Unstructured anisotropic mesh adaptation for 3D RANS turbomachinery applications

Loïc Frazza,* Adrien Loseille† and Frédéric Alauzet‡
Gamma3 Team, Inria Saclay Ile-de-France, 91120 Palaiseau, France

The scope of this paper is to demonstrate the viability and efficiency of unstructured anisotropic mesh adaptation techniques to turbomachinery applications. The main difficulty in turbomachinery is the periodicity of the domain that must be taken into account in the solution mesh-adaptive process. The periodicity is strongly enforced in the flow solver using ghost cells to minimize the impact on the source code. For the mesh adaptation, the local remeshing is done in two steps. First, the inner domain is remeshed with frozen periodic frontiers, and, second, the periodic surfaces are remeshed after moving geometric entities from one side of the domain to the other. One of the main goal of this work is to demonstrate how mesh adaptation, thanks to its automation, is able to generate meshes that are extremely difficult to envision and almost impossible to generate manually. This study only considers feature-based error estimate based on the standard multi-scale Lp interpolation error estimate. We presents all the specific modifications that have been introduced in the adaptive process to deal with periodic simulations used for turbomachinery applications. The periodic mesh adaptation strategy is then tested and validated on the LS89 high pressure axial turbine vane and the NASA Rotor 37 test cases.

I. Introduction

In modern Reynolds-Averaged Navier-Stokes (RANS) numerical simulations, the mesh generation and CAD discretization is known to be one of the main bottleneck for many applications. In particular, the generation of suitable meshes for turbomachinery geometries that involve complex enclosed geometries and periodicity is a difficult task. Traditional processes rely on the experience and intuition of a skilled engineer to predict and adapt the mesh prescription to the flow. It is usually addressed by the mean of block structured meshes which generation require a careful management and is even more difficult when cooling holes or bleeding are included. Following such meshing guidelines slows down the mesh generation process leading to a prohibitive cost in CPU time in the numerical simulation pipeline. This lack of automation is an impediment for many applications. This is why we intend to develop mesh adaptation strategies for turbomachinery that automatically generate and adapt meshes to the complex geometry and flow features.

Mesh adaptation is a well known topic for external flows. Anisotropic mesh adaptation consists in modifying an initial non-adapted mesh in order to better capture physical phenomena. This relies on the estimation of the error in the computation of the solution due to the discretization. Once appropriately computed, this error estimation enables:

- Global error control and reduction
- Optimal and efficient computations leading to computational time reduction
- Functional output error control
- Automatic mesh generation according to the physical solution.

Although mesh adaptation is a well known topic for external flows, turbomachinery applications presents specific characteristics which makes it a challenging topic for mesh adaptation:

*PhD student, Gamma3 Team
†Researcher, Gamma3 Team
‡Researcher, Gamma3 Team
• Complex enclosed geometries (small gaps and holes, moving geometries),
• Complex physical interactions (rotating frames, shocks, boundary layers, ...),
• Periodic resolution.

In this paper, we propose to extend the mesh-adaptive solution platform to periodic domain and to analyze the results obtained for turbomachinery applications using feature-based error estimates. We developed an approach inspired by\textsuperscript{7} which one of the rare paper on local remeshing for periodic domain. The main difference between the proposed approach and\textsuperscript{7} is that we require to keep the initial periodic domain geometry while in\textsuperscript{7} the domain can change drastically. We add this constraint because it eases the development of other parts of the adaptive process (flow solver, error estimate, solution interpolation) and facilitates any solution analysis.

The periodicity is strongly enforced in the flow solver using ghost cells to minimize the impact on the source code. In that case, there is a little memory overhead but periodic vertices are computed similarly to inner vertices. This clearly facilitates the implementation and the impact of the periodicity in the code is localized in a few number of functions. The computation of the error estimate uses the same principle as the flow solver. As regards mesh adaptation, the local remeshing is done in two steps. First, the inner domain is remeshed with frozen periodic frontiers. Here, there is no change with respect to the classical process except for the periodic boundaries that are frozen and cannot be modified. Then, some geometric entities are moved from one side of the domain to the other in order to have the periodic boundary inside the domain. This step requires to transport the metric field which can be updated if periodic rotations are involved. The periodic boundary and its neighbourhood are then remeshed to be adapted. Once done, geometric entities are moved back to recover the initial domain geometry. As the domain geometry is preserved, there is no change in the solution interpolation stage, only a correction step is done at the end to be sure that the solution is strictly periodic.

The paper is outlined as follows. Section II presents the anisotropic mesh adaptation algorithm and the strategy to automatically perform mesh-convergence study. Section III recalls feature-based error estimates. Section IV quickly describes the flow solver and gives the considered implementation to take into accounts periodic domain. The strategy to remesh periodic domains and how metric field are handled is described in Section V. And finally, the benefits using metric-based anisotropic mesh adaptation for RANS turbomachinery simulations are pointed out on the LS89 high pressure axial turbine vane and on the NASA Rotor 37 cases in Section VI.

II. Anisotropic mesh adaptation algorithm with mesh-convergence analysis

Mesh adaptation is a non-linear problem where the couple formed by the mesh and the solution needs to be converged at the same time. Therefore an iterative process is required which is usually achieved by means of a mesh adaptation loop starting from an initial mesh/solution couple \((H_0, W_0)\), and an initial mesh complexity \(C_0\) (the continuous counterpart of the mesh size, see Section III).

At each step of the mesh adaptation loop, a metric tensor \(\mathcal{M}_i\) is computed from the couple \((H_i, W_i)\) and the given mesh complexity \(C_i\), using the selected error estimate, see Section III. It contains information on sizes and directions of the elements of the adapted mesh we seek. This information, given by the metric tensor field \(\mathcal{M}_i\), is then used by the remesher to generate a new adapted mesh \(H_{i+1}\).\textsuperscript{15} Then \(W_i\) is interpolated on \(H_{i+1}\) to obtain \((W^{09})_{i+1}\) which is then used as a restart solution for the next flow solution of the mesh adaptation loop.\textsuperscript{1} Restart solutions are important to not waste time in the adaptive process and reuse the previous done work. This iterative process is depicted by the step 1 while loop in Algorithm 1.

The convergence criteria of step 1(f) is up to the user, it specifies when the couple mesh-solution is considered as converge in the process. In this work, for turbomachinery applications, we consider that the solution is converged at the given complexity if the mass flow is not varying by a given percentage \(\epsilon\) on three consecutive iterations. Here, we choose \(\epsilon = 0.01\).

In the context of a mesh convergence analysis this adaptation loop has to be repeated for several increasing mesh complexities. An efficient strategy consists in converging the couple mesh/solution for a given complexity and reuse the final mesh and solution state to initialize the next computations at an increased mesh complexity. Such a process enables a multiscale resolution of the flow by solving large scale features on

\[ \text{2 of 20} \]

American Institute of Aeronautics and Astronautics
coarse adapted meshes (at the smallest complexities) and the fine scale features of the flow on fine adapted meshes (at the largest complexities). This acts like a "multigrid effect" and enables faster convergence on fine adapted meshes. At each outer loop iteration, the complexity is increase by a factor $\alpha$. In this work, we have set $\alpha = 2$. We have found that it is very advantageous to converge on the smallest complexities because of lot of work is done in converging the solution and these iterations are inexpensive.

III. RANS error estimates

In the mesh adaptation process, the metric field $(\mathcal{M}(x))_{x \in \Omega}$ used to prescribe the new adapted mesh $\mathcal{H}$ is automatically deduced from the actual solution with different error estimates. The goal is to find the optimal mesh $\mathcal{H}_{Opt}$ which minimizes the given error model $E$ for a fixed number of elements $C(\mathcal{H}_{Opt}) = N$:

$$\mathcal{H}_{Opt} = \arg \min_{C(\mathcal{H})=N} E(\mathcal{H}).$$

This problem can be analytically solved by recasting it in the continuous mesh theoretical framework: the goal is to find the optimal continuous mesh $\mathcal{M}_{Opt}$ which minimizes the given continuous error model $\mathcal{E}$ for a fixed continuous mesh complexity $C(\mathcal{M}) = N$:

$$\mathcal{M}_{Opt} = \arg \min_{C(\mathcal{M})=N} \mathcal{E}(\mathcal{M}).$$

The continuous mesh complexity is the continuous counterpart of the discrete mesh size (number of points or elements) and is used to prescribed the mesh size during the adaptation process. It is given by relation $C(\mathcal{M}) = \int_{\Omega} \sqrt{\det \mathcal{M}} \, d\Omega$.

There is a direct relationship between the prescribed metric complexity and the number of elements of the generated mesh. If a unit mesh $\mathcal{H} = \bigcup_i K_i$ (where the $K_i$ are the tetrahedra of mesh $\mathcal{H}$) is generated

Algorithm 1 General mesh adaptation algorithm with mesh-convergence analysis

Initial mesh $\mathcal{H}_0^0$, solution $W_0^0$, and complexity $C_0$

//--- Outer loop to perform the convergence study

while $C_j \leq C_{max}$ do

//--- Inner loop to converge the mesh adaptation at fixed complexity

1. while $i \leq n_{adap}$ do

   (a) Compute optimal metric for the considered error estimate and complexity $\Rightarrow \mathcal{M}_i^{j-1}$

   (b) Generate new adapted mesh $\Rightarrow \mathcal{H}_i^j$

   (c) Interpolate the solution on the new mesh $\Rightarrow (W^0)_i^j$

   (d) Compute the solution $\Rightarrow W_i^j$

   (e) if (convergence check) then

       $i = n_{adap} + 1$

   else

       $i = i + 1$

fi

done

2. $\mathcal{H}_{0+1}^j = \mathcal{H}_{n_{adap}}^j$; $W_{0+1}^j = W_{n_{adap}}^j$; $C_{j+1} = \alpha \cdot C_j$

done
with respect to \((M(x))_{x \in \Omega}\), then the continuous mesh complexity and the mesh size are linked by:

\[
\mathcal{C}(\mathcal{M}) \approx \sum_K \sqrt{\det M_K} |K| \approx \sum_K \frac{\sqrt{2}}{12} = \frac{\sqrt{2}}{12} \times nt ,
\]

where \(|K|\) is the volume of \(K\), \(M_K\) is the average metric at element \(K\), and \(nt\) is the number of elements of the mesh. In this work, we only consider feature-based error estimates based on a control of the interpolation error in \(L^p\)-norm.

**Feature-based mesh adaptation.** The most natural and straightforward approach is to control the interpolation error of a sensor field \(u = f(W)^{5,8,11}\) which is defined from solution \(W\). Given a continuous sensor \(u\), it is represented by its discrete nodal values on the mesh \(u_i = u(x_i)\) and its piecewise linear representation \(\Pi_h u\) on mesh \(\mathcal{H}\). The \(L^p\)-norm of the interpolation error of the sensor field \(u\) is stated as

\[
E(\mathcal{H}) = \left( \int_\Omega |u - \Pi_h u|^p \right)^{1/p}.
\]

Feature-based mesh adaptation enables to control the global interpolation error of the given sensor field. Under certain assumptions, we can prove that this approach also controls the approximation error.\(^{14}\) The analytical expression of the optimal continuous mesh \(\mathcal{M}_{Opt}\) in \(L^p\)-norm is given by:\(^2\)

\[
\mathcal{M}_{L^p}(x) = N^\frac{d}{2} \left( \int_\Omega \det|H_u(x)| \frac{p}{p+1} \, dx \right)^{-\frac{2}{p}} \det|H_u(x)|^{-\frac{1}{p+1}} |H_u(x)| ,
\]

where \(N\) is the complexity, \(d\) is the space dimension and \(H_u\) is the Hessian of the sensor \(u\) computed using a double \(L^2\)-projection method.\(^2,6\)

### IV. RANS flow solver

WOLF is a vertex-centered (flow variables are stored at vertices of the mesh) mixed Finite Volume - Finite Element Navier-Stokes solver on unstructured meshes composed of triangles in 2D and tetrahedra in 3D.

The convective terms are solved by the Finite Volume method on the dual mesh composed of median cells. It uses a HLLC approximate Riemann solver to compute the fluxes at the cell interfaces. Second order space accuracy is achieved through a piecewise linear interpolation based on the Monotonic Upwind Scheme for Conservation Law (MUSCL) procedure which uses a particular edge-based formulation with upwind elements. A specific low dissipation scheme is considered using combination of centered (edge gradient) and upwind gradients (element gradient). A dedicated slope limiter is employed to damp or eliminate spurious oscillations that may occur in the vicinity of discontinuities. The viscous terms are solved by the \(P^1\) Galerkin Finite Element Method (FEM) which provides second order accuracy.

The implicit temporal discretization considers the backward Euler time-integration scheme. At each time step, the linear system of equations is approximately solved using a Symmetric Gauss-Seidel (SGS) implicit solver and a local time stepping is considered to accelerate the convergence to steady state. A Newton method based on the SGS relaxation is very attractive because it uses an edge-based data structure which can be efficiently parallelized. From our experience, we have made the following - crucial - choices to solve the compressible Navier-Stokes equations.

- the SGS relaxation iterates until the residual of the linear system is reduced by one or two orders of magnitude
- the Breadth First Search renumbering proves to be the more effective for the convergence of the implicit method and the overall efficiency
- we found very advantageous to fully differentiate the HLLC approximate Riemann solver\(^4\) and the FEM viscous terms
- to achieve high efficiency, automation and robustness in the resolution of the non-linear system of algebraic equations to steady-state, it is mandatory to have a clever strategy to specify the CFL. The CFL evolution depends on the convergence of the linear system and on the evolution of the solution at each time step. A similar strategy has been developped in USM3D\(^{20}\) (NASA).
For the turbulence model, the Spalart-Allmaras is loosely-coupled to the mean-flow equations, where the mean-flow and turbulence model equations are relaxed in an alternating sequence. The flow solver WOLF is thoroughly detailed in\textsuperscript{2,18} with all the associated bibliography.

**Periodicity implementation**

In the case of periodic flows, we assume that the solution flow has a given periodicity (i.e. it is invariant by a given geometrical transformation $u(T(x)) = u(x)$) due to periodic physical properties. We will consider translation and rotation periodicitics. We can then restrain computation to (at least) a single period and emulate the rest of the domain with periodic boundary conditions. The domain is thus delimited by an arbitrary surface $\Gamma_{\text{per}}$ and its periodic mapping $\Gamma'_{\text{per}} = T(\Gamma_{\text{per}})$. In the periodic mesh, surfaces $\Gamma_{\text{per}}$ and $\Gamma'_{\text{per}}$ are meshed identically, thus there is a one-to-one mapping between their vertices and faces. As the periodicity is enforced in the discretization so that each vertex on $\Gamma_{\text{per}}$ has a linked vertex on $\Gamma'_{\text{per}}$, there is no solution interpolation on the boundaries.

To enforce strongly the periodicity, we use ghost or virtual entities (vertices and elements). Indeed, vertices on periodic boundaries have initially only a part of their geometric stencil (i.e., a part of their geometric entities ball) represented inside the mesh of the domain. The other part of the geometric stencil is on the other side of the domain around the linked vertices. As these vertices have to be treated like inner vertices, their geometric stencil are completed with ghost or virtual entities (vertices and elements) using the geometric stencil of their linked periodic vertices. Two choices in the implementation are possible:

- **virtual entities** are linked on existing entities on the other side of the domain using the linked vertex. This is memory efficient because no extra-entities are created, just a memory link is used. However, such an implementation impacts all the source code. Thus, each new implemented functionality should take care of the periodicity

- **ghost entities** are created inside the mesh. Then, there is a little memory overhead but in that case periodic vertices are computed similarly to inner vertices. This clearly facilitates the implementation and the impact of the periodicity on the source code, the periodicity is localized in a few number of functions.

In WOLF, we choose the second method. Therefore, the considered mesh (in red) - Figure 1 (left) - is completed at the beginning of the simulation with ghost entities (in yellow) as can be seen in Figure 1 (right).

**Time implicit resolution.** Theoretically, with an implicit resolution, inner nodes fluxes should depend directly on inner nodes values and ghost nodes should not play any role. Thus, the matrix should be corrected and ghost nodes not included. As we assume (for translation periodicity)

\[ u^\text{ghost}_i = u_i \]

we know that

\[ \frac{\partial R_j}{\partial u_i} = \frac{\partial R_j}{\partial u^\text{ghost}_i} \]

which allows to appropriately correct the matrix by simply copying the ghost coefficients in their paired nodes coefficients.

For sake of simplicity, we chose not to take into account directly the periodicity in the linear system. Periodic vertices are treated as Dirichlet points with an imposed variation of the solution. As for standard Dirichlet points their contribution to the Jacobian is fixed to identity, but instead of being fixed to 0, the right hand side is equal to the variation of the solution on the periodic vertex. The solution variation $\delta W$ is thus copied, at each SGS iteration, from the inner vertices to the ghost vertices through the right hand side. This weakens the coupling through the periodic frontier but still takes it into account.

The correction in the matrix is here taken into account iteratively, but it is strictly equivalent. Indeed, at convergence, we have

\[ \frac{\partial R_i}{\partial u_j} \delta u_j = -R_i, \]
so that for an inner point $P_i$ connected to a ghost point $P_j^{\text{ghost}}$ we have

$$\frac{\partial R_i}{\partial u_j^{\text{ghost}}} \delta u_j^{\text{ghost}} = -R_i.$$ 

As $\delta u_j^{\text{ghost}} = \delta u_j$ has been copied and converged, we have

$$\frac{\partial R_i}{\partial u_j^{\text{ghost}}} \delta u_j = -R_i,$$

and as by construction $\frac{\partial R_i}{\partial u_j^{\text{ghost}}} = \frac{\partial R_i}{\partial u_j}$, we actually verify the periodicity in the linear system.

V. Periodic mesh adaptation

As we choose to enforce strong geometric periodicity in the flow solver, it has to be enforced in the mesh at each adaptation. Theoretically it simply requires to simultaneously perform the local mesh operations on both sides of the domain. However, it would require several structural modifications in the mesh adaptation tool which is not the case right now. In a first attempt, we developed an approach inspired by\cite{7}. The idea is that the periodic frontier has been chosen arbitrarily and is not different from any other separation in the domain and could be adapted in the volume. To do so, we only need to immerse the frontier in the periodic domain, displacing a few layers of elements from one side of the domain to the other, as depicted in Figure 2 (right), where green elements have been moved. However, this requires a local remesher, here Feflo.a\cite{15} which is able to manage non-manifold surfaces. This process, contrary to\cite{7} guarantees the conservation of the geometry of the numerical domain, which simplifies latter operations like interpolation.
We provide in Figure 3 the schematic process of periodic mesh adaptation. We start from a periodic mesh in the vertical direction, lines in blue and green representing both sides of the periodic domain. Lines in red represent any other frontiers. Starting from the initial mesh, Figure 3 (a), we go through the elements and wrap the mesh on itself, replacing the vertices of one side of the domain by their periodic counterpart. This step affects all volume elements (thin black lines) and the non-periodic frontiers (red lines). This leads to a topologically periodic mesh with elements of negative volume, shown in Figure 3 (b). The vertices and periodic edges that have been disconnected from the rest of the domain, shown in blue in Figure 3 (b), are removed.

Vertices can be moved independently, keeping the topology of the mesh. We then simply translate a few
layers (any number) of vertices to their periodic location without changing the connectivity. In Figure 3 we move only one layer of vertices, those on the other side of the periodic domain, in green and obtain Figure 3 (c). The elements that have been displaced in this process are flagged in order to be identified later, in yellow in Figure 3 (d).

The vertices on the edges of the periodic domain are identified and duplicated in order to recreate their periodic counterpart. Elements are then reconnected to the proper vertices in order to recover a mesh with elements of positive volume, as shown in Figure 3 (d). The new periodic frontiers are identified and added to the final mesh, in pink and purple in Figure 3 (d).

The mesh can then be adapted, the old periodic line, in green, is treated as non-manifold and adapted while keeping the geometry of the two domains. The new periodic lines are left unchanged, guaranteeing the periodicity of the mesh. We thus obtain Figure 3 (e), where the periodic line in green has been adapted.

The new mesh could be used for the next computation step, however, as the domain has been changed.

![LS89-2D](image)

![RO37-3D](image)

Figure 4. Periodic mesh adaptation: domain translation and rotation on the LS89 blade and the NASA Rotor 37 cases. The grey elements are in the initial domain, the green elements have been displaced to their periodic location by the transformation.
(translated), for instance, it would make it difficult to interpolate solutions between to successives meshes. To circumvent this, we apply the same periodic transformation in a backward direction in order to recover the initial domain. We start by removing one periodic frontier and wrap the mesh on itself, translate all vertices in a flagged element to their periodic position and reconstruct the topology (boundary edges, duplicate vertices, ...). We finally obtain mesh (f) in Figure 3, where the periodic frontiers have been successfully adapted.

Practical cases are shown in Figure 4. Several layers of elements have been displaced in both cases, flagged in green. Displacing more than one layer of elements gives more space to the mesh adaptation tool Felfo.a$^{15}$ to perform mesh operations. This is helpful in presence of strong anisotropy such as in shocks or in boundary layers regions, as shown in Figure 5. We can see that in this case, the mesh adaptation tool is left with very little margin. Fortunately, these regions have been adapted in the volume previously, and the periodic frontier that we are interested in is immersed in the new domain.

We can see the adaptation of this frontier illustrated in Figure 6. The complexity being raised at this step, the size of the elements decreases globally. But as the periodic boundary is left unadapted during the initial step, we can see that we are left with elements that are too coarse in Figure 6 (left). Though, as the frontier is immersed in the new domain, it can be adapted as shown in Figure 6 (right).

![Figure 5. Elements displaced in presence of a boundary layer. Lateral view (left) and front view (right).](image5)

![Figure 6. Adaptation of the immersed periodic frontier. Mesh before (left) and after (right) adaptation.](image6)

**Handling the metric field**

In order to adapt the mesh in the transformed configuration, we need the corresponding metric. Hence, it has to be transformed consistently with the mesh, in particular, a unit edge in the initial configuration must remain unit in the transformed configuration. Obviously, the metric is the same in the part of the domain...
that is not displaced. For translation periodicity, the metric space is unchanged, so that copying the metric from the initial node to the translated node is sufficient. This is not the case for rotation periodicity. A rotation changes the directions of the metric, hence the metric has to be copied and rotated from the initial node to the translated node. Let us denote by $e$ a unit vector in the metric $\mathcal{M}$, representing an edge:

$$e\mathcal{M}e = 1.$$ 

The edge will be moved across the domain by the rotation and its orientation will change. We thus assimilate here the edge to a vector determining its orientation. This vector becomes in the rotation process

$$\bar{e} = Re,$$

where $R$ is the matrix of the rotation. As this new edge $\bar{e}$ has to remain unit in the new metric $\bar{\mathcal{M}}$, we deduce

$$1 = \bar{e}\bar{\mathcal{M}}\bar{e} = e(R^T\bar{\mathcal{M}}R)e.$$ 

Thus, we deduce

$$\bar{\mathcal{M}} = R\mathcal{M}R^T, \quad (2)$$

where we use that for a rotation $RR^T = Id$.

VI. Numerical results

In this numerical section, we consider turbomachinery applications. The first case is a well-know 2D case, the LS89 high pressure axial turbine vane, to emphasize which benefits can be expected in three-dimensions. The second case is the NASA Rotor 37.

A. LS89 blade

1. Cases description

We consider the LS89 high pressure axial turbine vane case. In order to illustrate the versatility of mesh adaptation we choose different outflow pressures (summed up here after in Table 1) so that a shock appears on the suction side in different places impacting the boundary layer and the wake (see Figure 7).

The computation starts from a non-adapted mesh composed of 31 790 vertices, refined around the blade, with no structured boundary layer (see Figure 10 (bottom left)). In order to have a quasi-structured mesh, the boundary layer is automatically adapted with the metric aligned method.\textsuperscript{16,17} The periodic domain is adapted using the method described in Section V, five layers of elements close to one side of the periodic domain are moved to the other side of the domain to perform the adaptation of the periodic surface.

Physical conservative fields are used as sensor to adapt, providing an optimal metric minimizing the interpolation error of each field, the final metric being the intersection of all. Note that no particular criterion (such as shock or vortex detection) has to be specified, the resulting mesh is the mesh minimizing

| Reference density | 1.275349 $kg.m^{-3}$ |
| Reference velocity | $\left(400, 0, 0\right) m.s^{-1}$ |
| Reference pressure | $10^5 Pa$ |
| Dynamic viscosity | $1.716 \times 10^{-5} Pa.s^{-1}$ |
| Total inflow pressure | $1.828 \times 10^5 Pa$ |
| Total inflow temperature | 413.3K |
| Static outflow pressure | $\{1.08 \times 10^5 Pa$, $1.04 \times 10^5 Pa$, $1.00 \times 10^5 Pa$, $0.90 \times 10^5 Pa\}$ |

Table 1. LS89 test cases description. Several flow prescriptions which depend on the static outflow pressure.
the interpolation error in $L^2$-norm of each field. This guarantees the convergence of the solution and the proper representation of any physical phenomenon of interest. Meanwhile, the use of the $L^2$ norm contrary to $L^\infty$ norm ensures a balance between strong and weak phenomenon. The wake of the blade is thus refined alongside shocks and boundary layers. This case already shows the versatility and efficiency of mesh adaptation for turbomachinery applications and in this process the importance of periodic adaptation. As the pressure difference increases, the initial subsonic flow becomes transonic and a shock appears on the suction side of the blade. This shock is initially weak and requires an appropriate fine discretization in the proper position to be computed. Then, this shock moves toward the trailing edge and becomes stronger. Similarly, the wake is subject to a strong diffusion if not discretized properly, although it is relatively easier to predict.

2. Results comments

We intentionally started with an inappropriate mesh shown in Figure 10, simply refined near the blade with no adaptation of the boundary layer, shock or wake. Figure 8 shows adapted meshes generated for each case while the corresponding solutions are shown in Figure 7. The meshes are automatically adapted to the wake and shock depending on their position with no a priori knowledge of the solution. Meanwhile, the boundary layer is progressively discretized, the height of the first cell decreasing from $y^+ = 20$ in the first meshes to $y^+ = 1$ in the final meshes. Details of the adapted mesh for case with $\alpha = 0.5689$ are shown in Figure 9. We can clearly see how the unstructured mesh becomes quasi structured in the boundary layer, shock and wake thanks to the metric-aligned method. These meshes are difficult to envision initially due to the numerous
Figure 8. LS89 cases. Adapted mesh generated for different pressure differences $P_{\text{out}} = \alpha P_{\text{tot}}$, $\alpha = 0.5908$ (top left, 124,610 vertices), 0.5689 (top right, 126,366 vertices), 0.5470 (bottom left, 125,965 vertices) and 0.4923 (bottom right, 127,149 vertices).

Figure 9. LS89 $\alpha = 0.5689$ case. Close up view on the adapted mesh in the shock and the boundary layer (left) and in the wake (right).
wakes crossing the domain and their interactions with the shock wave coming from the blade. This clearly illustrates the benefits of using a fully automated process to get the optimal mesh.

In Figure 10, we can see the influence of mesh adaptation on the quality of the solution of the second case ($\alpha = 0.5689$) with a moderate shock. The shock appears here only if it is appropriately discretized, while the initial boundary layer is too thick. Moreover the shock interacts with the boundary layer, thickening it, which does not appear on the initial mesh, the boundary layer is evenly thicker. These features influence both heat transfer at blade’s skin and turbine performances, and thus have to be appropriately computed.

Figure 11 demonstrates the importance of periodic mesh adaptation. When the periodic frontiers are appropriately adapted, we can see the shock crossing the frontier without being affected. This is what we expect from a numerical periodic frontier, to behave as an other domain. When the periodic frontiers are left unadapted, as the inner mesh is continuously refined, as fine as the initial mesh can be on the boundary, it inevitably ends to be too coarse compared to the required mesh size at a point of the adaptation. This leads to an inversion of the local anisotropy which deteriorates the quality of the solution and finally generates instabilities leading to unphysical results and finally the break down of the solver. Figure 12 shows a zoom on the density iso-contour on the other side of the periodic domain. It is clear that the shock has been diffused by the periodic boundary condition in absence of proper adaptation.

A similar constatation is done for the wake of the blade. Figure 13 shows coarse adapted meshes and a zoom on finer adapted meshes generated without (left) and with (right) periodic adaptation. We can see that periodic mesh adaptation yield very few differences on coarse meshes as the discretization of the periodic boundary condition is sufficient. But it is clear on finer meshes that its absence induces strong noise in the

![Figure 10. LS89 $\alpha = 0.5689$ case. Comparison between the initial mesh (right) and the adapted mesh (left) on the second case. Mach number isocontours (top) on the adapted adapted mesh (bottom left, 125 965 vertices) and unadapted initial mesh (right, 31 790 vertices).](image-url)
field and in turn in the process as it is captured by mesh adaptation. The influence of these meshes on
the solution is also depicted in Figure 14 where we can see the turbulence variable and the velocity being
abruptly diffused while crossing the periodic frontier.

Figure 11. LS89 $\alpha = 0.5689$ case. Influence of periodic adaptation on shock: adapted mesh (top left) and
Mach number isocountours (bottom left) without periodic adaptation and adapted mesh (top right) and Mach
number isocountours (bottom right) with periodic adaptation.

Figure 12. LS89 $\alpha = 0.5689$ case. Influence of periodic adaptation on the propagation of the shock through the
periodic frontier. Zoom on the density isocontour on the other side of the periodic frontier: without periodic
adaptation (left) and with periodic adaptation (right).
Figure 13. LS89 $\alpha = 0.5689$ case. Coarse (top) and fine (bottom) adapted meshes generated without (left) and with (right) periodic frontier adaptation. On coarse meshes, we didn't see a lot of difference between both cases but on fine meshes a lot of noise appears without a proper adaptation of the periodic frontier.

Figure 14. LS89 $\alpha = 0.5689$ case. Influence of periodic adaptation on the turbulence variable (top) and velocity norm (bottom), without (left) and with (right) periodic mesh adaptation.
B. NASA Rotor 37

The NASA Rotor 37 is a low aspect ratio compressor inlet stage 3D test case, which geometry is shown in Figure 2 (left). The regime considered is described in Table 2. The compressor being transonic, shocks appear, interacting with other blades through periodicity which requires an appropriate discretisation, unknown a priori and that depends on the flow regime. The bow shock formed in front of each blade induces a boundary layer detachment on the next blade, which modifies the mass flow. Similarly, the tip gap vortex that forms in the gap between the blade and the casing then interacts with the neighbouring blade and is responsible for instabilities in some cases. It is thus critical to accurately predict the behaviour of these flow features.

For numerical simulation, the flow is assumed to be periodic, so that the numerical domain can be restricted to a single blade in a $10^6$ sector depicted in Figure 2. In this work, we perform feature-based mesh adaptation, minimizing the interpolation error of the Mach field in $L^2$-norm. Five iterations are computed at each of the following complexities:

$$
\{200\,000, \ 400\,000, \ 800\,000, \ 1\,600\,000, \ 3\,200\,000\}.
$$

We compare adapted meshes and solutions with and without adapting the periodic frontier.

Figure 15 shows the initial periodic surface mesh that is kept constant without periodic mesh adaptation (top left), and the adapted periodic surface mesh (top right). The benefits on the solution accuracy is illustrated by showing the Mach number solution field computed on these meshes in Figure 15 (bottom). We can see that the rich shock structure and wake have been automatically adapted, improving their representation. It is worth noting that the geometry of the domain has been successfully preserved by the adaptive process. Figure 16 displays the result for a cut plane in the domain perpendicular to the blade. On the left picture, we observe noise in the mesh due to the non-adaptation of the periodic frontier while, on the right picture, the mesh is clean with adaptation of the periodic frontier. Similarly, the solution has more details with a proper adaptation of the periodic boundaries. Note that the non-adaptation of the periodic boundaries also creates artefacts in the solution.

Figure 17 gives a close up view of the adapted mesh where it crosses the periodic frontier. The miss-match between the mesh size required in the shock and the initial discretization on the frontier is clearly visible. This actually lead to rapid changes in the anisotropy of the elements that strongly degrades the precision of the solution and ultimately lead to the break down of the solver. A consistent strategy to adapt the periodic frontier is mandatory. Figures 18 and 19 illustrate again the pollution induced by the absence of adaptation of the periodic frontier (left) while the transition is seamless with a proper treatment (right). We can see that shocks are artificially diffused and reflected in the absence of periodic adaptation, even producing an artificial wake in the volume (see Figure 18 (left)).

| Inflow Total Pressure                        | $1.013 \times 10^5 \text{ Pa}$ |
| Inflow Total Temperature                      | 300.0 K                        |
| Outflow Pressure                              | $1.013 \times 10^5 \text{ Pa}$ |
| Rotating Speed                                | $1800 \text{ rad.s}^{-1}$     |
| Dynamic viscosity                             | $1.716 \times 10^{-5} \text{ Pa.s}^{-1}$ |

Table 2. NASA Rotor 37 test case description.
Figure 15. NASA Rotor 37 case. Comparison of the mesh (top) and the Mach number solution field (bottom) obtained with feature-based mesh adaptation without (left) and with (right) adaptation of the periodic frontier.

Figure 16. NASA Rotor 37 case. Global view of the adapted mesh (top) and the Mach number solution field (bottom) for a cut plane in the domain perpendicular to the blade without (left) and with (right) periodic mesh adaptation.
Figure 17. NASA Rotor 37 case. Closeup view on the shock crossing the periodic frontier, without (left) and with (right) periodic mesh adaptation.

Figure 18. NASA Rotor 37 case. Cut in the adapted mesh, without (left) and with (right) periodic mesh adaptation.

Figure 19. NASA Rotor 37 case. Close-up view of the cut in the adapted mesh, without (left) and with (right) periodic mesh adaptation.
VII. Conclusion and Future Works

In this paper, we have presented a mesh-adaptive solution strategy for periodic domain with application to turbomachinery. On the flow solver side, the periodicity is treated through ghost cells to minimize the impact on the source code. The main modifications reside in the implicit solver. On the local remesher side, the mesh is adapted in two steps. Starting from the initial periodic mesh, it is first adapted, keeping the periodic frontier unchanged to guarantee the periodicity of the domain. Displacing a few layers of elements from one side of the domain to the other, the two periodic frontiers are reduced to a single immersed surface. This immersed surface is then adapted and, finally, the elements that have been moved are then sent back to their initial position, which recreates the initial periodic surface. With such a strategy, the initial domain geometry is preserved which facilitates solution analysis.

The proposed method has been successfully applied to the LS89 blade and the NASA Rotor 37 cases using feature-based mesh adaptation based on a control of the interpolation error in $L^2$-norm. The numerical results clearly illustrate the benefits in terms of accuracy provided by the mesh adaptation, in particular to capture the shocks and the wakes. We have pointed out that the use of mesh adaptation without a proper adaptation of the periodic frontiers leads to noises in the adapted mesh and artefacts in the solution. It is mandatory to adapt appropriately periodic boundaries.

This study has validated the periodic mesh adaptation procedure but we have only considered the feature-based mesh adaptation strategy. Goal-oriented mesh adaptation strategy being known to be more efficient, we intend to assess its performance in a near future. This requires some modifications in the adjoint solver to properly deal, as the other tools, with periodicity, and to add specific cost functions using for turbomachinery applications. We will also investigate in more details the prediction of known flow features such as the tip gap vortex, mass flow and isentropic efficiency.

Acknowledgement

This work was partially supported by funding from Safran Tech with technical monitors Frédéric Feyel and Vincent Brunet.

References


