
Article:
Vivarelli, G., Qin, N., Shahpar, S. et al. (2020) Sequential feature-based mesh movement and adjoint error-based mesh refinement. International Journal for Numerical Methods in Fluids, 93 (1). pp. 249-272. ISSN: 0271-2091

https://doi.org/10.1002/fld.4882

Received: 1 February 2020 | Revised: 22 May 2020 | Accepted: 10 June 2020

DOI: 10.1002/fld.4882

RESEARCH ARTICLE

WILEY

Sequential feature-based mesh movement and adjoint error-based mesh refinement

Guglielmo Vivarelli[1] | **Ning Qin**[1] | **Shahrokh Shahpar**[2] | **David Radford**[3]

[1]Department of Mechanical Engineering, University of Sheffield, Sheffield, UK

[2]Future Methods, Rolls-Royce plc, Derby, UK

[3]Civil Aerospace, Rolls-Royce plc, Derby, UK

Correspondence
Guglielmo Vivarelli, Department of Mechanical Engineering, University of Sheffield, D01 Ella Armitage Building, 40 Leavygreave Rd, Sheffield S3 7RD, UK.
Email: guglielmovivarelli@googlemail.com

Funding information
Rolls-Royce, Grant/Award Number: 5002531676

Summary

Nowadays, aerodynamic computational modeling is carried out on a daily basis in an industrial setting. This is done with the aim of predicting the performance and flow characteristics of new components. However, limited resources in terms of time and hardware force the engineer to employ relatively coarse computational grids, thus achieving results with variable degree of inaccuracy. In this article, a novel combination of feature and adjoint-based mesh adaptation methods is investigated and applied to typical three-dimensional turbomachinery cases, such as compressor and fan blades. The proposed process starts by employing feature-based mesh movement to improve the global flow solution and then adjoint refinement to tune the mesh for each quantity of interest. Comparison of this process with one utilizing only the adjoint refinement procedure shows significant benefits in terms of accuracy of the performance quantity.

KEYWORDS

adjoint error, feature-based adaptation, mesh movement, mesh refinement, *Riemannian* metric, turbomachinery

1 | INTRODUCTION

Computational fluid dynamics (*CFD*), has widespread applications in many industrial settings. In fact, many simulations are carried out on a daily basis with the aim of determining flow behavior around assorted components, from jet engine compressors, to cooling fans present in computers. However, despite the technological advances allowing *CFD* to be employed in the engineering design process (see Reference 1 for examples), significant issues are still present. Firstly, mesh generation is manual, requiring experienced personnel to spend several hours attempting to improve grid quality, while allowing sufficient clustering where they estimate more convoluted flow behavior. Moreover, as discussed by Reference 2, the geometry complexity has increased, causing smaller flow features to appear: these too will have to be appropriately captured by the final solution.

Another issue present in *CFD* simulations concerns their accuracy. In fact, the estimated flow will contain a degree of error, that is generally unavoidable. Therefore, increasingly finer meshes are needed to carry out mesh independence studies to validate the solution. This is a tedious, expensive process, and as discussed by Reference 3 mesh-independent solutions obtained with alternative starting grids, may have a non-negligible variation in results.

According to References 4,5, the sources of inaccuracies in a *CFD* solution can be divided into *discretization error*, *round-off approximations*, *iterative* and *statistical sampling*, while Reference 6 also includes *modeling inaccuracies*, *input*

uncertainties and *post processing errors*. Between these potential causes, the most problematic one is discretization error, as also confirmed by References 4,5. It is this kind of inaccuracy that is targeted by mesh adaptation: this process consists of the automatic and efficient modification of a grid with the aim of reducing its effect, without having to resort to several uniform refinement steps.[7] Moreover, as discussed by Reference 8, it will aid improve the mesh generation turn-around time, as very limited flow behavior knowledge is required, since refinement and clustering towards high flow and geometrical complexity regions will be dealt automatically by the adaptation algorithm.

Mesh adaptation has been developed as a separate research topic from the early 1980s with the first specialized conference taking place in 1982.[7] Despite the significant effort to develop a variety of techniques over the last forty years or so, it has never found widespread use either in academia or industry. One of the main reasons behind this is the limited success that several approaches have had: many publications deal with *two-dimensional* (*2D*) cases, with *three-dimensional* (*3D*) geometries mainly concerning open-flows with a limited amount of flow features appearing (to this end simplifications such as inviscid or incompressible conditions are not uncommon).

Within the wide variety of mesh adaptation techniques, some of the most popular ones fall into the *feature*- and *adjoint*-based categories. In the former case, the error estimation is carried out computing variations of physical quantities that allow to determine where complex flow features may appear. Therefore, simple difference of values, gradients and second-order derivatives have been successfully employed.[7] Nevertheless, one of the most important aspects regarding feature-based adaptation concerns *anisotropy*. In fact, it is well known that most flow complexities present a much greater variation across, rather than in other directions. Therefore, being able to refine or cluster only in the normal direction would allow to improve accuracy minimizing the grid's node count. Moreover, as stated by Reference 9, by achieving grid alignment with shocks, the *Rankine-Hugoniot* relationship is better satisfied, thus improving the solution. According to Reference 10, the first instances of anisotropic mesh adaptation occurred towards the end of the 1980s in Reference 11. In this case, *2D* grids were regenerated including information concerning the stretching and orientation of elements. A *3D* version of the proposed approach appeared shortly after in References 12,13. However, the most important advancement in terms of feature-based anisotropic adaptation was described in Reference 14. The authors employed second-order derivatives (*Hessian* matrix) of a scalar quantity able to determine a *Riemannian* metric including information concerning stretching and orientation of grid edges. Shortly after, Reference 15 was able to exploit this error definition in a spring stiffness-based mesh movement algorithm to cluster and align the grid appropriately to improve flow resolution. This technique was then employed by References 16-20 in a repeated manner coupled with edge split, swap and collapse.

More recently, mesh regeneration based on the *Hessian* of flow quantities has started to gain more and more popularity. One of the two approaches proposed is that of References 9,21,22. In this case, features are identified by combining flow parameters and then fully quadrilateral/hexahedral high-quality blocks are placed around them. The remainder of the domain is then filled with hybrid elements. Despite showing improvement in feature resolution, it can present issues with elaborate geometries, particularly when multiple flow complexities interact. Therefore, to increase flexibility and capability of handling more intricate cases, a second approach regenerating the entire grid with tetrahedral elements has gained particular interest. By employing the *Hessian* matrix of flow variables, size, stretching, and orientation of each cell is determined. To this end, Reference 23 were able to compute an error-upper bound independent of the problem at hand. Shortly after, in References 24-26 they managed to achieve a continuous representation of the error field by employing an appropriate interpolation operator. Finally, in Reference 27 they showed how to optimize the cell distribution based on their previous developments.

As mentioned, another group of relatively successful adaptation techniques is based on adjoint quantities. These started to appear towards the mid-1990s in References 28,29 and are based on the fact that the adjoint vector gives the functional sensitivity to the flow residuals. Unlike feature-based adaptation, employing the adjoint error estimation has repeatedly proven to be more reliable. In fact, the former techniques will deal with complex regions of the flow, regardless of whether these require further refinement/clustering, and do not necessarily target where the error may originate from Reference 30. In particular, one of the main reasons in favor of adjoint-adaptation is its capability to determine the effect that local errors may have on the global quantity of interest, whilst neglecting any other sources of inaccuracies. The first few adjoint error estimation approaches were limited to relatively simple cases, such as *one-dimensional* (*1D*) flow or the inviscid *NACA* 0012 aerofoil. Shortly after, References 31-33 devised a multilevel error estimator approach that is probably the most popular and successful method to date. This requires the determination of linearly and quadratically interpolated flow and adjoint solutions to an embedded mesh. While this approach successfully managed to reduce the error in *1D* and inviscid *2D* cases, when adapting viscous flows, anisotropy was included by exploiting the *Riemannian* metric previously mentioned. The combination of adjoint error and anisotropic metric was achieved by scaling the

eigenvalues of the *Hessian* matrix by the adjoint error. The resulting quantity was then applied to a triangular mesh regeneration algorithm. Within the same publication, significant advantages *with respect to* (*w.r.t.*) feature-based techniques were shown. Inviscid *3D* applications of this approach were first published during the mid-2000s by Reference 34, with viscous flows being considered in Reference 35. In both cases, the technique showed improvements of the quantities of interest, although in the second publication, issues with high-order interpolation of turbulent quantities were highlighted. Effectively, the high-order prolongation to the embedded grid of flow and adjoint quantities was the focus of the studies of References 34,36. In both cases, it was concluded that more accurate interpolation operators would allow a sharper error estimation, whilst lower-order procedures would produce a more conservative error bound. Another issue that has been found to appear in this process is noise in the embedded mesh sensor fields, as may be seen in Reference 37. To minimise its effect, Reference 38 showed how control-volume weighted repeated smoothing of the flow residuals aided the error smoothness. Unfortunately, this seemed to be limited to inviscid flows, as in a later publication[39] they reported how, for viscous cases, smoothing immediately distorted the sensor field.

Despite the significant research on the methods initially devised by Reference 31, there are other mesh adaptation techniques developed around the adjoint solution. One of the most original ones is that of References 40,41, in which it was argued that up to 90% of the error in a flow solution is due to *artificial dissipation*. In particular, the analysis was based on the *Jameson-Schmidt-Turkel* (*JST*)[42] flux discretization scheme, and the sensitivity of the quantity of interest *w.r.t.* the *JST* coefficient was determined and employed to ascertain an error sensor. Despite successfully managing to improve the quantities of interest and avoid the need for an embedded grid, it was limited to only one type of flux reconstruction scheme.

A more generic approach that does not require an adjoint solution is that of Reference 43. In this case, entropy variables were employed and it was shown that they do satisfy an auxiliary adjoint relation. These could then aid in reducing artificial dissipation. Despite showing promising results for subsonic flows, issues appeared with shocks, as entropy is no longer conserved.

A final method that ought to be considered, is that of References 44-47. In this case, they employed the mesh adjoint functional sensitivity to the grid coordinates to be able to adapt the mesh.

As shown within the literature, there is a significant amount of research indicating that adjoint-based mesh adaptation technology is superior *w.r.t.* feature approaches. However, it has never really been remarked, that they are much more costly, as they require an additional adjoint simulation to be carried out, thus roughly doubling the turnaround time. In the particular case of discrete adjoint solvers coded with the aid of *automatic differentiation* packages, the flow *Jacobian* matrix required to solve the adjoint relations may have to be recomputed for each node every iteration,[48] further increasing simulation time. Moreover, this type of grid modification will only be able to adapt a grid to optimize the accuracy of a single functional at a time, thus for n quantities of interest, n-adjoint adaptation processes ought to be run. As the reader may have readily noticed, this can be very costly in terms of time and hardware, particularly if varied flow conditions have to be analysed, as the entire process will need to be repeated for each one. On the other hand, despite being less effective, feature-based adaptation aims to improve the entire flow field and thus a single process may be employed to improve all quantities of interest. Therefore, devising a procedure that would utilise feature-based approaches to start with, followed by adjoint-error estimation can be helpful in achieving a more efficient method. Moreover, the latter are better suited in capturing flow anisotropy. As previously mentioned, a simultaneous combination of the two technologies has already been carried out, however, this will not avoid the need to split processes for every functional, and will not take advantage of the initial reduction of the adjoint-error that feature-based approaches can provide.

In this work, to achieve a robust and efficient technique, the first part of the adaptation will employ *Hessian*-based mesh movement, that will be iteratively repeated till the feature-based error converges. On achieving the aligned grid, adjoint adaptation (refinement) exploiting the error formulation devised by Reference 49 will be applied to minimize the errors for each functional in separate processes. The resulting adapted mesh accuracy will be compared with an adjoint refinement applied to the starting mesh (i.e., no feature-based mesh movement). It will be shown that the use of the *Hessian*-based mesh movement compares favorably by providing an increased accuracy.

As turbomachinery components are considered in this work, it is also necessary to briefly review the mesh adaptation literature considering these. Despite the developments of feature-based error estimation, only[18] and more recently[27] attempted to automatically modify grids for compressors. In the first case, the authors employed the process discussed earlier on, i.e. move, swap, split and collapse of edges based on a *Riemannian* metric. On the other hand, in Reference 27, again exploiting a *Riemannian* metric, part of the domain's mesh was regenerated, except for the prismatic boundary layer and the periodic surfaces. In both cases, other than improving flow complexities resolution, no significant benefit was shown concerning quantities of interest. As for the feature-based methodologies, limited examples of the application

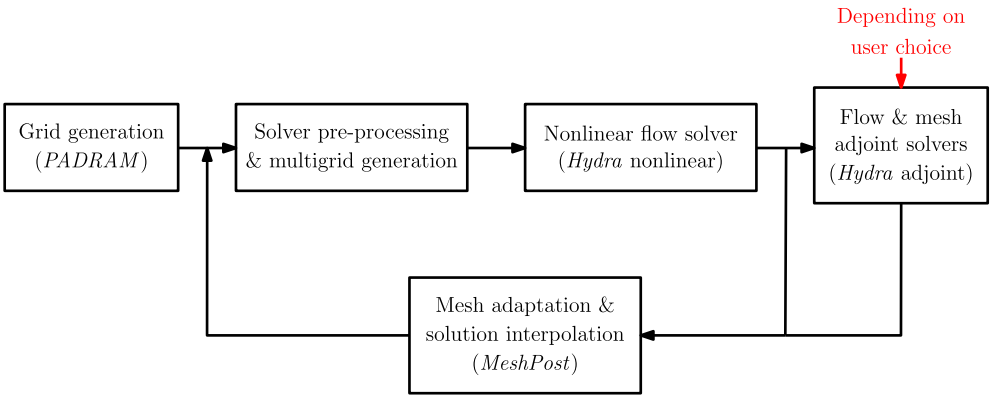


FIGURE 1 Overall *CFD* software system employed[59] [Color figure can be viewed at wileyonlinelibrary.com]

of adjoint-error estimation to turbomachinery cases are available in the literature. For example, Reference 50 considered a *2D* laminar flow through a compressor cascade, with the performance quantities of interest being total pressure loss and entropy generation. More recently in Reference 51, an attempt was made to determine an adjoint-error sensor on a turbine stator, but the authors did not show any consequent grid modification. Other publications successfully applying feature and adjoint mesh adaptation to turbomachinery cases are References 52,53, which also form the basis for this work. Therefore, it may be concluded, that turbomachinery components have received very limited attention by the mesh adaptation community and that till now, very limited success has been achieved in improving their flow solution accuracy.

In the following sections, a brief description of the mesh generation process along with flow and adjoint solvers will be provided (Section 2). In subsequent parts, a more in-depth analysis of the feature- and adjoint-based adaptation schemes employed and how they have been combined is discussed (Sections 3.1-3.3, respectively). Prior presenting the results, an overview of the compressor and fan cases analysed is included (Section 4). Next, a detailed description of results for each test-case along with the benefits that each separate method provides are reported (Section 5). Prior discussing the conclusions, a brief summary of the overall findings is provided (Section 6). Finally, concluding remarks and suggestions for future developments are provided (Section 7).

2 | SOFTWARE

The system employed for this work consists of the *Rolls-Royce* in-house mesh generator (*PADRAM*[54]), flow/adjoint solver (*Hydra*[55,56]) and mesh adaptation software (*MeshPost*[57-59]). The overall process is summarized in Figure 1, with a more detailed description of each component provided in the following sections.

2.1 | Mesh generation

The grids utilized for this work were all generated by means of *PADRAM* (*PArametric Design and RApid Meshing for complex turbomachinery configurations*[54]). The philosophy behind the software's development has been to produce a library of templates to represent the typical components that compose a jet engine (e.g. nacelles, fans, compressors and turbines). These are then meshed employing a *structured multi-block* data structure allowing the generation of good quality fully *hexahedral* grids. It should be clear that, while the software is limited to geometrical templates included within its source code, it does minimize the user interaction during the tedious and time-consuming grid generation process.

2.2 | Flow solver

Hydra[55,56] is an *unstructured* second-order *finite-volume* flow solver, employing a *median-dual* spatial discretization (thus belonging to the *vertex-centred* class). Due to the latter characteristic, it can make use of *edge*-based data structures to

efficiently solve the equations. As this uses the *equation of state* to relate density and pressure, it falls within the *density*-based flow solver category. The *Reynolds-Averaged-Navier-Stokes* (*RANS*) equations are closed employing the *Spalart-Allmaras* (*SA*) turbulence model[60] as the adjoint solver has a consistent and robust discretization of it.

The *Navier-Stokes* (*NS*) discretization at node i can be formulated as Reference 56:

$$\mathbf{R}_i = \frac{1}{V_i}\left(\sum_{j=1}^{\mathrm{E}_i}(\mathbf{F}_{ij}^I + \mathbf{F}_{ij}^V)\Delta\mathbf{s}_{ij} - \mathbf{S}_i V_i\right), \tag{1}$$

where $\mathbf{R}$ are the residuals, $\mathbf{F}^I$ and $\mathbf{F}^V$ represent the inviscid and viscous flux contributions, $\Delta\mathbf{s}_{ij}$ is the control surface between neighboring nodes i and j, $\mathbf{S}_i$ represents the source terms and V is the control volume. The summation is carried out over all the set of edges (E_i) connected to node i. The inviscid flux reconstruction at control volume boundaries between nodes i and j is carried out by means of a modified centered scheme, that is, *JST* with *matrix dissipation*[61,62]:

$$\mathbf{F}_{\mathrm{ij}}^I = \frac{1}{2}\{\underbrace{\mathbf{F}^I(\mathbf{Q}_i) + \mathbf{F}^I(\mathbf{Q}_j)}_{\text{Centred formulation}} - \overbrace{|\,A_{\mathrm{ij}}\,|\,[\Phi(\underbrace{\mathbf{Q}_i - \mathbf{Q}_j}_{\substack{\text{Second-order}\\ \text{differences}}}) - \frac{1}{3}(1-\Phi)(\underbrace{L(\mathbf{Q}_i) - L(\mathbf{Q}_j)}_{\substack{\text{Fourth-order}\\ \text{differences}}})]}^{\text{Artificial dissipation}}\}, \tag{2}$$

where $\mathbf{Q}$ is the nodal vector of *conservative* variables, A_{ij} is the *Jacobian* matrix, $L(.)$ is the discrete *Laplacian* operator and Φ is the *JST* switch[42] defined by differences of static pressure. The idea behind this approach is to provide a second-order accurate reconstruction of fluxes through the centered scheme. Where the flow is relatively smooth (e.g., no shocks), the dissipative part will remove all the higher than second-order terms. On the other hand, in the opposite case, the weight of lower-order quantities is increased reducing the overall accuracy to first, thus damping oscillations. The reason behind choosing such an approach is related to its speed,[61] however, to increase its accuracy *w.r.t.* the basic scheme, *upwinding* is included in the dissipative term by considering the reasoning of Reference 63. Higher-order reconstruction of the dissipative fourth-order terms is provided by the *MUSCL* scheme (*Monotonic Upwind scheme for Scalar Conservation Laws*).[64,65]

Solution of the discretized relations is carried out by marching in time through an *explicit Runge-Kutta* formulation[66] preconditioned with *Block-Jacobi*.[56]

2.3 | Adjoint solver

The flow adjoint equations are generally defined as:

$$\frac{\partial\mathbf{R}}{\partial\mathbf{Q}}\Big|^T\boldsymbol{\Psi} = \frac{\partial f}{\partial\mathbf{Q}}\Big|^T, \tag{3}$$

where $^{\partial\mathbf{R}}/_{\partial\mathbf{Q}}$ is the flow *Jacobian*, $\boldsymbol{\Psi}$ is the adjoint/dual vector and the *right-hand-side* (*R.H.S.*) term is the functional/integral quantity (f) sensitivity to the conservative flow variables. The dual solver present in *Hydra* falls into the *discrete* category, as it corresponds to a consistent backward linearization of the discretised flow solver. This is achieved by employing *automatic/algorithmic differentiation* (*AD*) of the source code in reverse, thus sensibly reducing the development burden. The *AD* package employed to do so is *TAPENADE*,[67] developed by *INRIA*. The pseudo-time stepping scheme utilised in this work is the standard *Runge-Kutta* approach as described by Reference 68.

3 | MESH ADAPTATION

3.1 | Feature-based

The feature-based approach considered in this work is based on the determination of the *Hessian* matrix of a scalar quantity of interest (q):

$$\mathbf{H}(q) = \begin{bmatrix} \frac{\partial^2 q}{\partial x^2} & \frac{\partial^2 q}{\partial xy} & \frac{\partial^2 q}{\partial xz} \\ \frac{\partial^2 q}{\partial yx} & \frac{\partial^2 q}{\partial y^2} & \frac{\partial^2 q}{\partial yz} \\ \frac{\partial^2 q}{\partial zx} & \frac{\partial^2 q}{\partial zy} & \frac{\partial^2 q}{\partial z^2} \end{bmatrix}.$$

It is often the case that some degree of smoothing is required to remove spurious oscillations from the computed *Hessian* field,[69-72] this issue being particularly evident close to the wall. In this work, instead of applying any artificial smoothing, the calculation of derivatives through the use of a *Least-Squares* (*LSQ*) formulation[73] was attempted. In fact, when developing this gradient operator, rather than employing only nodes directly connected to each other, three layers of neighbours were considered (see Figure 2). This meant that, in the case of a fully hexahedral grid, roughly 100 nodes would be considered to compute the derivatives.

It should be noted that, in general, the *Hessian* matrix entries are calculated by applying gradient operators twice, once to the scalar value and then to the first-order derivatives. However, in this case the *LSQ* matrices are sufficiently large to directly compute the second-order derivatives in a single iteration:

$$\underbrace{\begin{bmatrix} \Delta x_{1,j} & \Delta y_{1,j} & \Delta z_{1,j} & (\Delta x_{1,j})^2 & \Delta x_{1,j}\Delta y_{1,j} & \Delta x_{1,j}\Delta z_{1,j} & \Delta y_{1,j}\Delta x_{1,j} & (\Delta y_{1,j})^2 & \Delta y_{1,j}\Delta z_{1,j} & \Delta z_{1,j}\Delta x_{1,j} & \Delta z_{1,j}\Delta y_{1,j} & (\Delta z_{1,j})^2 \\ \Delta x_{2,j} & \Delta y_{2,j} & \Delta z_{2,j} & (\Delta x_{2,j})^2 & \Delta x_{2,j}\Delta y_{2,j} & \Delta x_{2,j}\Delta z_{2,j} & \Delta y_{2,j}\Delta x_{2,j} & (\Delta y_{2,j})^2 & \Delta y_{2,j}\Delta z_{2,j} & \Delta z_{2,j}\Delta x_{2,j} & \Delta z_{2,j}\Delta y_{2,j} & (\Delta z_{2,j})^2 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \Delta x_{n,j} & \Delta y_{n,j} & \Delta z_{n,j} & (\Delta x_{n,j})^2 & \Delta x_{n,j}\Delta y_{n,j} & \Delta x_{n,j}\Delta z_{n,j} & \Delta y_{n,j}\Delta x_{n,j} & (\Delta y_{n,j})^2 & \Delta y_{n,j}\Delta z_{n,j} & \Delta z_{n,j}\Delta x_{n,j} & \Delta z_{n,j}\Delta y_{n,j} & (\Delta z_{n,j})^2 \end{bmatrix}}_{\mathbf{\Delta x}_{ij}} \cdot \underbrace{\begin{bmatrix} \frac{\partial q}{\partial x} \\ \frac{\partial q}{\partial y} \\ \frac{\partial q}{\partial z} \\ \frac{1}{2}\frac{\partial^2 q}{\partial x^2} \\ \frac{\partial^2 q}{\partial x \partial y} \\ \frac{\partial^2 q}{\partial x \partial z} \\ \frac{\partial^2 q}{\partial y \partial x} \\ \frac{1}{2}\frac{\partial^2 q}{\partial y^2} \\ \frac{\partial^2 q}{\partial y \partial z} \\ \frac{\partial^2 q}{\partial z \partial x} \\ \frac{\partial^2 q}{\partial z \partial y} \\ \frac{1}{2}\frac{\partial^2 q}{\partial z^2} \end{bmatrix}}_{\begin{bmatrix} \nabla q_j \\ \mathbf{H}(q)_j \end{bmatrix}} = \underbrace{\begin{bmatrix} \Delta q_{1,j} \\ \Delta q_{2,j} \\ \cdots \\ \cdots \\ \cdots \\ \Delta q_{n,j} \end{bmatrix}}_{\mathbf{\Delta q}_{ij}}$$

An important subtlety that ought to be mentioned, is the treatment of rotational periodic boundaries, such as those present in turbomachinery. To account for this type of condition, the $\mathbf{\Delta x}$ and $\mathbf{\Delta q}$ arrays were increased in

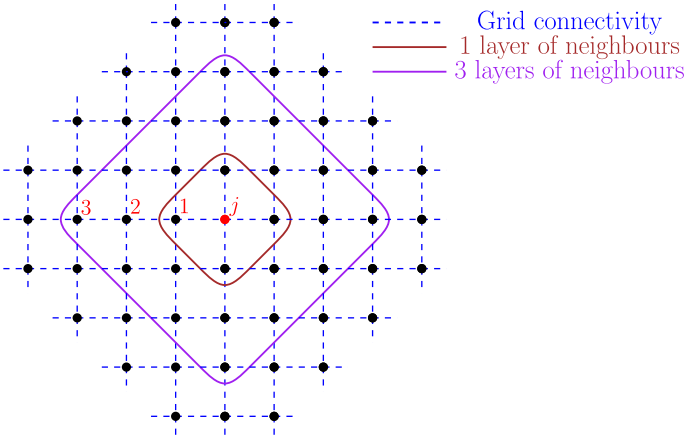


FIGURE 2 Two-dimensional stencil for *LSQ* derivative calculation of node *j* [Color figure can be viewed at wileyonlinelibrary.com]

size, with the aim of storing the original and rotated values of these boundaries. By employing a flag to indicate on which side of the periodic surface the node of interest is, it is possible to determine which parameter to use.

As stated by Reference 74, by including a diagonal matrix $\mathbf{W}$ containing the inverse of *Euclidean* distances between nodes, issues with high-aspect ratio cells can be diminished. Therefore, the system may be written as:

$$\mathbf{W} \cdot \Delta\mathbf{x} \cdot \begin{bmatrix} \nabla q \\ \mathbf{H}(q) \end{bmatrix} = \mathbf{W} \cdot \Delta\mathbf{q}.$$

This is then solved employing a *QR* factorisation consisting of a decomposition of the $\Delta\mathbf{x}$ matrix into orthonormal and upper triangular matrices.[75]

Once the *Hessian* matrix of second derivatives has been evaluated, this is employed to determine the feature-related error. In a *FVM* solution, multiplication of second-order derivatives by the characteristic edge length represents the error present relative to the parameter employed. In other words, it can be seen as the deviation from a linear behavior of the quantity along that edge.[15] By splitting the *Hessian* matrix into eigenvalues (λ) and eigenvectors ($\mathbf{v}$) it is possible to define a *Riemannian* metric ($\mathbf{M}$):

$$\mathbf{M} = \mathbf{v} \mid \lambda \mid \mathbf{v}^T. \tag{5}$$

It should be mentioned that to actually obtain such a metric, the matrix ought to be *positive-semi-definite* reason for which the absolute values of the eigenvalues are considered. The crucial idea behind this approach is to extract error directionality, i.e. achieve a fully *anisotropic* representation. In fact, while the inverse of the eigenvalues indicates how the edge should shrink/stretch to reduce the error, the eigenvectors produce information concerning directionality of the edge stretching/shrinking (Figure 3).

The *Riemannian* metric represents the error relative to a certain flow parameter, therefore by *equidistributing* it evenly over grid edges allows to achieve the optimal grid.[76] Considering the edge length in *Cartesian* space as $\mathbf{d}$, this requirement corresponds to Reference 6:

$$\mathbf{d}^T\mathbf{M}\mathbf{d} = 1 \tag{6}$$

thence, to achieve the optimal mesh, each edge ought to have unit length in *Riemannian* space. This means the error is identical in all directions. However, where this is not the case, a new edge length can be determined by:

$$d^* = \sqrt{\mathbf{d}\mathbf{M}\mathbf{d}} \tag{7}$$

and is consequently *equidistributed* over the mesh. In this work, even distribution of the error is achieved by a linear spring-stiffness approach.[15] This considers each grid edge (between nodes i, j) as a spring with stiffness set to:

$$\kappa_{ij} = \frac{d^*_{ij}}{\|\mathbf{x}_i - \mathbf{x}_j\|}. \tag{8}$$

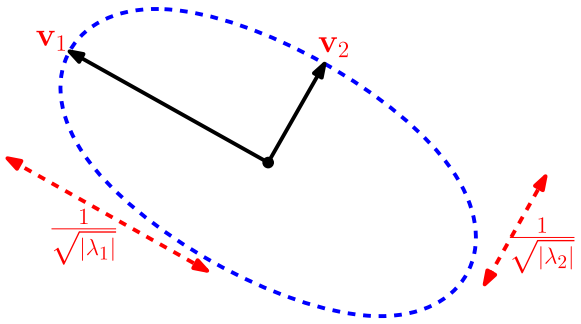


FIGURE 3 Two-dimensional *Riemannian* metric in physical space
[Color figure can be viewed at wileyonlinelibrary.com]

Therefore, the larger the error, the greater the spring stiffness. To achieve error *equidistribution*, the potential energy of the system of springs is minimised. For a generic node i, this corresponds to moving to a new position determined by:

$$\mathbf{x}_i = \frac{\sum_{j=1}^{\mathrm{E}_i}(\kappa_{ij}\mathbf{x}_j)}{\sum_{j=1}^{\mathrm{E}_i}\kappa_{ij}}, \tag{9}$$

where the summation is carried out over all the edges connected to node i. This process would not allow a smooth relocation of the grid's nodes, possibly generating negative volumes. Therefore, movement is carried out in a series of increments:

$$\Delta\mathbf{x}_i = \mathbf{x}_i^{\mathrm{new}} - \mathbf{x}_i^{\mathrm{old}} = \frac{\sum_{j=1}^{\mathrm{E}_i}\kappa_{ij}(\mathbf{x}_j^{\mathrm{old}} - \mathbf{x}_i^{\mathrm{old}})}{\sum_{j=1}^{\mathrm{E}_i}\kappa_{ij}} \tag{10}$$

the new grid coordinate can then be determined as:

$$\mathbf{x}_i^{\mathrm{new}} = \mathbf{x}_i^{\mathrm{old}} + \omega\Delta\mathbf{x}_i, \tag{11}$$

where ω is a relaxation factor $\omega = [0, 1]$.

3.2 | Adjoint-based

The adjoint adaptation technique employed is that developed by Reference 49 given it has repeatedly shown to be a reliable process.[35,77] It is based on a linear and quadratic interpolation (υ and Υ operators, respectively) of flow and adjoint quantities from a coarse mesh (spacing h), to an embedded mesh (spacing H), where the sensor calculation is carried out. It should be noted that neither the fine grid flow nor adjoint converged solutions are used in the interest of minimising time consumption. Once these quantities have been determined, the error sensor (Err) may be computed as:

$$\mathrm{Err} = \frac{1}{2}\sum_j \{|\underbrace{[\mathbf{R}(\upsilon\boldsymbol{\Psi}_H)_h]_j^T[\Upsilon\mathbf{Q}_H - \upsilon\mathbf{Q}_H]_j|}_{A} + |\underbrace{[\Upsilon\boldsymbol{\Psi}_H - \upsilon\boldsymbol{\Psi}_H]_j^T[\mathbf{R}(\upsilon\mathbf{Q}_H)_h]_j|}_{B}\}, \tag{12}$$

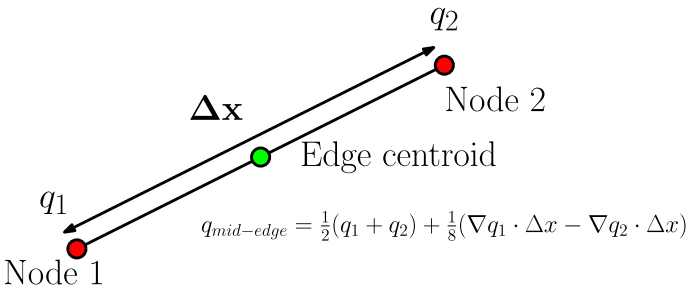
where terms A and B are different formulations of the error introduced subject to using an interpolated adjoint solution. The idea behind considering both terms, relates to the so-called *duality-gap*, i.e. the non-linearities present in the flow solution that are filtered out by the adjoint solver. A final note regarding this relation, concerns the use of linear and quadratic prolongation operators: this is necessary to mimic the difference between a high- and a low-order solution on the embedded mesh.

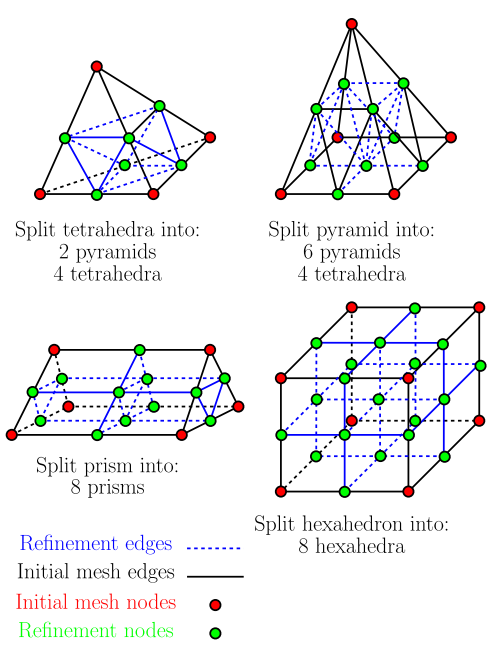
At this point, a description of linear and quadratic interpolation functionality from coarse to fine grids is required. In both cases, for embedded mesh nodes also present on the coarse counterpart, the values were simply copied. Concerning the linear operator, the remainder of the parameters was obtained by a simple averaging process, resulting in a perfectly *monotone* solution. Hence, for nodes splitting a coarse mesh edge, flow variables would be determined by averaging those points forming the original edge. On the other hand, the quadratic operator is more embroiled, requiring the determination of gradients. To achieve a smoother derivative representation, a similar approach as discussed in Section 3.1 was employed: three layers of neighbouring nodes formed the stencil in the *LSQ* matrices. Once the required gradients had been established, an *Hermitian* polynomial was used to evaluate the interpolated quantities (see Figure 4).

An important subtlety that ought to be mentioned relates to the *monotonicity* of flow and adjoint fields resulting from the quadratic interpolation. In fact, oscillatory behavior can occur in the approximated quantities. This is particularly true for nodes in the vicinity of cells exhibiting large size jumps or strong geometrical curvature. However, as discussed by Reference 35, the main issue is related to negative turbulent viscosity appearing. In this work, any unphysical quantity was removed by including appropriate checks reverting the interpolation to linear if necessary.

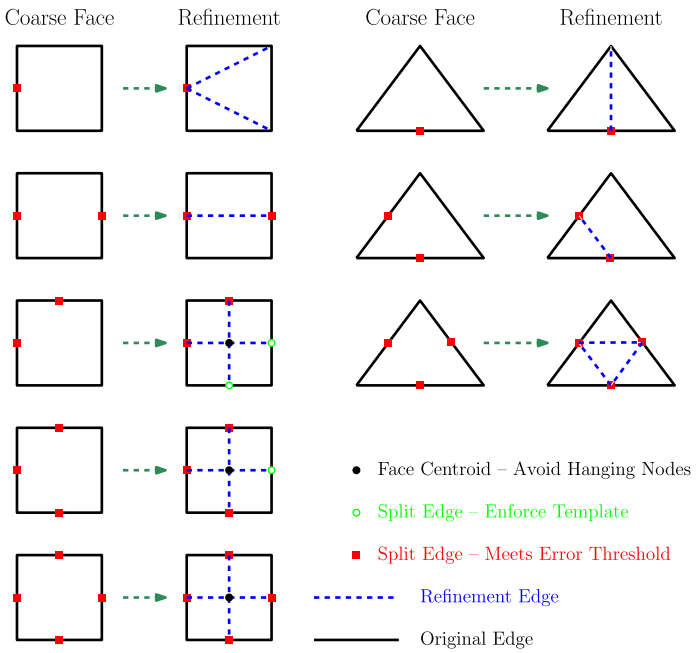
Once the error sensor has been evaluated, it is restricted to the coarse mesh by means of a control-volume weighted interpolation similar to that employed in certain *multigrid* schemes.[56] After the error has been interpolated back to the starting grid, cell-based refinement is carried out, i.e. if the average error of the nodes forming that volume is higher

FIGURE 4 *Hermitian* interpolation along an edge[59]
[Color figure can be viewed at wileyonlinelibrary.com]





(A) 3D cell refinement templates[59].



(B) 2D face refinement templates[53].

FIGURE 5 Refinement strategy [Color figure can be viewed at wileyonlinelibrary.com]

than the user-defined threshold, then all edges are split. For a generic mesh comprising of standard elements (tetrahedra, pyramids, prisms and hexahedra), the templates shown in Figure 5A are used to refine a marked cell.

The flow solver is only able to handle *conformal* grids, therefore an important clarification should be made concerning cells neighbouring those marked to be split. In this case, the refinement algorithm will follow a different approach: it will first split each element into its faces and determine whether any extra edges ought to be marked to match the templates displayed in Figure 5B. Once this has been carried out, each cell's faces are reassembled so the refinement can take place. In the eventuality that face marking combinations do not match any of the cell splitting templates, extra edges are refined. An important subtlety that must be mentioned concerns the near-wall refinement. In fact, the code employed in this work allows to propagate the wall refinement by a user-defined number of layers in the normal direction to the geometrical surface. This means that if an edge tangential to the surface is marked for splitting, the user may select to propagate this to the neighbouring layers (see Figure 6). This capability is helpful in maintaining a good quality grid for turbulence model purposes. After repeated tests carried out during this research, it was found that 10 layers of propagation allow the best robustness.

3.3 | Overall process

The main novelty of this procedure consists in the sequential use of feature and adjoint-based mesh adaptation approaches to improve a grid solution. Repeated feature-based mesh movement is exploited at the start to maximize the grid alignment

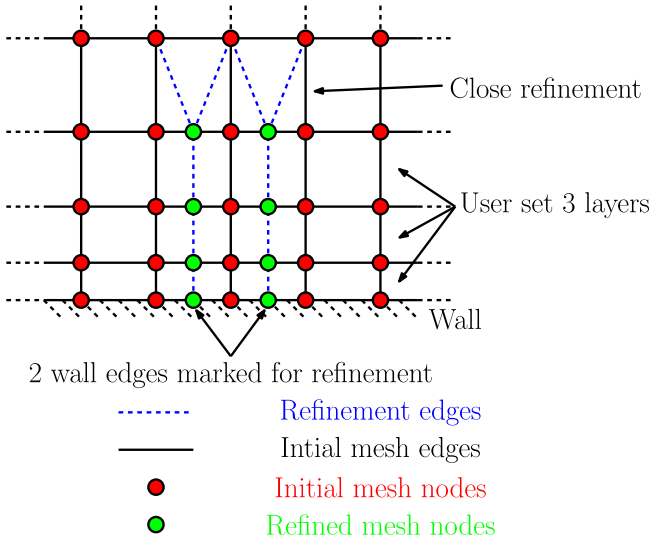


FIGURE 6 Two-dimensional example of wall-normal refinement[59] [Color figure can be viewed at wileyonlinelibrary.com]

with the complex flow features. This technique is relatively inexpensive as the node count and connectivity remains constant. Moreover, the flow solution from the current grid can be used as the initial one for the following adapted mesh.

The reason for starting with the approach described in Section 3.1 relates to the consequent improvement of adjoint error estimates, as focal regions of the mesh will be better resolved. In fact, it is generally the case that part of the flow complexities affect the performance quantity of interest.

The second part of the process consists of a single adjoint-based mesh refinement, carried out in a cell-based manner.

A block diagram of the overall process is reported in Figure 7: it can easily be seen that the process is composed by two separate techniques run in sequence.

4 | TEST-CASES

4.1 | *NASA Rotor* 37 compressor blade[78,79]

In order to show the grid modification process validity, once the starting coarse mesh had been generated, an accurate comparison for this was achieved by uniformly refining it twice. Given the ∼0.6M node count of the starting mesh, this means a 36.5M grid was created. In the following sections, after a brief summary of the case setup, the finer case result at operating point (mass flow of 20.51 kg/s) is compared with experimental data to check its validity, and then used as a means to describe the physical characteristics of the blade.

The computational domain is displayed in Figure 8 (location of static and moving parts and their velocity follow the specifications of Reference 80). To achieve the full annulus, periodic repeats of each single blade may be computed as a post-processing step.

The inlet boundary condition consisted of a subsonic inflow, set according to the data provided by Reference 78. On the other hand, subsonic radial equilibrium was imposed at the outflow.

The 36.5M solution at mid-span is reported in Figure 9. The main feature consists of the shock forming due to the blade's blunt *leading-edge* (*LE*), accurately captured by means of the structured mesh. This then propagates both, in the rotational and counter-rotational directions. In the rotational direction, it hits the adjacent element's suction side, with separation occurring due to the adverse pressure gradient. This results in the formation of a large wake, that is then convected downstream. Counter-rotationally, instead, shock propagation is smeared out till it hits the computational domain inlet.

The experimental absolute total temperature ratio radial profile (evaluated between inlet/outlet and circumferentially mass-averaged) at operating point (98% choke, corresponding to 20.51 kg/s) is compared with the fine mesh solution in Figure 10A. This shows how the *CFD* parameter trend follows generally quite well the experimental data. There are, however, some exceptions. In the lower span region (≤10%) the *CFD* data overpredicts the temperature ratio trend. This behavior has been noted before, and according to Reference 81, it is due to hub leakage flow upstream of the blade not

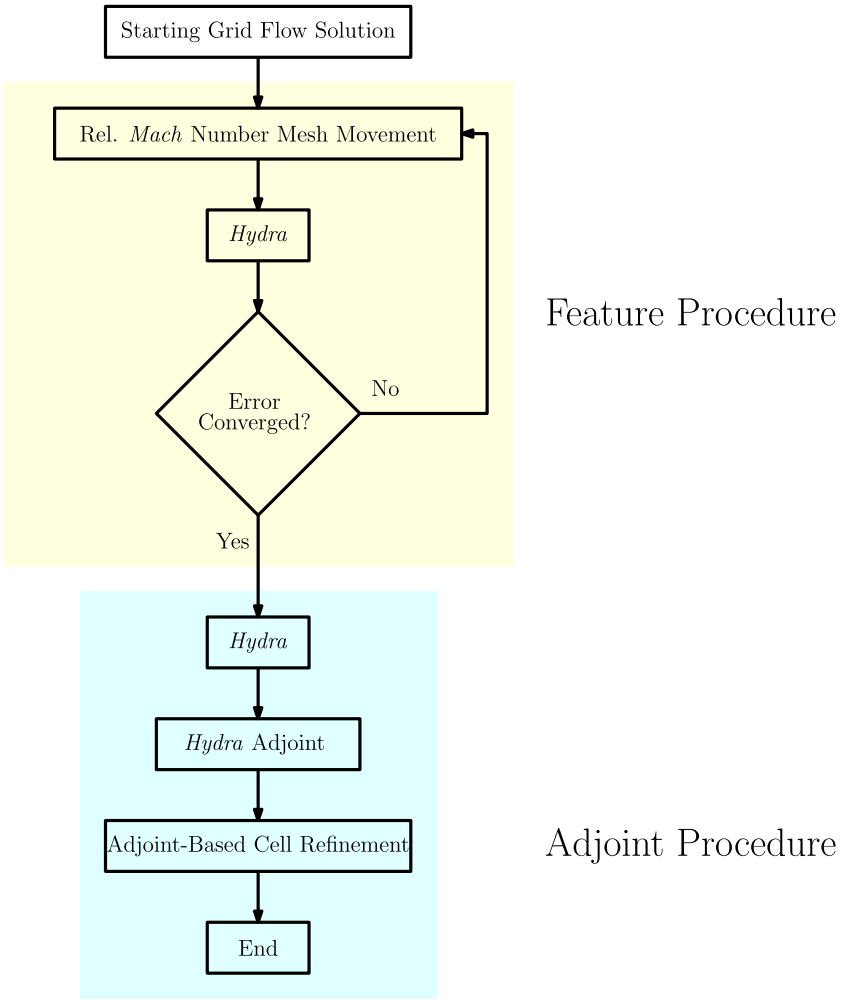


FIGURE 7 Combined feature-based mesh movement and adjoint-based refinement schematic[59] [Color figure can be viewed at wileyonlinelibrary.com]

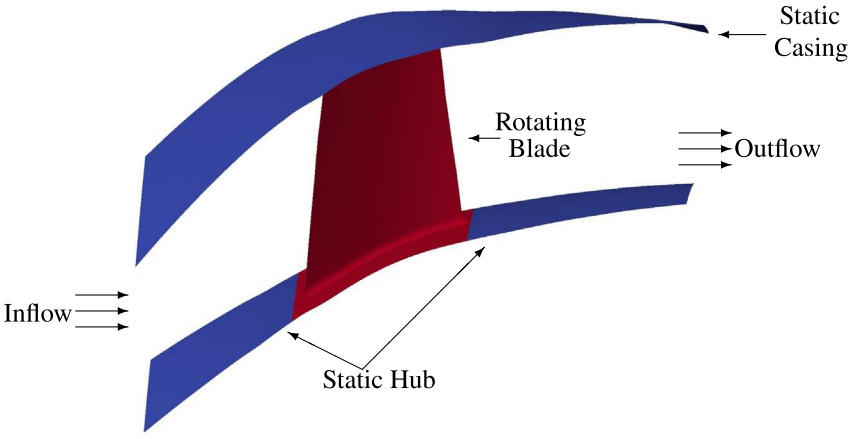


FIGURE 8 *NASA Rotor* 37 computational domain [Color figure can be viewed at wileyonlinelibrary.com]

being included in the computational modelling. Another region where experimental/*CFD* data discrepancy is frequently encountered is the tip. This was attributed by Reference 82 to the employment of *RANS* equations, unable to handle unsteadiness forming due to the tip-leakage-shock interaction.

An additional comparison between experimental and *CFD* data is provided in Figure 10B. This shows the chord-wise relative *Mach* number profile along the mid-pitch line at 70% span. As it can clearly be seen, the numerical simulation is in good agreement with the experiment. The location of flow complexities is appropriately captured by *Hydra*. Slight over- and undershoots are visible where stronger changes in the flow conditions appear.

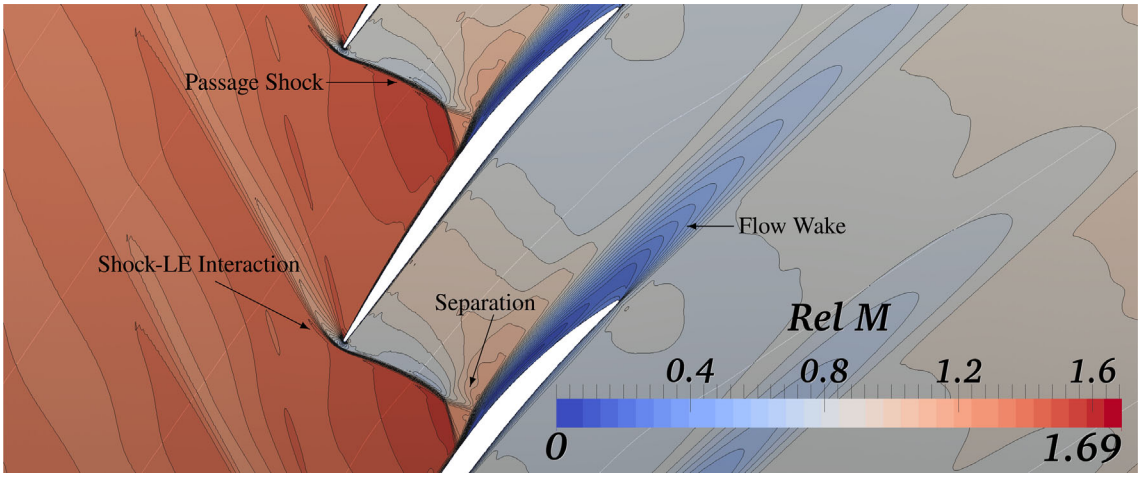
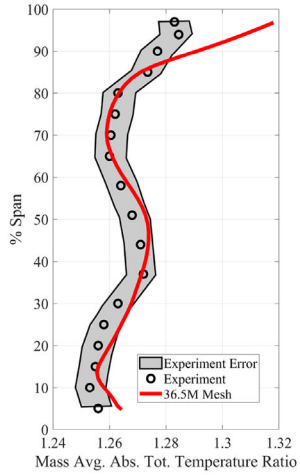
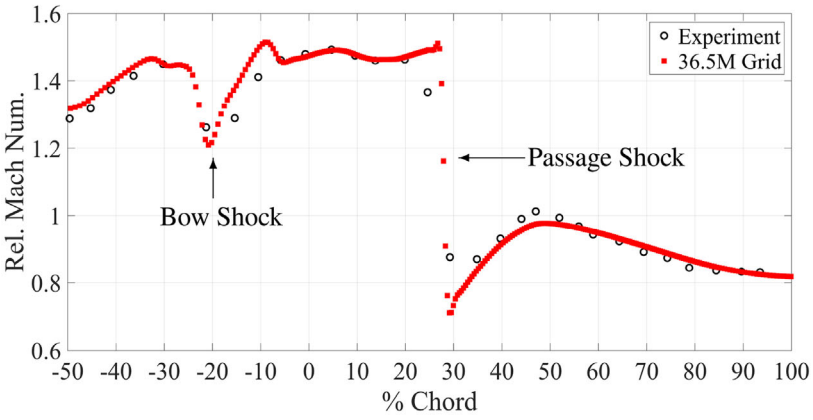


FIGURE 9 36.5M mesh 50% span-wise cut displaying the relative *Mach* number variation at operating point [Color figure can be viewed at wileyonlinelibrary.com]



(A) Radial profile of absolute temperature ratio (mass averaged).



(B) Chord-wise relative *Mach* number along the mid-pitch line at 70% span.

FIGURE 10 *NASA Rotor* 37 case *CFD* (36.5M) and experimental data[80] comparison [Color figure can be viewed at wileyonlinelibrary.com]

From the data presented in this section, significant agreement between the experimental data and that produced by the flow solver has been shown. It can therefore be concluded that the 36.5M grid is an appropriate benchmark to use for comparison relative to the proposed adaptation algorithm.

4.2 | Fan blade

The other test-case considered in this work concerned a typical fan blade with splitter, present in modern jet engines. The *Reynolds number* for this particular simulation, estimated employing mid-span quantities, was equal to 2.2×10^5.

In this case, the starting structured multi-block mesh had 0.8M nodes: the resulting comparison solution was achieved on a grid generated by repeated uniform refinement of the initial one (final node count of 50M). Similarly to the previously discussed compressor, this will be employed to describe the flow characteristics of this rotor. The test-case *CFD*

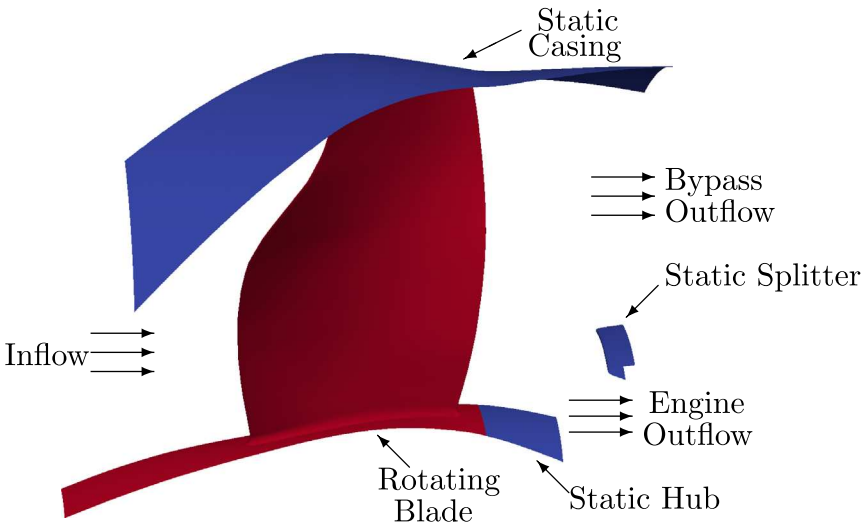
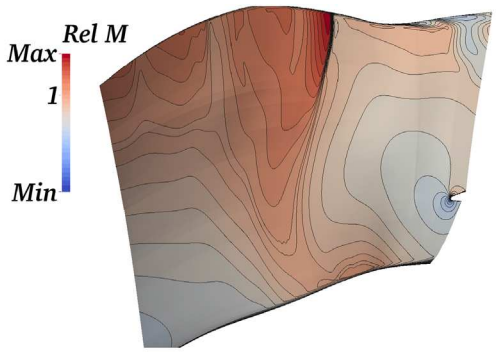
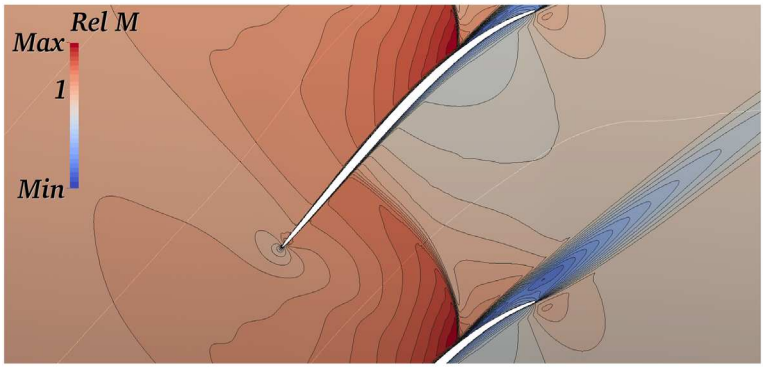


FIGURE 11 Computational setup of fan blade with splitter (not to scale) [Color figure can be viewed at wileyonlinelibrary.com]



(A) Periodic boundary.



(B) 75% span.

FIGURE 12 Fan blade relative *Mach* number distribution on the 49.6M grid [Color figure can be viewed at wileyonlinelibrary.com]

setup is shown in Figure 11. It is clearly visible, that the casing, splitter, and downstream part of the hub are static, while blade and upstream part of the hub are rotating. The flow exiting the domain above the splitter enters the bypass region, while the one below is directed towards the engine core. Concerning the boundary conditions, the inlet was set to subsonic, while both exits had the same setup as the compressor of the previous section, namely, subsonic radial equilibrium.

Choke flow conditions at the bypass were considered for this case, with a strong shock forming on the suction side above mid-radius causing a degree of separation (Figure 12A). Moving towards the casing, this feature's strength increases, propagating across the passage hitting the neighbouring blade's pressure side. This results in separation and a wake forming that widens around 75% span (Figure 12B).

5 | RESULTS

5.1 | *NASA Rotor* 37

Following the detailed discussion in Section 4.1, the integral quantities of interest chosen were adiabatic efficiency (η) and absolute total pressure (Pr), both evaluated between inlet-outlet planes and circumferentially mass-averaged. In addition, two area-averaged parameters were considered: the mass-flow ($\dot{m}$) integrated over the exit plane and torque (τ), computed over the blade surface. Their estimates on the coarse mesh (0.6M) and the finer counterpart (36.5M) are reported in Table 1.

The differences between efficiency, mass-flow and torque values are clearly noticeable, while Pr has a relatively smaller inaccuracy, due to its reduced magnitude.

5.1.1 | Feature-based step

Given the starting mesh, the grid movement adaptation procedure was repeated until the average edgewise error converged. Comparisons between the starting and adapted grids are reported in Figures 13 and 14. The first set of images

Grid	Node count	η (%)	$\dot{m}$ (kg/s)	Pr	τ ($N \cdot$ m)
Coarse	590294	85.116	20.374	2.0573	−879.428
Fine	36526821	85.523	20.509	2.0699	−889.628

TABLE 1 Performance quantities estimate on coarse and fine mesh for *NASA Rotor* 37

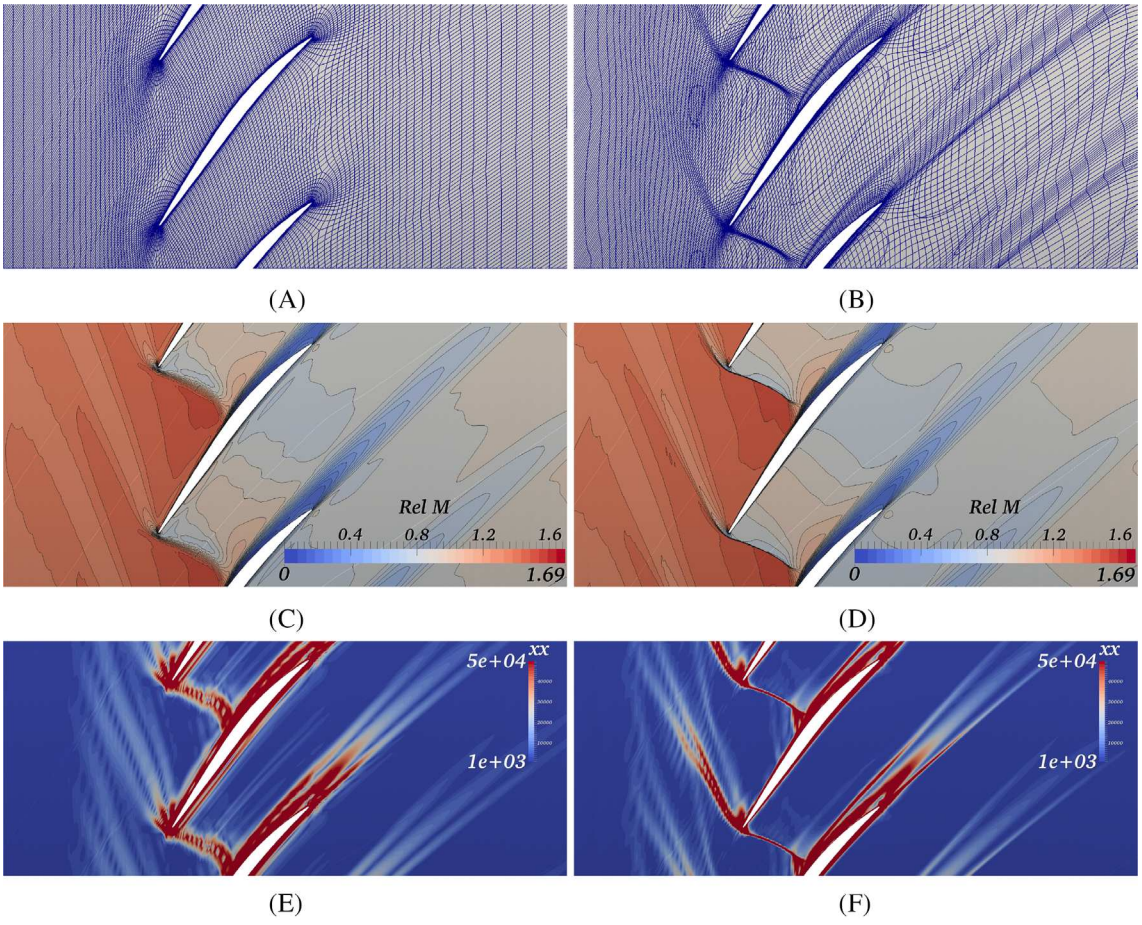


FIGURE 13 Starting (*LHS* column) and feature-based mesh moved (*RHS* column) grid, solution and $\left|{}^{\partial^2 M}/_{\partial x^2}\right|$ at mid-span [Color figure can be viewed at wileyonlinelibrary.com]

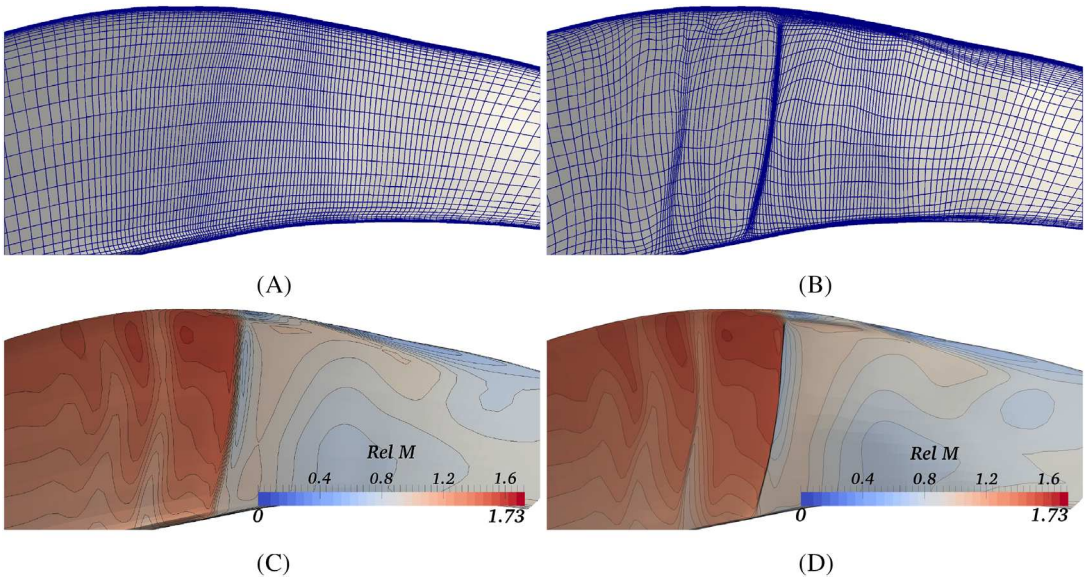


FIGURE 14 Starting (*LHS* column) and feature-based adapted (*RHS* column) grid and solution along the periodic boundary [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 2 Performance quantities evolution after feature-based mesh adaptation *NASA Rotor* 37

Grid	Node count	η [%]	$\dot{m}$ [kg/s]	*Pr*	τ [*N* · m]
Coarse	590294	85.116	20.374	2.0573	−879.428
Feature adapted	590294	85.277	20.437	2.0627	−883.649
Fine	36526821	85.523	20.509	2.0699	−889.649

shows how, at mid-span, the mesh movement has appropriately clustered and aligned the grid towards flow complexities (Figures 13A,B). The most aggressive nodal relocation has occurred around the strongest parts of the shock. This resulted in a very neat resolution of it (Figure 13D). It should be noted, that the shock/boundary layer interaction on the blade suction side is much better estimated, with the lambda shape being clearly visible. The wake has also attracted a significant amount of attention by the mesh adaptation software.

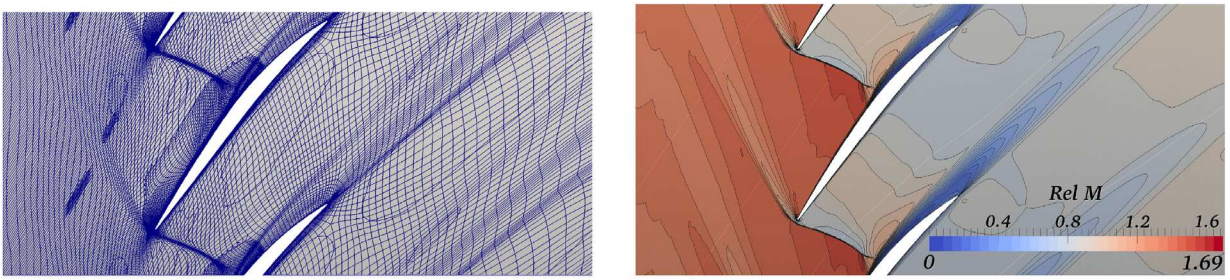
The first *Hessian* matrix eigenvalue $\left(\partial^2 M/\partial x^2\right)$ of the starting and adapted grids is reproduced in Figures 13E,F, respectively. Similarly to the actual flow solution, this shows how the second-order derivatives are better estimated. The shock has clearly thinned, particularly along the passage. Subtle differences are also visible at the wake.

Concerning the radial evolution of the mesh, the periodic boundaries of the starting and final feature adapted grids are shown in Figure 14. The passage shock resolution (Figure 14D) is much better. As shown in Figure 14B, there is a significant clustering of nodes towards the casing downstream of the shock, where the tip-gap vortex crosses the periodic surface.

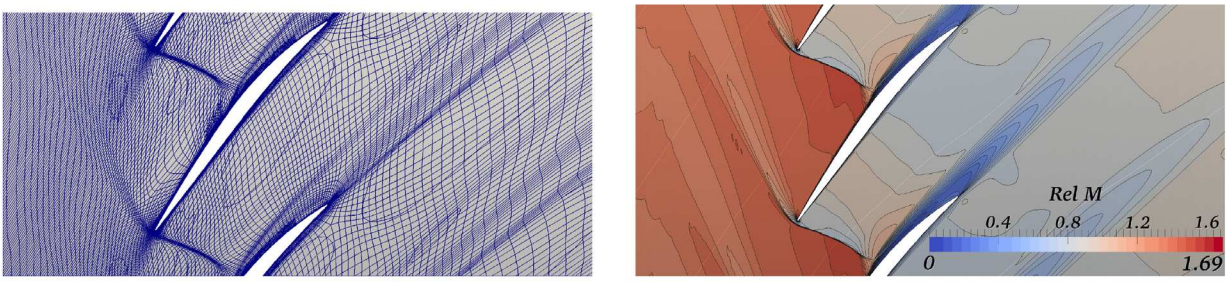
With regards to the evolution of performance quantities, these are reported, along with starting mesh and target values, in Table 2. All parameters have clearly improved, moving towards the accurate 36.5M mesh. This is helpful in preparing the mesh for adjoint adaptation. In fact, all these quantities have been enhanced simultaneously employing a single process that has not added any nodes.

5.1.2 | Adjoint-based step

As discussed in the previous section, the use of feature-based adaptation has helped align grid edges with the salient flow features. This has produced an improvement in the estimate of all quantities of interest. Figure 15 shows the resulting grids at mid-span after the feature-based mesh movement and the relative adjoint-error refinement for adiabatic efficiency and torque. There are visible differences and limited similarities in the adaptation. In fact, the torque-based refinement



(A) η-adjoint adaptation.



(B) τ-adjoint adaptation.

FIGURE 15 *NASA Rotor* 37 feature and adjoint adapted grids (*LHS* column) and relative *Mach* number (*RHS* column) at mid-span for various performance quantities [Color figure can be viewed at wileyonlinelibrary.com]

has targeted the near-blade surface on both, suction and pressure side. The remainder of the flow field has received very little attention, with the nodal addition propagating further off the blade surface only at the *LE* and shock-boundary layer interaction. On the other hand, for efficiency, the refinement extends from the suction-side into the passage, starting at the rotor's *LE* till just downstream of the shock. Clearly, improving the resolution of its interaction with the boundary layer and its propagation into the flow-field is crucial to the correct estimation of η. This is visible in their respective relative *Mach* number fields. Additionally, the blade pressure-side has not been targeted for adaptation in the efficiency case. As previously hinted to, in both cases the blade *LE* has been refined: this to better the stagnation point prediction.

A separate discussion should be made concerning the two refinement patches that appear in the upstream region. These relate to the functional error consisting of both flow and adjoint solutions along with relative residuals. In fact, they are due to the intersection of the adjoint "*reversed wake*" propagating from the blade suction-side to the inlet and the shock spreading off the blade *LE* in the upstream direction. To prove this point, the η adjoint field at mid-span has been included in Figure 16. As it may be seen, there is a streak of high sensitivity starting on the blade suction side at the *trailing edge* (*TE*). This then propagates towards the inlet, traversing the entire rotor. Effectively, it is indicating that any perturbation in this region of the flow will have a larger impact on the adiabatic efficiency than the regions surrounding it. From a mathematical point of view, this is due to the adjoint operator reversing the direction of first-order convection derivates in the *NS* relations. In simple terms, the adjoint solution is linearly predicting that the flow in this part of the domain will end-up onto the blade's suction-side, where most of the physical complexities take place. Therefore, any inaccuracy in capturing it will result in a higher error where the most significant flow features appear, meaning that the functional of interest will not be reliably predicted. In the case of torque, the upstream region was not refined, as the functional had much higher sensitivity to the near-wall portion of the mesh.

Interestingly, neither of the performance quantities required further wake improvement by means of refinement, indicating that the mesh movement has sufficiently clustered the mesh lines in this region.

The exact node count for all the grids is reported in Table 3. The adjoint refinement was carried out by adding approximately half the starting mesh number of nodes. This with the aim of better controlling the mesh size and to allow a fairer comparison between the cases.

Finally, the most important part of the study, i.e. the performance quantity prediction. Table 4 displays these values for the starting and target estimate, along with feature and adjoint-adapted grids and adjoint-only refined meshes. Additionally, Table 5 is provided, as it gives a better idea of the reduction in error provided by the feature and adjoint combined

FIGURE 16 *NASA Rotor* 37 adiabatic efficiency adjoint momentum terms at mid-span after the feature-based adaptation [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 3 Resulting adapted grids node count for *NASA Rotor* 37

Grid	Mesh size			
Coarse	590294			
Feature adapted	590294			
Fine	36526821			
Grid	η	$\dot{m}$	Pr	τ
Adjoint adapted	969397	935962	936361	1054964
Feature and adjoint adapted	927871	935663	942632	924712

TABLE 4 Performance quantities comparison after feature-, adjoint-, and feature and adjoint-based adaptation for *NASA Rotor* 37

Grid	η [%]	$\dot{m}$ [kg/s]	Pr	τ [$N \cdot m$]
Coarse	85.116	20.374	2.0573	−879.428
Feature adapted	85.277	20.437	2.0627	−883.649
Adjoint adapted	85.739	20.552	2.0772	−892.739
Feature and adjoint adapted	85.560	20.522	2.0728	−887.486
Fine	85.523	20.509	2.0699	−889.649

TABLE 5 Absolute difference of performance quantities between target and coarser grids for *NASA Rotor* 37

Grid	$\lvert\Delta\eta\rvert$ [%]	$\lvert\Delta\dot{m}\rvert$ [kg/s]	$\lvert\Delta Pr\rvert$	$\lvert\Delta\tau\rvert$ [$N \cdot m$]
Coarse	0.407	0.135	0.0126	10.221
Feature adapted	0.246	0.072	0.0072	6.000
Adjoint adapted	0.216	0.043	0.0073	3.090
Feature and adjoint adapted	0.037	0.013	0.0029	2.163

approach. In fact, in all cases, this strategy is the most effective in minimising inaccuracies. The use of adjoint adaptation on its own has provided a benefit, but this seemed to be closer to the feature-based performance than the combined procedure. In the case of pressure ratio, the error is greater than using feature-based mesh movement alone. For efficiency, it is of the same order of magnitude, while for mass flow it sits roughly halfway between the other two adaptation approaches. Only in the case of torque is the adjoint-only process closer to the combined technique. This is due to the higher sensitivity of the error to the blade near-wall mesh.

5.2 | Fan blade

The performance quantities of interest considered for the test-case described in Section 4.2 were the mass-averaged adiabatic efficiency evaluated between inlet and *bypass* outlet (η_{BP}) and *engine* outflow (η_{EN}) and the area-integrated torque (τ) over the blade surface. The actual difference between the 50M and starting 0.8M grid estimates are reported in Table 6.

Functional	50M - Starting 0.8M
$\lvert\Delta\eta_{BP}\rvert$ [%]	0.59
$\lvert\Delta\eta_{EN}\rvert$ [%]	0.58
$\lvert\Delta\tau\rvert$ [N · m]	93.41

TABLE 6 Absolute difference between target and coarse mesh performance quantities before adaptation

5.2.1 | Feature-based step

The resulting grids achieved by applying the mesh movement are reported in Figures 17 and 18. In the first, at 75% span, the shock clustering is clearly visible, along with that towards the fan surface on both pressure and suction side. As for the compressor case, significant movement has occurred towards the wake, with the original mesh lines closer to the outlet changing direction by roughly 45° *w.r.t.* the original grid (Figures 17A,B). These mesh modifications have significantly improved the shock propagation from the blade surface into the passage and the wake thickness has slightly changed (Figures 17C,D). More interestingly, though, the *xx*—component of the *Hessian* matrix (Figures 17E,F) is showing how the original shock is poorly resolved. In fact, not only does it smear out very close to the blade suction side surface, but it seems to be composed of two separate structures. Once the feature-based node movement has been applied, the complexity extends further into the flow field and the two lines merge into a single one. Figure 17F also shows that, by applying the vertex movement, the wake high-derivatives extend further, reaching the outflow.

Looking at the periodic boundary profile in Figure 18, the strongest change is the dense line from hub to tip improving the shock resolution. Despite this being strongest in the upper half of the span, where it also forms, node clustering

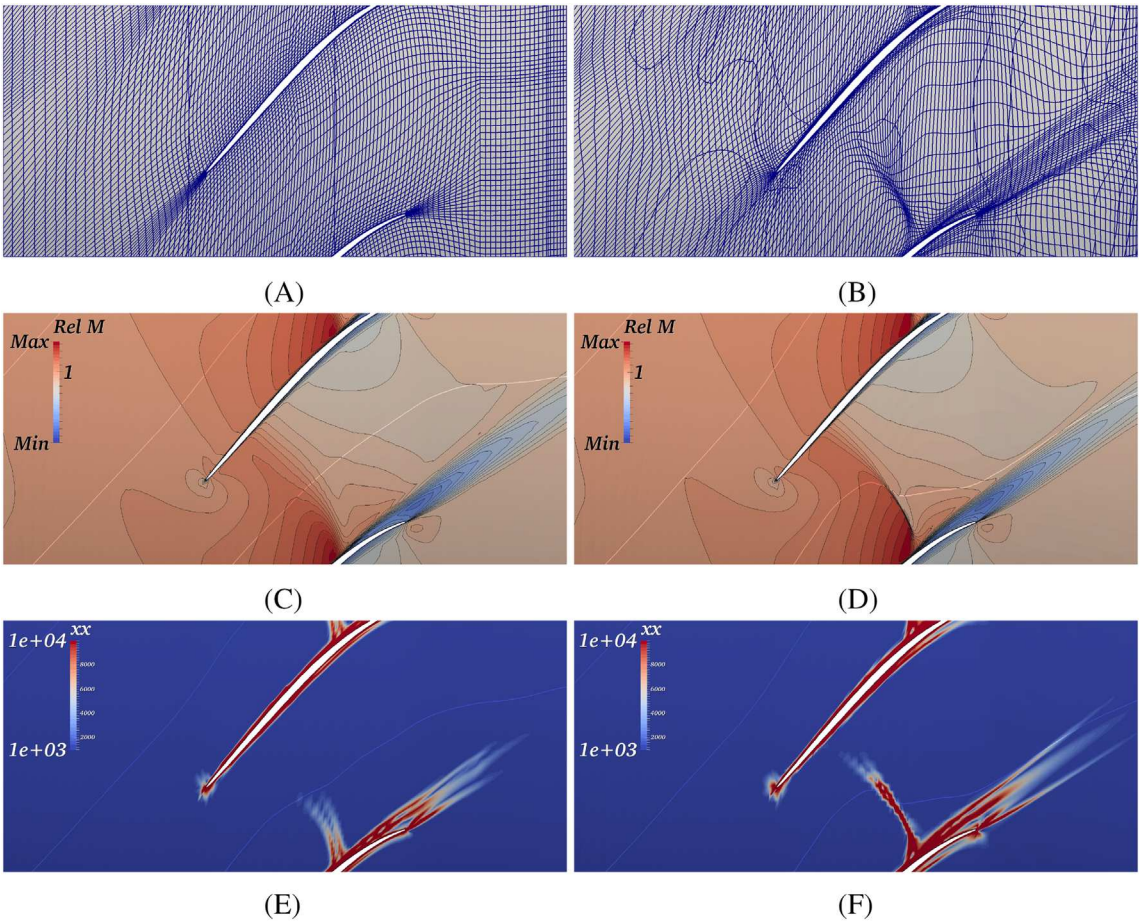


FIGURE 17 Starting (*LHS* column) and feature-based adapted (*RHS* column) grid, solution and error comparison at 75% span [Color figure can be viewed at wileyonlinelibrary.com]

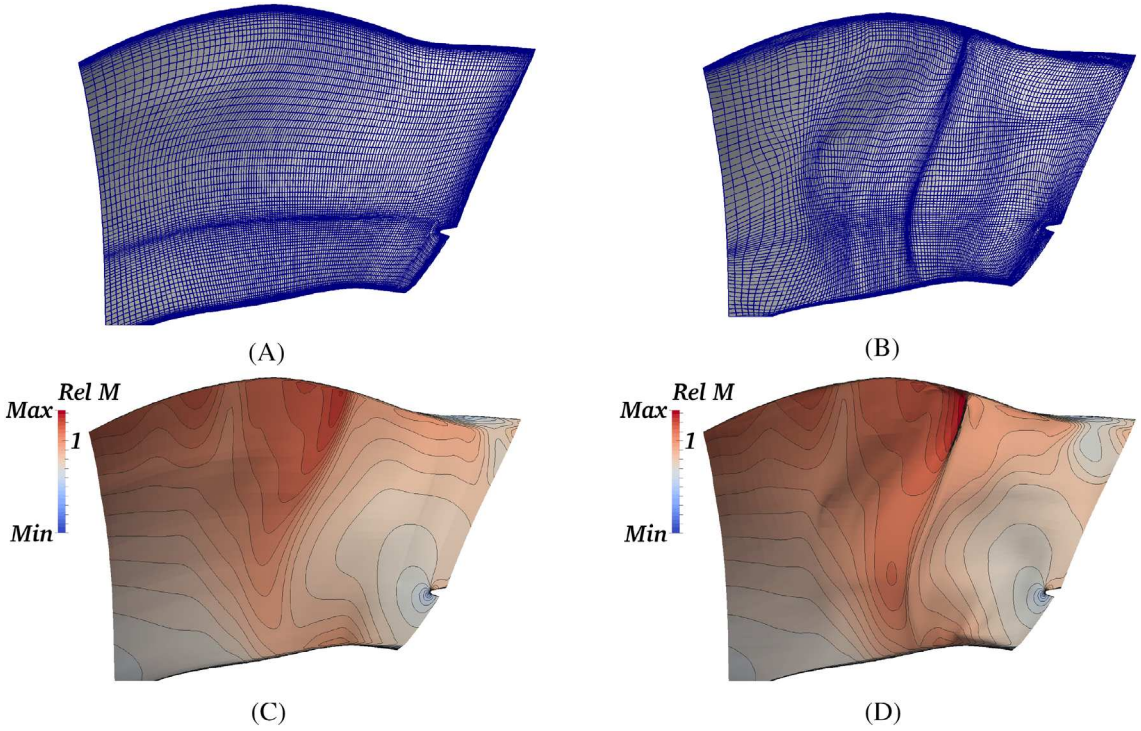


FIGURE 18 Starting (*LHS* column) and feature-based adapted (*RHS* column) grid and solution comparison along the periodic boundaries [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 7 Absolute difference between target and coarse mesh performance quantities after feature-based adaptation

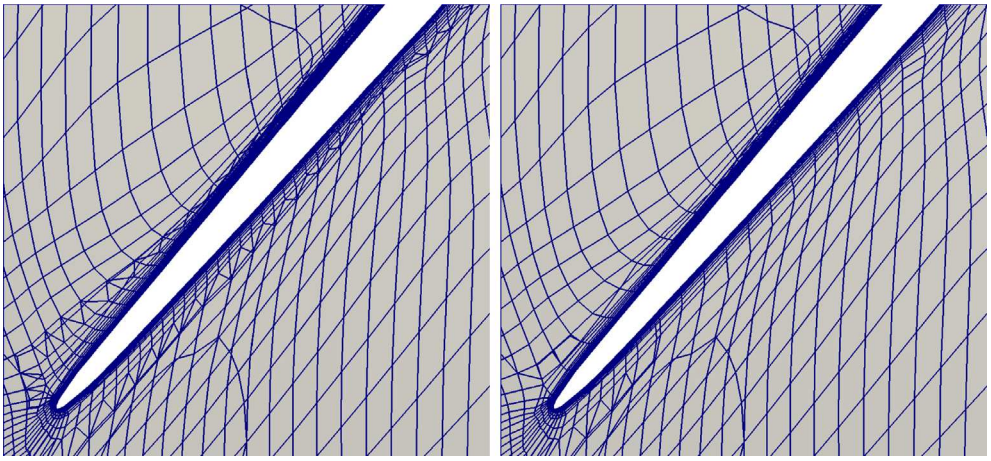
Functional	50M - Feature adapted 0.8M
$\lvert\Delta\eta_{BP}\rvert$ [%]	0.43
$\lvert\Delta\eta_{EN}\rvert$ [%]	0.28
$\lvert\Delta\tau\rvert$ [N · m]	30.09

towards it occurs at lower span locations as well. The tip-region downstream of the shock has also attracted a considerable amount of clustering: this is for the tip-gap flow interacting with the local flow features and propagating downstream. Finally, nodes have also been pushed towards the splitter, although this has not visibly changed the flow field around it.

As in the case of *NASA Rotor* 37, the performance quantities of interest have improved, with the difference between target and current estimate reduced *w.r.t.* that of the starting grid (see Table 7). Concerning torque, the nodal clustering towards the blade surface has clearly had the effect of significantly reducing the error, that is now over three times smaller. The error in adiabatic efficiency evaluated over the engine outflow has been halved, while that of the η_{BP} has seen a smaller improvement. This is due to the more complex flow field present at higher span, with the shock being much stronger in this region.

5.2.2 | Adjoint-based step

Following the feature-adaptation, the grid was separately refined for the efficiency values and torque. The resulting grids showed a degree of similarity, with the near-casing and the blade wall being targeted by the adaptation algorithm. However, differences were also visible, and to this end, Figure 19 has been included. This shows the near-wall refinement at 75% span, for the adiabatic efficiency integrated over the bypass (Figure 19A) and engine outflow (Figure 19B). It may be seen that a larger amount of nodes has been added when the η_{BP}-based refinement is applied. Moreover, the η_{EN} adaptation was the only one to target the hub and increase the near-wall refinement at lower radius locations. This can be easily explained by considering the surface of integration of the two quantities: while the engine outlet is radially located in the



(A) η_{BP} adapted grid at 75% span.

(B) η_{EN} adapted grid at 75% span.

FIGURE 19 Adjoint refinement examples for the fan case [Color figure can be viewed at wileyonlinelibrary.com]

lower third, the bypass exit covers the rest of the blade height. Interestingly, none of the quantities highlighted the flow features as being sources of high error, indicating that the feature-based approach has aided in reducing their inaccuracy contribution.

Unlike the compressor case, no refinement was present upstream of the blade (see Figure 15). While the adjoint "*reversed wake*" of high sensitivity propagating from the blade suction-side was visible in the relative solution, no flow feature was crossing it upstream of the *LE*, meaning that the error contribution would be much smaller.

The resulting mesh size for each functional is reported in Table 8. This includes the separate adjoint-refinement process not utilising the feature-based mesh movement. Approximately half of the starting mesh nodes were added as in the compressor case, to control the mesh size and allow a fairer comparison.

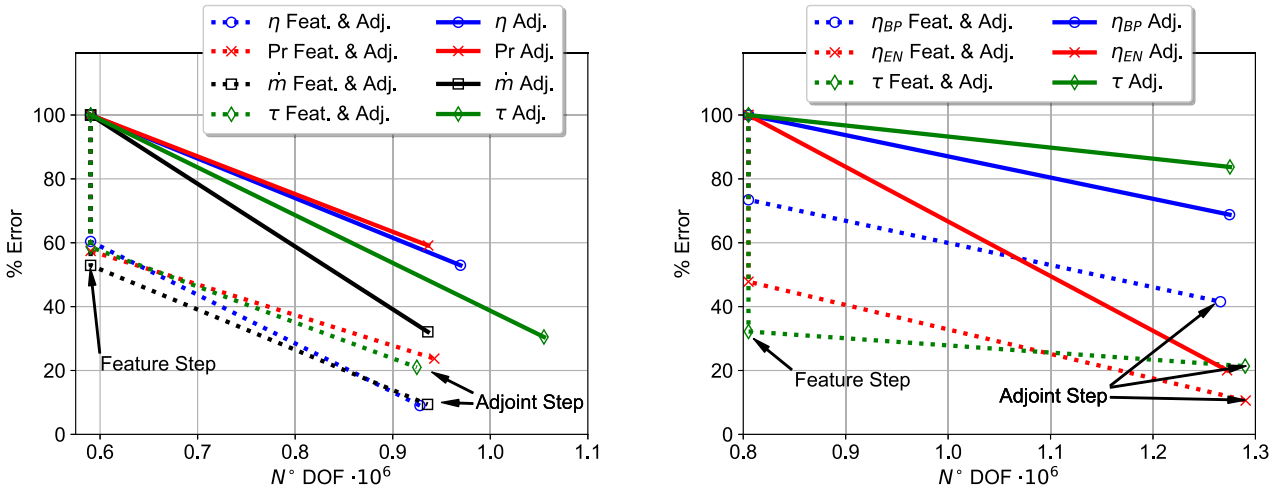
The errors relative to the fine mesh solution are shown in Table 9. It is obvious that the combination of feature and adjoint processes has generated non-negligible benefits. In fact, the use of feature-based methods on their own has given a reduction in error that is comparable to the standard adjoint refinement process. In the case of torque, the former has produced a much more significant error reduction, while for the efficiency integrated over the bypass the difference is only 0.03%. This is a clear indication that these quantities are significantly influenced by flow features and appropriate capturing of them is necessary to estimate them correctly. Multiple adjoint refinement steps would have been necessary to target the shock and wake, but this would have resulted in a much finer mesh.

TABLE 8 Resulting adapted grids node count for the fan blade case

Grid	**Mesh size**		
Coarse	805121		
Feature adapted	805121		
Fine	49677951		
Grid	η_{BP}	η_{EN}	τ
Adjoint adapted	1274727	1272352	1275148
Feature and adjoint adapted	1265887	1290487	1290049

TABLE 9 Absolute difference of performance quantities between target and coarser grids for the fan blade case

Grid	$\lvert\Delta\eta_{BP}\rvert$ **[%]**	$\lvert\Delta\eta_{EN}\rvert$ **[%]**	$\lvert\Delta\tau\rvert$ **[N · m]**
Coarse	0.59	0.58	93.41
Feature adapted	0.43	0.28	30.09
Adjoint adapted	0.40	0.12	78.21
Feature and adjoint adapted	0.24	0.06	19.95



(A) *NASA Rotor* 37 performance quantities.

(B) Fan blade performance quantities.

FIGURE 20 Error % relative to starting mesh error [Color figure can be viewed at wileyonlinelibrary.com]

6 | OVERALL PERFORMANCE COMPARISON

To summarize and clearly compare the performance of the combined approach with that consisting of adjoint refinement only, Figure 20 has been included. The error percentage is calculated *w.r.t.* that of the starting mesh (equal to 100%), while the abscissa values indicate the mesh degrees of freedom. The dotted lines represent the sequential feature and adjoint adaptation approach, while the continuous counterpart are the adjoint refinement results. A note should be made concerning the feature-based mesh movement: as this does not increase the number of grid nodes, the error reduction is a vertical straight line.

As previously discussed, all quantities are significantly improved by applying the proposed adaptation methodology *w.r.t.* the refinement only technique. For the case of *NASA Rotor* 37 (Figure 20A) there is an average error improvement with the combined approach of 84%, while using only adjoint refinement resulted on average in an accuracy enhancement of 56%. That is, for *NASA Rotor* 37 the former technique on average provides a further 28% error reduction on meshes with approximately the same node count.

Concerning the fan case (Figure 20B), the average improvement of the sequential feature and adjoint approach is again 85%, while the adjoint refinement on its own, on average provides an improvement of 53%, slightly less than in the compressor setup.

7 | CONCLUSIONS AND FUTURE WORK

This work has shown how feature-based methods can be successfully integrated into a combined adjoint-adaptation process in a sequential manner. This allows to simultaneously improve multiple quantities of interest, while maintaining the number of nodes constant. Moreover, the flow solution from the previous mesh may be employed as a starting one for the next adapted grid.

Consequent adjoint-based refinement showed further improvement of the quantities of interest on a very coarse mesh *w.r.t.* the grid-independent solution. In all cases considered and for all performance quantities, the combined approach provided the largest reduction in error, yet with approximately the same node count.

The methodology presented in this work was successfully applied to relatively complex industrial cases presenting multiple *3D* features interacting with each other. This represents an advancement to the standard *2D* and *3D* test-cases proposed within published literature.

Future developments of the approach consist of the inclusion of feature-based edge split and collapse applied during the mesh movement, to see whether this can further help reduce the performance quantities error with a limited increase in number of degrees of freedom. Moreover, it would be of helpful to apply adjoint-based coarsening, with the aim of reducing the computational demands in terms of hardware and run-time.

Finally, it would be interesting to test the sequential feature and adjoint-based technique in an adjoint geometrical optimisation framework to be able to ascertain what kind of improvements, in terms of performance and time consumption, it is able to provide *w.r.t*. the adjoint refinement only.

ACKNOWLEDGEMENTS

The authors would like to express their appreciation and gratitude to *Rolls-Royce plc* for the financial support, permission to utilise the *Hydra* software system, a pre-generated fan blade mesh and to publish this work.

ORCID

Guglielmo Vivarelli https://orcid.org/0000-0003-4731-3509

REFERENCES

1. Johnson FT, Tinoco EN, Yu NJ. Thirty years of development and application of CFD at Boeing commercial airplanes seattle. *Comput Fluids*. 2005;34(10):1115-1151.
2. Alauzet F, Loseille A. A decade of progress on anisotropic mesh adaptation for computational fluid dynamics. *Comput-Aid Des*. 2016;72:13-39.
3. Fidkowski KJ, Darmofal DL. Review of output-based error estimation and mesh adaptation in computational fluid dynamics. *AIAA J*. 2011;49(4):673-694.
4. Roy C. Strategies for driving mesh adaptation in CFD (Invited). Paper presented at: Proceedings of the 47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition; 2009;1302.
5. Roy C. Review of discretization error estimators in scientific computing. Paper presented at: Proceedings of the 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition; 2010;126.
6. Fidkowski KJ. Output-based error estimation and mesh adaptation for steady and unsteady flow problems. In: Deconinck H, Horvath T, eds. *Proceedings of the 38th Advanced CFD Lectures Series*. Waterloosesteenweg, Belgium: von Karman Institute for Fluid Dynamics; 2015.
7. Löhner R. Mesh adaptation in fluid mechanics. *Eng Fract Mech*. 1995;50(5-6):819-847.
8. Baker TJ. Mesh adaptation strategies for problems in fluid dynamics. *Finite Elements Anal Des*. 1997;25(3-4):243-273.
9. Qin N, Liu X. Flow feature aligned grid adaptation. *Int J Numer Methods Eng*. 2006;67(6):787-814.
10. Alauzet F. *Metric Based Anisotropic Mesh Adaptation*. CEA, Cadarache, France: Course material, CEA-EDF-INRIA Schools Numerical Analysis Summer School; 2010.
11. Peraire J, Vahdati M, Morgan K, Zienkiewicz OC. Adaptive remeshing for compressible flow computations. *J Comput Phys*. 1987;72(2):449-466.
12. Löhner R. Three-dimensional fluid-structure interaction using a finite element solver and adaptive remeshing. *Comput Syst Eng*. 1990;1(2-4):257-272.
13. Peraire J, Peiro J, Morgan K. Adaptive remeshing for three-dimensional compressible flow computations. *J Comput Phys*. 1992;103(2):269-285.
14. George PL, Hecht F, Vallet MG. Creation of internal points in Voronoi's type method. Control adaptation. *Adv Eng Softw Workstat*. 1991;13(5-6):303-312.
15. Ait-Ali-Yahia D, Habashi WG, Tam A, Vallet MG, Fortin M. A directionally adaptive methodology using an edge-based error estimate on quadrilateral grids. *Int J Numer Methods Fluids*. 1996;23(7):673-690.
16. Habashi WG, Dompierre J, Bourgault Y, Fortin M, Vallet MG. Certifiable computational fluid dynamics through mesh optimization. *AIAA J*. 1998;36(5):703-711.
17. Habashi WG, Fortin M, Dompierre J, Vallet MG, Bourgault Y. *Anisotropic mesh adaptation: a step towards a mesh-independent and user-independent CFD*. Dordrecht, Netherlands: Springer; 1998:99-117.
18. Robichaud M, Ait-Ali-Yahia D, Peeters M, Baruzzi G, Kozel V, Habashi W. 3-D anisotropic adaptation for external and turbomachinery flows on hybrid unstructured grids. Paper presented at: Proceedings of the Fluids 2000 Conference and Exhibit; 2000;2248.
19. Habashi WG, Dompierre J, Bourgault Y, Ait-Ali-Yahia D, Fortin M, Vallet MG. Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent CFD. Part I: general principles. *Int J Numer Methods Fluids*. 2000;32(6):725-744.
20. Ait-Ali-Yahia D, Baruzzi G, Habashi WG, Fortin M, Dompierre J, Vallet MG. Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent CFD. Part II. structured grids. *Int J Numer Methods Fluids*. 2002;39(8):657-673.
21. Harris MJ, Qin N. Using the medial axis to represent flow features for feature-aligned unstructured quad-dominant mesh generation. *Comput Fluids*. 2014;102:1-14.
22. Harris MJ, Qin N. Geometric representation of flow features using the medial axis for mesh generation. *AIAA J*. 2014;53(1):246-259.
23. Frey PJ, Alauzet F. Anisotropic mesh adaptation for CFD computations. *Comput Methods Appl Mech Eng*. 2005;194(48-49):5068-5082.
24. Mesri Y, Alauzet F, Loseille A, Hascoët L, Koobus B, Dervieux A. Continuous mesh adaptation models for CFD. *CFD J*. 2008;16(4):346-355.
25. Loseille A, Alauzet F. Continuous mesh framework Part I: well-posed continuous interpolation error. *SIAM J Numer Anal*. 2011;49(1):38-60.
26. Loseille A, Alauzet F. Continuous mesh framework Part II: validations and applications. *SIAM J Numer Anal*. 2011;49(1):61-86.

27. Frazza L, Loseille A, Alauzet F. Anisotropic mesh adaptation for turbomachinery applications. Paper presented at: Proceedings of the 23rd AIAA Computational Fluid Dynamics Conference; 2017;3299.
28. Becker R, Rannacher R. *Weighted a Posteriori Error Control in Finite Element Methods*. Technical Report. Universitat Heidelber; 1996.
29. Giles MB, Larson M, Levenstam M, Suli E. *Adaptive error control for finite element approximations of the lift and drag coefficients in viscous flow*. Oxford, UK: University of Oxford; 1997.
30. Giles MB. *On Adjoint Equations for Error Analysis and Optimal Grid Adaptation in CFD*. New York, NY: World Scientific; 1998:155-169.
31. Venditti DA, Darmofal DL. Adjoint error estimation and grid adaptation for functional outputs: application to quasi-one-dimensional flow. *J Comput Phys*. 2000;164(1):204-227.
32. Venditti DA, Darmofal DL. Grid adaptation for functional outputs: application to two-dimensional inviscid flows. *J Comput Phys*. 2002;176(1):40-69.
33. Venditti DA, Darmofal DL. Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows. *J Comput Phys*. 2003;187(1):22-46.
34. Park MA. Adjoint-based, three-dimensional error prediction and grid adaptation. *AIAA Journal.* 2004;42(9):1854-1862.
35. Park M. Three-dimensional turbulent RANS adjoint-based error correction. Paper presented at: Proceedings of the 16th AIAA Computational Fluid Dynamics Conference; 2003;3849.
36. Nemec M, Aftosmis M. Adjoint error estimation and adaptive refinement for embedded-boundary Cartesian meshes. Paper presented at: Proceedings of the 18th AIAA Computational Fluid Dynamics Conference; 2007;4187.
37. Lee-Rausch E, Park M, Jones W, Hammond D, Nielsen E. Application of parallel adjoint-based error estimation and anisotropic grid adaptation for three-dimensional aerospace configurations. Paper presented at: Proceedings of the 23rd AIAA Applied Aerodynamics Conference; 2005;4842.
38. Kim HJ, Takano Y, Nakahashi K. Error estimation and grid adaptation using euler adjoint method. *J Aircraft*. 2006;43(5):1317-1324.
39. Kim HJ, Nakahashi K. Output-based error estimation and adaptive mesh refinement using viscous adjoint method. Paper presented at: Proceedings of the 44th AIAA Aerospace Sciences Meeting and Exhibit; 2006;1395.
40. Dwight RP. Goal-oriented mesh adaptation for finite volume methods using a dissipation-based error indicator. *International Journal for Numerical Methods in Fluids.* 2008;56(8):1193-1200.
41. Dwight RP. Heuristic a posteriori estimation of error due to dissipation in finite volume schemes and application to mesh adaptation. *J Comput Phys*. 2008;227(5):2845-2863.
42. Jameson A, Schmidt W, Turkel E. Numerical solution of the Euler equations by finite volume methods using Runge Kutta time stepping schemes. Paper presented at: Proceedings of the 14th Fluid and Plasma Dynamics Conference; 1981;1259.
43. Fidkowski KJ, Roe PL. An entropy adjoint approach to mesh refinement. *SIAM J Sci Comput*. 2010;32(3):1261-1287.
44. Peter J, Nguyen-Dinh M, Trontin P. Goal oriented mesh adaptation using total derivative of aerodynamic functions with respect to mesh coordinates–With applications to Euler flows. *Comput Fluids*. 2012;66:194-214.
45. Nguyen-Dinh M, Peter J, Sauvage R, Meaux M, Désidéri JA. Mesh quality assessment based on aerodynamic functional output total derivatives. *Europ J Mech-B/Fluids*. 2014;45:51-71.
46. Todarello G, Vonck F, Bourasseau S, Peter J, Désidéri JA. Finite-volume goal-oriented mesh adaptation for aerodynamics using functional derivative with respect to nodal coordinates. *J Comput Phys*. 2016;313:799-819.
47. Resmini A, Peter J, Lucor D. Mono-block and non-matching multi-block structured mesh adaptation based on aerodynamic functional total derivatives for RANS flow. *Int J Numer Methods Fluids*. 2017;83(11):866-884.
48. Alonso JJ. Introduction to sensitivity analysis and the adjoint method: derivations, uses and properties. In: Deconinck H, Horvath T, eds. Proceedings of the 38th Advanced CFD Lectures Series. Waterloosesteenweg 72 B-1640 Sint-Genesius-Rode Belgium: von Karman Institute for Fluid Dynamics; 2015.
49. Venditti DA. *Grid Adaptation for Functional Outputs of Compressible Flow Simulations*. Massachusetts Institute of Technology; 2002.
50. Zervogiannis T, Liakopoulos P, Papadimitriou D, Giannakoglou KC. A grid enrichment and movement strategy for a posteriori error analysis in viscous flows. Paper presented at: Proceedings of the 7th European Conference on Turbomachinery, Fluid Dynamics and Thermodynamics; 2007; Athens.
51. Gugala M, Meyer M, Muller JD. Towards an output-based re-meshing for turbomachinery applications. Paper presented at: Proceedings of the ECCOMAS Congress 2016, VII European Congress on Computational Methods in Applied Sciences and Engineering; 2016.
52. Vivarelli G, Qin N, Shahpar S. Combined Hessian and Adjoint error-based anisotropic mesh adaptation for turbomachinery flows. Paper presented at: Proceedings of the 55th AIAA Aerospace Sciences Meeting; 2017;1946.
53. Vivarelli G, Qin N, Shahpar S, Radford D. Efficient adjoint-based mesh adaptation applied to turbo-machinery flows. Paper presented at: Proceedings of the American Society of Mechanical Engineers ASME Turbo Expo Turbomachinery Technical Conference and Exposition GT2018-77066; 2018.
54. Milli A, Shahpar S. PADRAM: parametric design and rapid meshing system for complex turbomachinery configurations. *ASME Paper No. GT2012-69030*. Two Park Avenue New York, NY 10016-5990: American Society of Mechanical Engineers; 2012.
55. Lapworth L. Hydra-CFD: a framework for collaborative CFD development. Paper presented at: Proceedings of the International Conference on Scientific and Engineering Computation (IC-SEC); 2004; Singapore.
56. Moinier P. *Algorithm Developments for an Unstructured Viscous Flow Solver*. Oxford University; 1999.
57. Sasaki D, Dhanasekaran C, Dawes W. *MeshPost Unserguide Version 1.1.0*. Moor Lane, Derby: Rolls-Royce; 2008.
58. Carnie G, Wang Y, Qin N. *MeshPost The User Guide of the Hessian-Based Mesh Adaption*. Moor Lane Derby DE24 8BJ: Rolls-Royce; 2011

59. Vivarelli G. *Anisotropic & Edgewise Adjoint Error Estimation & Grid Adaptation with Applications to Turbomachinery Flows*. University of Sheffield; 2019.
60. Spalart P, Allmaras S. A one-equation turbulence model for aerodynamic flows. Paper presented at: Proceedings of the 30th Aerospace Sciences Meeting and Exhibit; 1992;439.
61. Swanson RC, Turkel E. On central-difference and upwind schemes. *J Comput Phys*. 1992;101(2):292-306.
62. Swanson RC, Radespiel R, Turkel E. On some numerical dissipation schemes. *J Comput Phys*. 1998;147(2):518-544.
63. Roe PL. Approximate Riemann solvers, parameter vectors, and difference schemes. *J Comput Phys*. 1981;43(2):357-372.
64. van Leer B. Towards the ultimate conservative difference scheme. II. monotonicity and conservation combined in a second-order scheme. *J Comput Phys*. 1974;14(4):361-370.
65. Hirsch C. *Numerical Computation of Internal and External Flows: Volume 2: Computational Methods for Inviscid and Viscous Flows*. Chichester: John Wiley & Sons; 2002.
66. Jameson A. Origins and further development of the Jameson–Schmidt–Turkel scheme. *AIAA J*. 2017;55(5):1-23.
67. INRIA . TAPENADE http://www-sop.inria.fr/tropics/tapenade.html. Accessed March 10, 2019.
68. Giles MB. *On the Use of Runge-Kutta Time-Marching and Multigrid for the Solution of Steady Adjoint Equations*. Oxford, UK: University of Oxford; 2000.
69. Borouchaki H, Hecht F, Frey PJ. Mesh gradation control. *Int J Numer Methods Eng*. 1998;43(6):1143-1165.
70. Li X, Remacle JF, Chevaugeon N, Shephard MS. Anisotropic mesh gradation control. *13th International Meshing Roundtable*. Williamsburg, Virginia: Sandia National Laboratories; 2004:401-412.
71. Alauzet F. Size gradation control of anisotropic meshes. *Finite Element Anal Des*. 2010;46(1-2):181-202.
72. Joubarne E, Guibault F. 3D metric-based anisotropic mesh adaptation for vortex capture. *Math Comput Simulat*. 2011;82(1):163-180.
73. Blazek J. *Computational Fluid Dynamics: Principles and Applications*. Amsterdam, Netherlands: Elsvier; 2001.
74. Harris MJ. *Flow Feature Aligned Mesh Generation and Adaptation*. University of Sheffield; 2013.
75. Gander W. Algorithms for the QR decomposition. *Res Rep*. 1980;80(02):1251-1268.
76. Baker TJ. Mesh generation: art or science? *Progress Aerosp Sci*. 2005;41(1):29-63.
77. Balasubramanian R, Newman JC III. Comparison of adjoint-based and feature-based grid adaptation for functional outputs. *Int J Numer Methods Fluids*. 2007;53(10):1541-1569.
78. Dunham J. *CFD Validation for Propulsion System Components (la Validation CFD des Organes des Propulseurs)*. Neuilly-Sur-Seine, France: Advisory Group for Aerospace Research and Development; 1998.
79. Denton JD. Lessons from rotor 37. *J Thermal Sci*. 1997;6(1):1-13.
80. Suder KL. *Experimental Investigation of the Flow Field in a Transonic, Axial Flow Compressor with Respect to the Development of Blockage and Loss*; 1996.
81. Shabbir A, Celestina ML, Adamczyk JJ, Strazisar AJ. The effect of hub leakage flow on two high speed axial flow compressor rotors. Paper presented at: Proceedings of the ASME 1997 International Gas Turbine and Aeroengine Congress and Exhibition; 1997.
82. Hah C. Large eddy simulation of transonic flow field in NASA rotor 37. Paper presented at: Proceedings of the 47th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition; 2009;1061.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of this article.

How to cite this article: Vivarelli G, Qin N, Shahpar S, Radford D. Sequential feature-based mesh movement and adjoint error-based mesh refinement. *Int J Numer Meth Fluids*. 2020;1–24. https://doi.org/10.1002/fld.4882