel in Cleveland. The fan stage comprises 22 wide chord rotor blades and (in the baseline configuration, investigated here) 54 stator blades, with a rotor hub-to-tip ratio of 0.3 (see Figure 5).

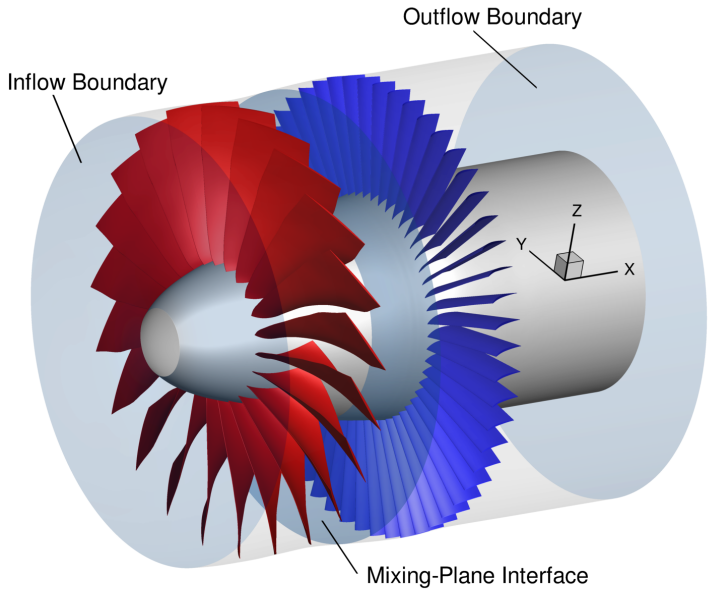
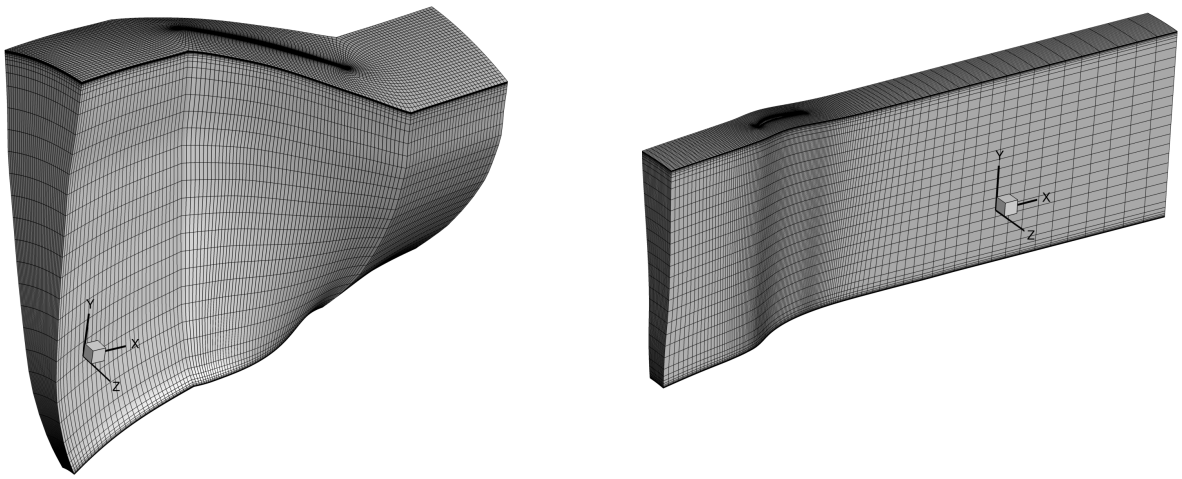


Figure 5: NASA SDT: Schematic representation showing the inflow and outflow boundaries and the mixing-plane interface.

In the experimental work a large collection of data were acquired. These include near- and far-field acoustic measurements, as well as performance data, velocity profiles and total pressure and total temperature radial profiles from duct rake measurements. In this initial study we present comparisons of the radial profiles of total pressure and temperature alongside the global performance data.

The computational mesh used in all numerical simulations is shown in Figure 6. The mesh comprises a total of approximately 720,000 cells, with circa 450,000 in the rotor domain and the remaining cells in the downstream stator domain. In the rotor domain the mesh has 60 cells in the radial direction with 12 cells located in the tip gap region. In the stator domain the mesh has 48 cells in the radial direction. All solid surfaces are modelled with a no-slip, adiabatic wall boundary condition, with the mesh on these surfaces designed to ensure $y^+ \approx 1$. The *relative* velocity of all solid surfaces $\bar{u}_w$ is zero in both domains, with the exception of the outer casing wall boundary in the rotor domain, which is stationary in the absolute frame of reference and therefore has the velocity $\bar{u}_w = -(\bar{\omega} \times \bar{r}_w)$ in the rotating frame of reference. The inflow boundary is located approximately one rotor chord length upstream of the rotor, whereas the outflow boundary is approximately five stator chord lengths downstream of the stator. The mixing-plane is located approximately midway between the rotor and stator. Importantly, the introduction of the mixing-plane boundary condition, with the exchange of only circumferentially averaged variables between the blade rows, ensures the time-mean flows in each blade passage are identical. It is therefore only necessary to compute the solution in a single passage of each row as indicated in Figure 6 rather than the full annulus as shown in Figure 5, thereby saving considerable computational resources.

To compute the flow through the fan stage the compressible, Reynolds-Averaged Navier-Stokes (RANS) equations, in combination with the Spalart-Allmaras (SA)-neg turbulence model [9], are solved using CODA. The governing equations are discretized in space using the Finite-Volume method together with the Green-Gauss method and the Venkatakrish-



(a) Rotor mesh. (b) Stator mesh.

Figure 6: NASA SDT: Mesh Overview.

nan limiter (with $K = 5$) to obtain second-order spatial accuracy. Numerical fluxes are computed using Roe's approximate Riemann solver. The fluid is modeled as a calorically perfect gas.

To solve the resulting system of ordinary differential equations a pseudo-time term $\frac{\partial q}{\partial \tau}$ is added to the governing equations and the linearized implicit Euler method, with local pseudo-time-stepping, is used to discretize the resulting system of equations, to obtain:

$$\left[\frac{1}{\Delta\tau} + \left.\frac{\partial R}{\partial q}\right|_{q^n}\right] \Delta q = -R(q^n) \tag{19}$$

with $\Delta q = q^{n+1} - q^n$. The solution to Equation 19 is obtained by iterating in pseudo-time using a specific number of iterations to drive $\Delta q \to 0$. In this work, at each pseudo-time iteration, Equation 19 is solved with a preconditioned, adjoint-based matrix-free (restartable) GMRES method using 25 iterations. It is implemented in the sparse linear system solver library Spliss which is used by CODA and allows for flexible configuration and combination of different solvers and preconditioners [10]. The GMRES solver is preconditioned using 50 Jacobi iterations applied to an approximation of the left hand side matrix. The number of Jacobi iterations in each application of the preconditioner is kept constant in order to ensure an effectively constant preconditioning matrix P. The inversion of the (generalized) diagonal is done using an implicit lines inversion method as described in [11]. Furthermore, an overrelaxation factor of 0.8 for the Jacobi solver proved beneficial for convergence of the given case.

Results are presented below for simulations performed at 61.7% fan speed, which corresponds to the *approach* aircraft flight operating point used for engine noise certification. For this speedline the rotational speed $N \approx 7809$ RPM. At the domain inflow boundary the total pressure and total temperature are specified as 101,325 Pa and 288.15 K, respectively. Additionally, the radial and circumferential flow angles are both zero, so that the flow is purely axial (in the absolute frame of reference). At the outflow boundary a value of static pressure is specified and varied between simulations to compute the flow along the speedline at difference massflow rates. Along both the inflow and outflow boundaries

a Riemann invariant based numerical boundary condition is used to suppress spurious numerical reflections and impose the physical boundary conditions.

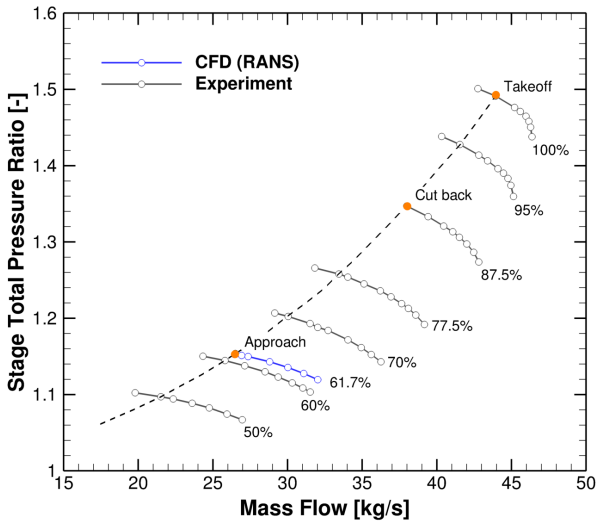


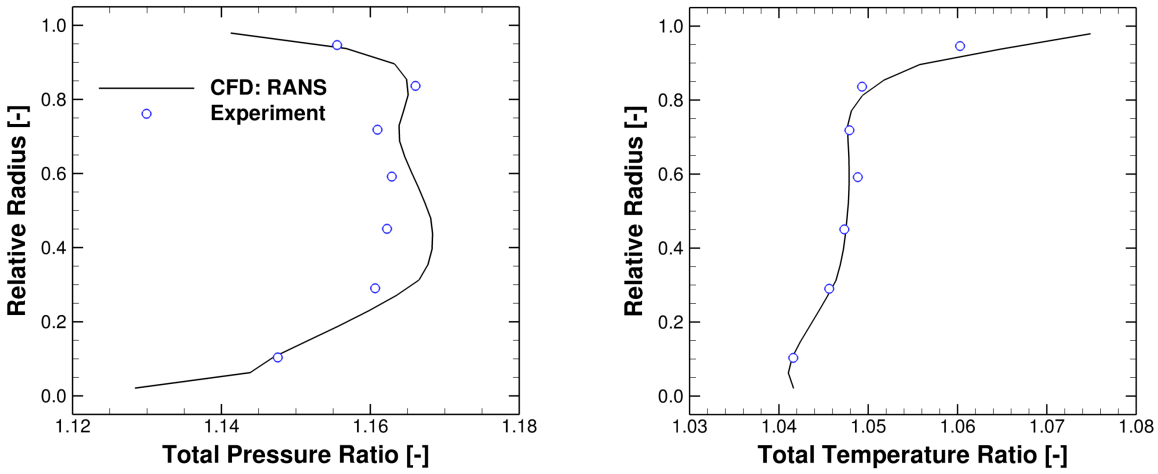
Figure 7: NASA SDT: Fan stage performance map.

The results of these simulations are shown in Figure 7 in terms of the overall performance characteristics of the fan stage. Here the computed speedline at 61.7% the nominal design speed is shown alongside the experimental data for a number of speedlines ranging from 50% to 100% nominal speed. In addition, the three operating points: approach, cut-back and takeoff, which are relevant for engine noise certification, are highlighted. The stage total pressure ratio is the ratio of the total pressure downstream of the stator vanes to the total pressure upstream of the rotor. Although experimental data are not available for the entire speedline at 61.7% design speed the trend of the curve matches well with the 60% speedline. In addition, the approach operating point is accurately predicted.

For a more detailed first analysis of the flow Figure 8 shows a comparison of the computed and measured radial profiles of the total pressure and total temperature ratios. The data are shown for the location $x = 0.15m$, where $x = 0$ is the location of the fan stacking axis, which is approximately one stator chord length upstream of the stator. In general the degree of agreement is very good and basic trends in both plots well captured, e.g. the basic influence of the hub and tip boundary layers in the total pressure profile (see Figure 8a), or the higher losses in the tip region due to the rotor tip vortex (see Figure 8b).

4 CONCLUSIONS

The CFD code CODA has been extended to facilitate the numerical simulation of basic turbomachinery components. To achieve this arbitrary rotating frames of reference, periodic boundary conditions and mixing-planes have all been implemented. Through comparison with experiment it has been shown that the implemented extensions are sufficient to accurately simulate a modern fan-stage. In ongoing and future work the mixing-plane will be extended to include two-dimensional non-reflecting boundary conditions, flux-averaged state variables and support meshes without any inherent band structure. It is also planned to extend the above methods to support the high-order spatial discretization



(a) Total Pressure Ratio.

(b) Total Temperature Ratio.

Figure 8: NASA SDT: Comparison of computed radial profiles with experimental data at the *approach* operating point.

methods available within the context of CODA.

ACKNOWLEDGEMENTS

The authors would like to thank the CODA core development team for their technical support, without which this work would not have been possible. CODA is the computational fluid dynamics (CFD) software being developed as part of a collaboration between the French Aerospace Lab ONERA, the German Aerospace Center (DLR), Airbus, and their European research partners. CODA is jointly owned by ONERA, DLR and Airbus.

REFERENCES

[1] S. Görtz, T. Leicht, V. Couaillier, M. Méheut, P. Larrieu and S. Champagneux, 2022, "CODA: A European Perspective for a Next-Generation CFD, Analysis and Design Platform," NATO AVT-366 Workshop on Use of Computational Fluid Dynamics and Analysis: Bridging the Gap Between Industry and Developers, 16–19 May 2022.

[2] M. Giles, 1991, "UNSFLO : A Numerical Method for Unsteady Inviscid Flow in Turbomachinery," Tech. Rep., Gas Turbine Laboratory Report GTL 205, MIT Dept. of Aero. and Astro.

[3] E. Van der Sman, B. Peerlings, J. Kos, R. Lieshout and T. Boonekamp, 2020, "Destination 2050 – A Route to Net Zero European Aviation," Netherlands Aerospace Centre, NLR-CR-2020-510, 2020.

[4] T. Leicht, D. Vollmer, J. Jägersküpper, A. Schwöppe, R. Hartmann, J. Fiedler and T. Schlauch, 2016, "DLR-Project Digital-X: Next generation CFD solver 'Flucs'," Deutscher Luft- und Raumfahrtkongress, 13–15 September 2016.

[5] M. Meinel and G. O. Einarsson, 2010, "The FlowSimulator framework for massively parallel CFD applications," PARA2010, Reykjavik, 6–9 June 2010.

[6] J. D. Denton, 1992, "The Calculation of Three Dimensional Viscous Flow Through Multistage Turbomachines," ASME Journal of Turbomachinery, 114 (1992), pp. 18–26.

[7] C. E. Hughes, 2022 "Aerodynamic Performance of Scale-Model Turbofan Outlet Guide Vanes Designed for Low Noise," 40th AIAA Aerospace Sciences Meeting & Exhibit, 14–17 January 2002.

[8] E. Envia, D. Tweedt, R. Woodward, D. Elliott, E. Fite, C. Hughes, G. Podboy and D. Sutliff, 2008, "An assessment of current fan noise prediction capability," 14th AIAA/CEAS Aeroacoustics Conference (29th AIAA Aeroacoustics Conference), 5–7 May 2008.

[9] S. R. Allmaras and F. T. Johnson, 2012, "Modifications and clarifications for the implementation of the Spalart-Allmaras turbulence model," Seventh international conference on computational fluid dynamics (ICCFD7) (Vol. 1902), July 2012.

[10] O. Krzikalla, A. Rempke, A. Bleh, M. Wagner, T. Gerhold, A. Dillmann, G. Heller, E. Krämer and C. Wagner, 2021, "Spliss: A Sparse Linear System Solver for Transparent Integration of Emerging HPC Technologies into CFD Solvers and Applications," New Results in Numerical and Experimental Fluid Mechanics XIII, 2021, pp. 635–645.

[11] A. Rempke, 2023, "Parallel line identification for line-implicit-solvers," BIT Numerical Mathematics, Vol. 63 (2023).